Programa de Pós-Graduação em Engenharia Elétrica
Associação Ampla entre a UFSJ e o CEFET-MG

Sídney José de Paiva Soares

Controle de um Robô Móvel utilizando realimentação linearizante robusta

São João del-Rei
Junho de 2011.

Sídney José de Paiva Soares

Controle de um Robô Móvel utilizando realimentação linearizante robusta

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, associação ampla entre a UFSJ e o CEFET-MG, como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Modelagem e Controle de Sistemas.
Linha de Pesquisa: Análise e Modelagem de Sistemas.

Orientador: Erivelton Geraldo Nepomuceno
Co-Orientador: Valter Júnior de Souza Leite

São João del-Rei
Junho de 2011.

SOARES, Sidney José de Paiva.
S676c Controle de um robô móvel utilizando realimentação linearizante robusta / Sidney José de Paiva Soares.– 2011.
152f.; il.

Orientador: Erivelton Geraldo Nepomuceno
Co-orientador: Valter Júnior de Souza Leite

Dissertação (mestrado) – Universidade Federal de São João del-Rei, Departamento de Engenharia Elétrica.

Referências: f. 115-126.

1. Modelagem – Teses. 2. Modelo Dinâmico - Teses. 3. Controle robusto – Teses. 4. Desigualdades matriciais lineares – Teses. I. Universidade Federal de São João del Rei. Departamento de Engenharia Elétrica. II. Título.

CDU: 681.51

Sídney José de Paiva Soares
Engenheiro Eletricista – UFSJ

Controle de um Robô Móvel utilizando realimentação linearizante robusta

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, associação ampla entre a UFSJ e o CEFET-MG, como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Modelagem e Controle de Sistemas.
Linha de Pesquisa: Análise e Modelagem de Sistemas.

Banca Examinadora:

Prof. Dr. Erivelton Geraldo Nepomuceno
DEPEL/UFSJ

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

Prof. Dr. Gleison Françoares Amaral
DEPEL/UFSJ

Prof. Dra. Anna Helena Reali Costa
PCS/EPUSP

São João del-Rei
2011

Este trabalho é dedicado a minha irmã Josete de Paiva Soares (in memorian).

Agradecimentos

Agradeço em primeiro lugar a minha família e a minha esposa Andiara, pelo apoio incondicional e pela paciência e compreensão nos momentos ausentes.

A Deus, pela saúde e força para que pudesse superar as dificuldades e concretizar este sonho.

Ao meu orientador, professor Dr. Erivelton Geraldo Nepomuceno e ao meu co-orientador, professor Dr. Valter Júnior de Souza Leite pela confiança depositada, pelo apoio técnico, pelas longas conversas e principalmente pela paciência.

A todos os professores do DEPEL, bem como a todo o corpo técnico e colegas de pós-graduação. Agradeço a todos os amigos do GCoM pelas inúmeras conversas que muito contribuíram para o resultado alcançado.

Enfim, a todos aqueles que, direta ou indiretamente, contribuíram para a concretização deste trabalho. Muito obrigado a todos.

A mente que se abre a uma nova idéia
jamais voltará ao seu tamanho original.

Albert Einstein

Resumo

Os robôs móveis autônomos são dotados de habilidades para executar tarefas com uma mínima intervenção do homem. Para executar essas tarefas, é necessário dotá-los de capacidade de navegação autônoma. Uma tarefa simples, como deslocar o robô de um ponto a outro em um ambiente, requer a execução de um conjunto de etapas distintas. Primeiramente, é necessário o reconhecimento e representação do ambiente para que a trajetória possa ser planejada com base nessa representação. Essa trajetória, após ter sido planejada, é então gerada e enviada para o robô para que o rastreamento seja realizado, função essa, de responsabilidade da etapa de controle de trajetória.

Neste trabalho, o robô utilizado nas simulações, um robô omnidirecional com quatro rodas, é apresentado e seus modelos cinemático e dinâmico são desenvolvidos. Uma abordagem para a escolha da disposição angular das rodas de forma a melhorar o desempenho do robô é apresentada.

São propostas técnicas de controle de trajetória para robôs móveis omnidirecionais utilizando leis de controle baseadas em linearização por realimentação de estados em que os ganhos são gerados por técnicas de controle robusto. Duas técnicas para síntese de controladores robustos baseadas em desigualdades matriciais lineares são apresentadas e analisadas.

Na análise dos controladores propostos, o modelo do robô é considerado como sendo incerto, em que os parâmetros pertencem a um politopo com vértices conhecidos. Alguns exemplos são desenvolvidos para demonstrar a eficiência dos controladores propostos.

Palavras-chave: Modelagem, Modelo Dinâmico, Controle Robusto, Linearização por Realimentação de Estados, Desigualdades Matriciais Lineares.

Abstract

The autonomous mobile robots are provided with skills to perform tasks with minimal human intervention. To perform these tasks, it is necessary to provide the capability of autonomous navigation. A simple task, such as moving a robot from one point to another in an environment requires the implementation of a series of distinct stages. First, the environment is recognised and represented in order to plan a trajectory. After planning, the trajectory, is generated and sent to the robot so that the tracking could be performed in the stage of trajectory control.

In this work, the robot used in the simulations, a four-wheeled omnidirectional robot, is presented and its kinematic and dynamic models are developed. One approach to choose the angular disposition of the wheels to improve the performance of the robot is presented.

Trajectory control techniques are presented for omnidirectional mobile robots using control laws based on state feedback linearization in which the gains are generated by robust control procedures. Two techniques for synthesis of robust controllers based on linear matrix inequalities are presented and analyzed.

In the analysis of the proposed controllers, the robot model is considered to be uncertain, in which the parameters belong to a polytope with known vertices. Some examples are developed to demonstrate the efficiency of the proposed controllers.

Key-words: Modeling, Dynamic Model, Robust Control, State Feedback Linearization, Linear Matrix Inequalities.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos e Notação

AGV's	Autonomous Ground Vehicles (veículos terrestres autônomos)
BIBO	Bounded-Input Bounded-Output (Entrada limitada Saída Limitada)
EQ	Estabilidade Quadrática
EQA	Erro quadrático da aceleração resultante
F180	Categoria da Robocup
GPS	Global Positioning System (sistema de posicionamento global)
IA	Inteligência Artificial
LMI	Linear Matrix Inequality (desigualdade matricial linear)
MIMO	Multiple Input Multiple Output (múltiplas entradas e múltiplas saídas)
PD	Proporcional Derivativo
PI	Proporcional Integral
PID	Proporcional Integral Derivativo
PWM	Pulse Width Modulation (Modulação por largura de pulso)
RMO	Robôs Móveis Omnidirecionais
RRT	Rapidly Exploring Random Trees
SDP	Semidefinite Programming Problem (problema de programação semi definida)
SISO	Single Input Single Output (uma entrada e uma saída)
SSL	Small Size League

$\mathbb{R}$	conjunto dos números reais
$\mathbf{I}$	matriz identidade de dimensão apropriada
$\mathbf{0}$	matriz de zeros de dimensão apropriada
N	especialmente utilizada para denotar o número de vértices de um politopo
δ	especialmente utilizado para representar as incertezas de um sistema
x_0	especialmente utilizado para denotar o ponto de equilíbrio do sistema
S_r	representa o sistema de referência global do robô
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 do robô
X_m	coordenada x do robô no sistema de referência móvel
Y_m	coordenada y do robô no sistema de referência móvel
θ	coordenada de posição angular do robô
ξ_r	vetor de postura do robô no sistema S_r
ξ_m	vetor de postura do robô no sistema S_m
Ψ	azimute do robô no sistema de referência global
γ	azimute do robô no sistema de referência móvel

$\mathbf{f}_1$	força provida pelo motor 1
$\mathbf{f}_2$	força provida pelo motor 2
$\mathbf{f}_3$	força provida pelo motor 3
$\mathbf{f}_4$	força provida pelo motor 4
f_1	módulo da força $\mathbf{f}_1$
f_2	módulo da força $\mathbf{f}_2$
f_3	módulo da força $\mathbf{f}_3$
f_4	módulo da força $\mathbf{f}_4$
$\mathbf{f}_R$	força resultante
φ_1	ângulo entre o eixo da roda 1 e o eixo X_m
φ_2	ângulo entre o eixo da roda 2 e o eixo X_m
φ_3	ângulo entre o eixo da roda 3 e o eixo X_m
φ_4	ângulo entre o eixo da roda 4 e o eixo X_m
φ_d	ângulo dianteiro das rodas do robô
φ_t	ângulo traseiro das rodas do robô
$\phi(x)$	representa o difeomorfismo
$\mathcal{L}$	utilizado para denotar a derivada de Lie

Capítulo 1

Introdução

1.1 Histórico

A robótica é uma área da ciência e tecnologia caracterizada por relacionar-se fortemente com as áreas de mecânica, eletrônica e computação. No geral, essa área trata de sistemas compostos por máquinas automáticas e controladas por circuitos eletrônicos programáveis (Miyagi e Villani, 2004).

Originalmente, a área de robótica desenvolveu-se baseada na necessidade de encontrar soluções adequadas para necessidades técnicas tais como acesso a ambientes confinados, reabilitação de pacientes e sondas espaciais (Garcia et al., 2007). Entretanto, a rápida evolução e sofisticação atingida pela área fomentou pesquisa em diversas outras áreas, tais como a área industrial.

Essa tecnologia, hoje adotada por muitas fábricas e indústrias, tem obtido de um modo geral, êxito em questões levantadas sobre a redução de custos, aumento de produtividade e redução dos vários problemas trabalhistas com funcionários, gerados principalmente pela realização de trabalhos perigosos. Devido a esse êxito, estudos relacionados a robôs têm alcançado grandes desenvolvimentos nos últimos anos, veja por exemplo (Borenstein et al., 1997; Canudas de Wit, 1998; Kalmár-Nagy et al., 2004; Lages e Alves, 2006; Liu et al., 2008).

O termo robô foi pela primeira vez usado pelo checo Karel Capek (1890-1938) numa peça de teatro - *R.U.R. (Rossum's Universal Robots)* - estreada em Janeiro de 1921 (Praga). O termo robótica foi popularizado pelo escritor de ficção cientifica Isaac Asimov, na sua ficção *"I, Robot" (Eu, Robô), de 1950* (Asimov, 1950). Nesse mesmo livro, Asimov elaborou leis, que segundo ele, regeriam os robôs no futuro. Leis da robótica:

1. Um robô não pode fazer mal a um ser humano e nem, por omissão, permitir que algum mal lhe aconteça.

2. Um robô deve obedecer às ordens dos seres humanos, exceto quando essas contrariarem a Primeira lei.

3. Um robô deve proteger a sua integridade física, desde que, com isto, não contrarie a Primeira e a Segunda leis.

A ideia de se construir robôs começou a tomar força no início do século XX com a necessidade de aumentar a produtividade e melhorar a qualidade dos produtos. É nessa época que o robô industrial encontrou suas primeiras aplicações. O primeiro robô digital programável foi inventado por George Devol em 1954 e foi chamado de "Unimate". O primeiro "Unimate" foi instalado em uma fábrica da General Motors em 1960 na cidade de *Trenton* em *New Jersey*, onde foi utilizado para elevar peças de metal quente de uma máquina de fundição. Atualmente, devido aos inúmeros recursos que os sistemas de microcomputadores oferecem, a robótica atravessa uma época de contínuo crescimento que poderá permitir a transformação da ficção de robôs inteligentes de décadas anteriores em realidade.

A robótica tem possibilitado às empresas redução de custos com o operariado e um significativo aumento na produção. O país que mais tem investido na robotização das atividades industriais é o Japão, um exemplo disso observa-se na Toyota (Sakai e Amasaka, 2007). Porém há um ponto negativo nisso tudo. Ao mesmo tempo que a robótica beneficia as empresas diminuindo gastos e agilizando processos, ela cria o desemprego estrutural, que é aquele que não é gerado por crises econômicas, mas pela substituição do trabalho humano por máquinas. Ressalta-se entretanto que há alguns ramos da robótica que geram impacto social positivo, veja por exemplo robôs bombeiros (Kececi, 2009), submarinos (Excell, 2008) e cirurgiões (Balicki et al., 2010), que são na realidade ferramentas para preservar o ser humano. Um robô pode ainda auxiliar a reintegrar algum profissional que teve parte de suas capacidades motoras reduzidas devido a doença ou acidente e, a partir da utilização da ferramenta robótica ser reintegrado ao mercado. Apesar dessa controvérsia, é incontestável a importância do desenvolvimento científico dessa área. Outras aplicações da robótica podem ser encontradas na nanotecnologia (para a construção de nano robôs a fim de realizar intervenções em seres humanos de forma menos invasiva) (Hill et al., 2008; Bogue, 2010; Swetha et al., 2010) e na produção industrial (os robôs que são criados para produção e desenvolvimento de mercadorias) (Masey et al., 2010).

Apesar de ainda ser predominante o uso de robôs de base fixa, ou robôs manipuladores, nas indústrias, os robôs de base móvel, ou simplesmente robôs móveis, estão cada vez mais presentes, exercendo tarefas de transporte de peças (Wu et al., 2010), inspeção de plataformas de perfuração de petróleo, inspeção de oleodutos, etc. A substituição do trabalho do homem na exploração de locais perigosos, como águas profundas, áreas radioativas, crateras de vulcões, ambientes espaciais são também exemplos de tarefas que podem ser exercidas por esses robôs.

Inicialmente, os robôs foram utilizados principalmente na área industrial, porém com o avanço tecnológico, os robôs começaram a ser utilizados para outros fins tais como lazer, atividades domésticas, etc. Nesse contexto, surgiram os robôs móveis autônomos com a proposta de realizar uma variedade de tarefas mais complexas que não eram possíveis a seus antecessores.

1.2 Robótica Móvel

A robótica móvel vem sendo objeto de estudo de vários pesquisadores devido ao uso em várias aplicações (Kim et al., 2000). No Brasil esse campo tem sido pesquisado por diversos grupos. Veja por exemplo (Ramirez et al., 2003; Selvatici e Costa, 2007; Ferreira et al., 2009).

Em várias situações o robô é inserido em um ambiente, onde deve tomar decisões para se locomover de um ponto a outro, desviar de obstáculos, seguir objetos dentre outros. Para que tais decisões sejam tomadas, o robô deve ter o mínimo de conhecimento sobre o ambiente ao seu redor (Sujan et al., 2006; Papadopoulos e Misailidis, 2007). É necessário que o robô possua comportamentos autônomos, visando atingir alvos estabelecidos, ao mesmo tempo em que desvia de obstáculos.

Um robô móvel autônomo necessita de mecanismos de locomoção que possibilitem o movimento pelo ambiente sem a necessidade de ser controlado pelo homem. Existe uma grande variedade de movimentos que podem ser executados por um robô, e a escolha de um sistema de locomoção apropriado é um aspecto importante para seu projeto.

Siegwart e Nourbakhsh (2004) classificam os robôs móveis em duas categorias, os acionados por pernas e os acionados por rodas. A principal desvantagem da locomoção com pernas é a complexidade no mecanismo do robô e consequentemente no seu sistema de controle. Os robôs dessa categoria são geralmente mais lentos que os robôs com rodas.

Os robô móveis com rodas são bastante utilizados na robótica por serem rápidos e de fácil locomoção e com uma mecânica de complexidade relativamente baixa, porém com difícil adaptação em certos terrenos. Por se mostrarem bastante eficientes e possuírem mecanismos relativamente simples de serem implementados, as rodas têm sido o mais popular mecanismo de locomoção.

Segundo Siegwart e Nourbakhsh (2004), as rodas podem ser classificadas em cinco categorias: rodas padrão fixa; rodas padrão manobráveis; rodas castor; rodas esféricas e rodas omnidirecionais. Cada um desses modelos possui diferentes graus de liberdade. A geometria, quantidade e disposição das rodas sobre a base do robô são responsáveis pelo seu grau de mobilidade, pela sua capacidade de manobra e pela sua estabilidade.

As rodas padrão fixas possuem apenas um grau de liberdade, a rotação sobre o eixo perpendicular à sua superfície de maior área. Ao contrário das omnidirecionais, elas possuem restrições de deslizamento lateral. Devido a essa restrição, os robôs diferenciais têm um grau de mobilidade reduzido, não podendo se deslocar em todas as direções sem mudar de orientação.

Os robôs omnidirecionais usam rodas que possuem rolamentos sobre a sua superfície de contato. Esses rolamentos diminuem o atrito de deslizamento lateral da roda fazendo com que essa passe a ter um grau de liberdade a mais. Um robô com pelo menos três rodas desse tipo pode se deslocar em qualquer direção, independentemente da sua orientação.

Os robôs móveis têm sido muito estudados nos últimos anos, em particular, os robôs móveis terrestres, também conhecidos como AGV's (autonomous ground vehicles). Os AGV's são veículos motorizados equipados com rodas e controlados por sistemas computadorizados que operam sem intervenção humana.

A principal característica de um robô móvel é a sua capacidade de locomoção em um determinado ambiente, sendo capaz de exercer tarefas individualmente ou em cooperação com outros robôs. Em geral, essa locomoção é feita de forma autônoma. A autonomia e a cooperação são características que envolvem pesquisas nas áreas de inteligência artificial, sobretudo no campo de agentes autônomos e multi agentes. Segundo Lages (1998), a capacidade de locomoção autônoma torna-se atrativa basicamente devido a dois fatores: necessidade de interação com um

ambiente pouco estruturado e necessidade de operação sem supervisão.

A falta de estrutura do ambiente no qual, geralmente, um robô móvel se encontra é proveniente do seu amplo espaço de trabalho. Diferentemente dos robôs de base fixa, o espaço de trabalho de um robô móvel não é limitado pelas suas características intrínsecas. Em consequência disto, o ambiente de operação é geralmente dinâmico e desconhecido, aumentando portanto a complexidade de navegação do robô.

O objetivo principal da navegação autônoma de robôs é permitir que o robô navegue pelo ambiente em segurança, evitando a colisão com obstáculos, até alcançar uma ou mais posições pré-estabelecidas no ambiente. O desenvolvimento de robôs autônomos torna-se mais complexo à medida que o conjunto de características intrínsecas ao ambiente no qual esses estão inseridos aumenta. Para a interação em ambientes reais, o robô deve ter a capacidade de contornar situações imprevistas e adaptar-se às mudanças. Uma das abordagens propostas para o problema da navegação autônoma de robôs é dividir a navegação em quatro etapas: representação do ambiente, planejamento, geração e controle da trajetória (Franco, 2006a).

Dentro do contexto de pesquisa e desenvolvimento de robôs móveis, algumas organizações foram criadas para fomentar o interesse nesse assunto. Um exemplo é a *Robocup*[1]. A *Robocup* é uma competição mundial que acontece anualmente. Visa o estudo e desenvolvimento da Inteligência Artificial (IA) e da Robótica, fornecendo desafios e problemas em que várias tecnologias e metodologias podem ser combinadas para obter os melhores resultados. Cada edição é constituída por duas partes essenciais: as competições e o simpósio. No simpósio, que decorre após as competições, são apresentados e discutidos trabalhos científicos da área. As equipes participantes nas competições apresentam as soluções desenvolvidas, sendo essa uma forma de incentivar a contínua evolução de uma equipe, assim como o aparecimento de novas equipes já com boa qualidade. A primeira edição ocorreu em 1997 em Nagoya no Japão, depois de se ter organizado um pré-Robocup em Osaka no Japão para identificar possíveis problemas e dificuldades na organização desse tipo de evento a uma escala global. A iniciativa foi lançada pelo Dr. Hiroaki Kitano (Kitano et al., 1997, 1998a,b), um pesquisador na área da Inteligência Artificial que se tornou o presidente e fundador da *RoboCup Federation*, com o objetivo de dinamizar a evolução da IA, em particular dos Sistemas Autônomos Multi-agente.

Inicialmente, o futebol foi a grande motivação para o Robocup. O futebol é um esporte bastante popular em quase todo o mundo, mas apresenta também desafios científicos bastante importantes, com problemas individuais (identificação e localização de objetos, dribles, chute, etc.), e ao mesmo tempo, problemas coletivos (estratégia, passes, etc.). A existência de um ambiente dinâmico e de ser jogado contra um adversário cujo objetivo é exatamente o oposto são também aspectos que obrigam a uma grande sofisticação dos jogadores (robôs). No *RobocupSoccer* estão englobadas as competições de futebol robótico, nas suas variadas formas.

Nos últimos anos, as competições de futebol de robôs têm sido objeto de estudo de grupos de pesquisa em diversas instituições de ensino em vários países (Douret et al., 2003; Yoshimura et al., 2003; Kraft e Hofmann, 2003; Schroeder et al., 2005; Cao et al., 2007a; Barnes e Zelinsky, 2008; Celiberto Jr. et al., 2008; Soares et al., 2009), inspirando pesquisas em sistemas multi-agentes, navegação autônoma, fusão multisensorial, reconhecimento de padrões, visão em tempo real

[1]Para mais informações consulte www.robocup.org

e controle inteligente de múltiplos robôs. Essas competições foram propostas como um problema de *benchmark* com complexidade suficiente para desenvolvimento e comparação de novos métodos nos campos de inteligência artificial e sistemas multi robôs.

Desde sua introdução para o campo de pesquisa em inteligência artificial, as competições de futebol de robôs tornaram-se uma ampla área de pesquisa. Essas competições apresentam um ambiente real, dinâmico e incerto, que exige respostas em tempo real de um time de robôs, que devem de forma autônoma trabalhar em cooperação afim de derrotar a equipe adversária (Gurzoni et al., 2011).

A razão para tal popularidade vem da diversidade de desafios presentes nessas competições, incluindo aprendizagem de máquina, sistemas multi-agentes, visão computacional, modelagem, teoria de controle e dentre outras. Existem vários tópicos a serem considerados e disciplinas necessárias para a tarefa de construir uma equipe capaz de competir em campeonatos nacionais e mundiais. Embora os principais tópicos de investigação sejam imediatamente evidentes, existe uma distância significativa entre a concepção da ideia e ter todos os robôs trabalhando em conjunto.

Dentro das competições da *Robocup*, existem várias categorias, sendo que uma das mais populares atualmente é a *SSL - Small Size League* da *Robocup*, também conhecida como F180. Nessa categoria, dois times de cinco robôs omnidirecionais[2] disputam uma partida com dois tempos de dez minutos cada, sendo as regras semelhantes às do futebol humano. As dimensões dos robôs devem estar de acordo com as regras da categoria F180, sendo que esses devem estar contidos em um cilindro com 180mm de diâmetro e altura de 150mm.

Jogos de futebol entre robôs constituem experiências reais e atividades de teste para o desenvolvimento de robôs inteligentes e que cooperam entre si para atingir uma meta. Esses robôs são classificados como robôs móveis autônomos, dessa forma, pode-se utilizar das quatro etapas de navegação citadas na início desta seção.

Em uma partida de futebol, os dois times de robôs móveis são monitorados por uma câmera de vídeo, que captura a imagem do jogo e envia para um computador para ser realizado o processamento digital da imagem conforme Figura 1.1.

Após o processamento da imagem, as posições dos robôs e da bola são conhecidas (etapa de representação do ambiente). O software residente no computador, deve então, por meio de algoritmos previamente implementados, definir qual a estratégia a ser utilizada (etapa de planejamento) e então gerar os sinais de comando (etapa de geração de trajetória) para serem enviados aos robôs através de uma rede sem fio (*wireless*). A última etapa, o controle de trajetória, é realizada através de controladores aplicados ao modelo matemático do robô e devem garantir fidelidade na execução da trajetória gerada.

Nas teorias de controle clássica e moderna, um passo importante para implementar o controle de um processo é derivar o modelo matemático que descreve tal processo. O procedimento para obtenção do modelo requer que se conheça detalhadamente o processo a ser controlado, o que nem sempre é factível se o processo é muito complicado. A qualidade do controle de um processo está diretamente relacionada com a qualidade do modelo obtido. Para desenvolvimento

[2]Esses robôs recebem esse nome porque possuem rodas omnidirecionais. Essas rodas possuem rolamentos de giro livre transversais ao eixo de rotação normal, de forma a reduzir o atrito quando a roda é "arrastada".

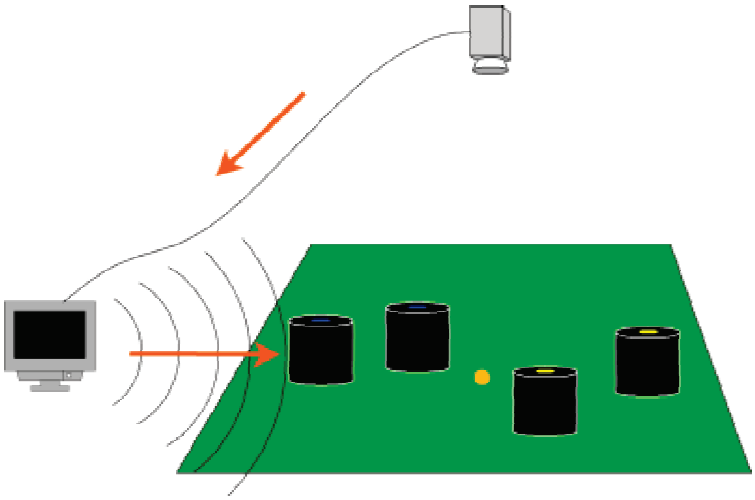


Figura 1.1: Representação do ambiente do futebol de robôs. (Fonte: www.robocup.org)

de um modelo matemático que represente com fidelidade o comportamento do robô, vários aspectos devem ser levados em consideração, principalmente aqueles relacionados à sua geometria. Aspectos de carácter geométrico são importantes pois possuem relação direta com os modelos cinemático e dinâmico do robô. A forma como as rodas são montadas no robô influenciam seu deslocamento e podem gerar atrito em excesso, elevado consumo energético e até mesmo comprometer a locomoção em determinadas direções se não forem corretamente dispostas. Dessa forma, para desenvolver o modelo matemático do robô, é necessário um estudo da influência da disposição física das rodas na periferia do robô de forma a construir um robô que melhore sua capacidade de locomoção.

Neste trabalho, como em vasta literatura, percebe-se que o desenvolvimento de um modelo e projeto de um controle para problemas de robótica móvel passam pela utilização de equações diferenciais não lineares (Slotine e Li, 1991; Khalil, 2002). Uma abordagem bastante utilizada para o trabalho com sistemas não lineares é a linearização, o que possibilita a aplicação do amplo conjunto de ferramentas existente para sistemas lineares (Chen, 1999). A linearização por realimentação de estados baseia-se na construção de uma lei de controle que cancele de forma exata as não linearidades do sistema, permitindo a imposição de uma nova dinâmica linear (Isidori, 1989). Essa abordagem possui a vantagem de ser válida para grande parte do espaço dos estados, porém para o projeto de tal controlador é condição necessária o conhecimento dos parâmetros e estados do sistema, o que não ocorre na maioria das situações reais. Quando houver incertezas paramétricas ou de medição, o cancelamento não será exato e o comportamento do sistema irá diferir do desejado, podendo inclusive levar à instabilidade.

A linearização clássica prevê a transformação do sistema não linear em uma cadeia de integradores na forma de Brunovsky, que sabidamente é pouco robusta a incertezas (Strang, 1988). Uma linearização por realimentação de estados menos sensível a variações paramétricas foi proposta por (Guillard, 2000). Nessa abordagem, os autores propõem que a dinâmica não linear do sistema seja transformada em sua aproximação linear em torno de um ponto através da linearização por realimentação de estados.

A incerteza que afeta um sistema pode ter sua origem nos erros de modelagem, nas variações

paramétricas, nas incertezas sobre os parâmetros por causa da precisão, nas aproximações de modelagem tais como linearização ou eliminação de dinâmicas elevadas. Sendo assim, é necessário representar tais incertezas de maneira adequada, de acordo com a sua origem, a fim de elaborar uma lei de controle que as leve em consideração. A preocupação com sistemas incertos deu origem à Teoria de Controle Robusto (Zhou et al., 1996).

Projetar controladores robustos para sistemas sujeitos a incertezas paramétricas tem sido um dos principais objetivos da teoria de controle robusto multivariável nas últimas décadas (Langner, 2004). O termo robusto está associado à capacidade do controlador em assegurar a estabilidade e/ou desempenho do sistema tanto para o seu modelo nominal quanto para toda família de modelos que o representa.

Estabilidade é um requisito mínimo em sistemas de controle. Na maioria das situações práticas, contudo, um bom controlador deve também garantir respostas temporais suficientemente rápidas e bem amortecidas. Uma maneira comum para garantir uma resposta transitória satisfatória é a alocação dos polos de malha fechada em uma determinada região no plano complexo. Essa técnica é conhecida como alocação regional de polos, em contraste com a alocação exata de polos, em que os polos são endereçados a um local específico no plano complexo.

Devido ao fato de que os sistemas reais sempre envolvem alguma incerteza, é natural a preocupação com a robustez do conjunto de polos que representa toda a família de modelos gerados pelas perturbações, isto é, é necessário saber se os polos permanecem na região especificada quando o modelo nominal é perturbado.

O estudo de sistemas lineares com parâmetros incertos avançou muito nas últimas duas décadas graças a técnicas de investigação de domínios de estabilidade e de controle robusto derivadas de funções de Lyapunov (Leite et al., 2004). A chamada estabilidade quadrática (EQ), isto é, a existência de uma mesma função de Lyapunov, independente dos parâmetros incertos, assegurando a estabilidade robusta do sistema para o domínio de incertezas considerado, foi talvez o resultado mais importante da década de 80 (Barmish, 1985). A partir das condições de estabilidade quadrática, inúmeros resultados de análise, controle e filtragem robusta com critérios como as normas $\mathcal{H}_2$ e $\mathcal{H}_\infty$ puderam ser desenvolvidos (veja por exemplo (Boyd et al., 1994) e referências internas).

Essas condições, quase sempre formuladas em termos de desigualdades matriciais lineares (em inglês, *LMIs - Linear Matrix Inequalities*), tornaram-se facilmente resolvíveis numericamente com o aparecimento de pacotes computacionais especializados (Gahinet et al., 1995; Sturm, 1999; Lofberg, 2004).

Embora a estabilidade quadrática seja especialmente adequada à análise de sistemas incertos com parâmetros variantes no tempo, os resultados obtidos podem ser bastante conservadores em muitas situações. Várias extensões têm aparecido na literatura para a análise e síntese de controladores para sistemas lineares incertos. Resultados menos conservadores têm sido obtidos a partir de funções de Lyapunov dependentes de parâmetros (Feron et al., 1996; Gahinet et al., 1996; Neto, 1999; Mori e Kokame, 2000). Em (Leite e Peres, 2005) três condições para a existência de um ganho de malha fechada robusto na forma de LMIs são apresentadas. Além de um ganho baseado na estabilidade quadrática e outro baseado em um sistema aumentado em que usa-se uma função de Lyapunov dependente de parâmetros, os autores propõem uma condição

que utiliza um número maior de equações e que gera um ganho de malha fechada dependente de parâmetros.

Neste trabalho, o modelo matemático do robô F180 é desenvolvido. O modelo proposto apresenta a característica de ser genérico, dessa forma, é possível sua utilização em qualquer robô omnidirecional de quatro rodas, bastando para isso ajustar seus parâmetros. Além disso, um estudo detalhado para escolha da distribuição angular das rodas é apresentado. O modelo obtido é considerado como sendo incerto e as incertezas são representadas na forma politópica em que tanto a matriz dinâmica do sistema quanto a matriz de entradas são afetadas por essas incertezas. Duas condições LMIs para síntese de controladores robustos são apresentadas e aplicadas ao modelo não linear do robô F180 submetido a uma linearização por realimentação de estados. A primeira condição é baseada na estabilidade quadrática e a segunda utiliza um sistema aumentado com função de Lyapunov dependente de parâmetros, conforme (Leite e Peres, 2005). Vários exemplos numéricos são desenvolvidos para demonstrar a eficiência dos métodos empregados.

1.3 Proposta

Algumas questões estão envolvidas no controle de trajetória para robôs móveis com rodas e devem ser analisadas cuidadosamente.

Os aspectos mecânicos, elétricos e físicos do robô são relevantes para a formulação das leis de controle usadas no projeto do controlador de trajetória, pois influenciarão nos modelos dinâmico e cinemático do robô. O número de rodas e como essas serão dispostas na periferia do robô, além de influenciarem no projeto mecânico e no desempenho dos controladores devido a sua influência nos modelos cinemático e dinâmico do robô, são essenciais na estabilidade do mesmo sobre sua superfície de contato. Além disso, na modelagem, assume-se que todos os atuadores são idênticos, no entanto, isso geralmente não acontece, o que implica que o modelo pode conter incertezas nos parâmetros. Essas incertezas podem influenciar na estabilização do modelo, fazendo com que o robô torne-se instável.

A proposta deste trabalho é o desenvolvimento do modelo matemático que descreva com fidelidade o comportamento de um robô móvel omnidirecional e a concepção de um módulo de controle que faça que esse seja capaz de convergir para uma determinada trajetória mesmo na presença de incertezas em seus parâmetros. Apresenta-se também uma contribuição para o posicionamento das rodas do robô F180.

1.4 Organização do Texto

Conceitos e definições relevantes para o desenvolvimento desta dissertação, como o conceito de rodas omnidirecionais e de linearização por realimentação são apresentados no Capítulo 2. No Capítulo 3, o modelo matemático do robô F180 é desenvolvido, incluindo o modelo cinemático e o modelo dinâmico da base móvel com inclusão dos atuadores. Um estudo detalhado sobre a distribuição de aceleração em função da distribuição das rodas na periferia do robô é apresentado. No Capítulo 4, a técnica de linearização por realimentação robusta é aplicada ao modelo do

robô F180. O modelo linearizado é então utilizado em uma topologia de controle do tipo seguimento de referência, em que a síntese dos controladores é realizada utilizando condições LMIs. 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.

Capítulo 2

Conceitos Preliminares e Definições

Este Capítulo apresenta uma breve revisão de literatura, em que itens relevantes para o entendimento deste trabalho são reproduzidos. Primeiramente, os tipos de modelo matemático para robôs acionados por rodas são apresentados e suas diferenças são destacadas. Em seguida são apresentadas técnicas para linearização por realimentação de estados.

2.1 Locomoção de um Robô Móvel

A função mais básica de um robô móvel é a sua locomoção, que geralmente é regida por um planejamento de trajetória e executada por uma estratégia de controle. Existem quatro etapas que devem ser cumpridas por um robô móvel autônomo para que essa função seja executada com eficiência. Essas etapas estão bem definidas em níveis hierárquicos de abstração, tendo no nível mais alto a representação do ambiente, seguida da etapa de planejamento da trajetória. No terceiro e no quarto níveis se encontram, respectivamente, a geração e o controle da trajetória conforme Figura 2.1.

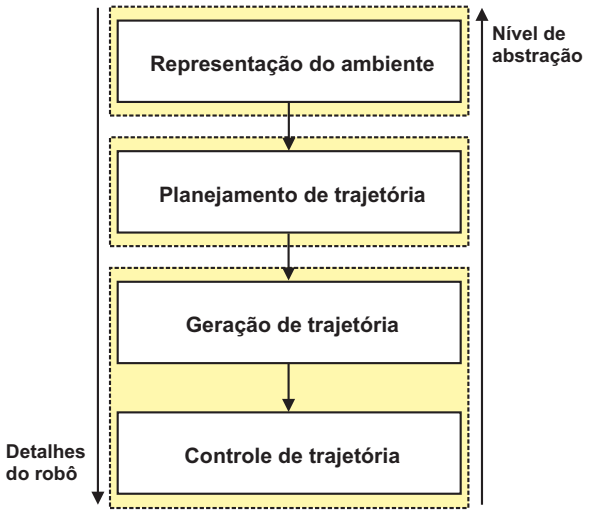


Figura 2.1: Diagrama mostrando as etapas de representação do ambiente, planejamento, geração e controle de trajetória. (Fonte: (Franco, 2006a)).

A representação do ambiente é a etapa de mais alto nível. A sua função é mapear o ambiente identificando os objetos nele contidos e suas respectivas localizações. Um ambiente pode ser estático, se suas características não mudam com o tempo, ou dinâmico, caso contrário.

O planejamento da trajetória depende explicitamente do ambiente no qual o robô está inserido e da tarefa a ser executada por ele, podendo ser considerados alguns critérios como menor gasto de energia na execução da trajetória, caminhos com menor número de obstáculos, menor distância entre uma posição de origem e de destino, tempo de execução do trajeto e diversos outros critérios que dependem do objetivo a ser alcançado. Em (Purwin et al., 2008) é apresentado um algoritmo para planejamento de trajetórias para agentes autônomos de forma a garantir trajetórias com desvio de obstáculos em tempo real.

Uma vez planejada, a trajetória deve ser construída pela etapa de geração de trajetória, por meio de uma interpolação polinomial ou por funções contínuas. O grande desafio dessa etapa é encontrar polinômios interpoladores contínuos ou um conjunto de funções contínuas que constituam, com fidelidade, a trajetória planejada (Franco, 2006a). Pode-se apontar alguns trabalhos dentro desse escopo: em (Paromtchik e Rembold, 1994; Paromtchik e Asama, 2000) é abordada a técnica de geração de trajetória por funções *spline*; Egerstedt e Martin (2001) apresentam uma técnica baseada em controle ótimo para geração de trajetórias por funções *splines* suaves; Andrade et al. (2001); Garrido et al. (2002) utilizam redes neurais para geração de trajetórias; Almeida (2005) propõe uma técnica de geração de trajetória por curvas parametrizadas através da técnica *B-Spline* com a utilização de um algoritmo para o ajuste da curva. Em (Arantes et al., 2009), é desenvolvimento de um sistema de navegação baseado em autômatos celulares e no algoritmo de propagação *Wavefront*.

A geração da trajetória constrói um caminho que deve ser rastreado pelo robô. O comportamento desse rastreamento é função da etapa de controle de trajetória. Nessa etapa, características como a dinâmica do robô e de seus atuadores, bem como a sua cinemática e as restrições cinemáticas do seu sistema mecânico de locomoção devem ser levadas em consideração. Exemplos de trabalho sobre controle de trajetória de robôs móveis podem ser encontrados em (Reis, 2005) que implementa controladores robustos para o rastreamento de trajetória e (Oubbati, 2006) que aplica um controlador adaptativo para o controle de robôs móveis. Podem ser encontrados em alguns trabalhos o uso de controladores adaptativos baseados em redes neurais (Nelson, 1989; Yang et al., 1998; Oliveira, 2001; Gu e Hu, 2002; Gomes, 2006).

O nível de menor abstração, o controle de trajetória, tem como objetivo manter o robô sobre uma trajetória de referência. É necessário, nesse nível, conhecer detalhes sobre os aspectos mecânicos, geométricos e elétricos do robô e de seus atuadores, formulando os seus modelos dinâmico e cinemático e, com isso encontrar as leis de controle que façam o robô convergir para a trajetória desejada através de sinais elétricos enviados para seus atuadores.

Segundo Lages (1998), os modelos para robôs acionados por rodas podem ser divididos em quatro tipos:

- Modelo cinemático de postura;

- Modelo cinemático de configuração;

- Modelo dinâmico de postura;

- Modelo dinâmico de configuração.

Os modelos cinemáticos descrevem o robô em função da velocidade e orientação das rodas, enquanto os modelos dinâmicos descrevem o robô em função das forças generalizadas aplicadas pelos atuadores, por exemplo, torques nas rodas. Alguns trabalhos encontrados na literatura (Bloch et al., 1992; Kolmanovsky e McClamroch, 1995; Murata e Hirose, 1993; Huang et al., 2004; Rojas e Förster, 2006) descrevem o robô em coordenadas cartesianas e utilizam apenas um modelo cinemático de postura, no entanto, um grande número de autores considera também o modelo dinâmico não linear (Yamamoto e Yun, 1994; Watanabe et al., 1998; Carter et al., 2001; Williams et al., 2002; Schroeder et al., 2005; Franco, 2006a; Cao et al., 2007b; Liu et al., 2008).

Os modelos de postura consideram como estado apenas a posição e orientação do robô, enquanto os modelos de configuração consideram além da postura outras variáveis internas, como deslocamento angular das rodas. Do ponto de vista de controle da posição e orientação espacial do robô, apenas os modelos de postura são necessários.

A Figura 2.2 mostra a interação entre as três últimas etapas apresentadas na Figura 2.1 e o robô.

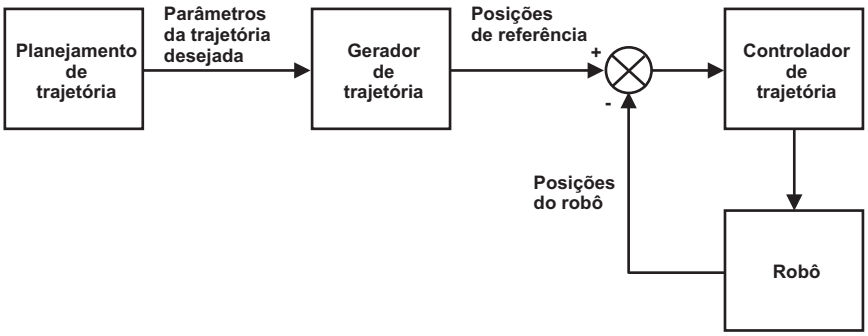


Figura 2.2: Interação entre as etapas de planejamento, geração e controle de trajetória e o robô móvel. (Fonte: (Franco, 2006a)).

O projeto de controladores para sistemas não lineares com incertezas paramétricas, tal como manipuladores robóticos e robôs móveis, é um problema de difícil solução (Purwin e Andrea, 2006). Essas incertezas podem ser geradas devido às mudanças drásticas na operação de tais sistemas, o uso de modelos simplificados ou imprecisão na medição de determinados parâmetros. Como a maioria dos sistemas físicos são naturalmente não lineares, existe definitivamente uma necessidade de compreender e controlar esses sistemas. Para esse fim, algoritmos computacionais desenvolvidos com base na teoria de controle não linear ou técnicas capazes de linearizar esse sistema de forma a utilizar controladores lineares são de interesse especial no controle moderno.

O projeto de tais controladores consiste de duas partes:(i) modelagem matemática e análise de estabilidade, (ii) desenvolvimento de algoritmos capazes de reduzir tais incertezas. A primeira parte trata de realizar uma análise teórica e de estabilidade com base na teoria de controle não

linear. A segunda diz respeito ao projeto do controlador, desenvolvimento de algoritmos e implementação.

Existe um interesse especial no desenvolvimento de modelos matemáticos que descrevam o comportamento de robôs e na implementação de algoritmos capazes de controlá-los. Um dos principais objetivos das pesquisas no campo da robótica móvel é o desenvolvimento de plataformas móveis que operam em ambientes povoados. Para muitas tarefas é, portanto, altamente desejável que um robô possa acompanhar as posições de objetos em movimento em seu redor.

Em muitas situações é necessário que tais robôs sejam capazes de identificar e seguir algum objeto. Essa técnica é conhecida na literatura como rastreamento, do inglês *tracking*. O rastreamento de objetos é amplamente utilizado em aplicações de robótica (Schulz et al., 2003). Para o rastreamento de objetos é, portanto, necessário conhecer sua trajetória. O planejamento de trajetória tem recebido grande atenção por parte dos pesquisadores, pois seu desenvolvimento está diretamente relacionado com a maior autonomia dos robôs. A complexidade do problema de planejamento de trajetória tem motivado o desenvolvimento dos mais diversos algoritmos. O progresso da pesquisa nessa área tem permitido o uso de robôs móveis em diversas situações, tais como aplicações domésticas, automotivas e de exploração, dentre outras, fazendo com que as interações entre um robô e um alvo a ser rastreado aumentem, levando a trajetórias mais complexas que não podem ser rastreadas por modelos simples, pois os robôs agem sobre os alvos alterando sua posição e velocidade (Kwok e Fox, 2004).

Um algoritmo de planejamento de trajetória é projetado para determinar o caminho que leva um objeto móvel (robô) de uma configuração inicial até uma configuração determinada (objetivo), evitando configurações indesejadas (obstáculos) nesse caminho. Uma das técnicas de planejamento de trajetória conhecidas na literatura é a que utiliza campos potenciais virtuais nos obstáculos e destino (Soares et al., 2009). Outras se baseiam no mapeamento do ambiente e subsequente procura pelo melhor caminho até o alvo. Em qualquer situação pode-se procurar um caminho que maximize ou minimize determinada quantidade, como tempo, energia, aceleração e assim por diante. No entanto, é desejável que o robô desvie de obstáculos e simultaneamente seja capaz de executar outras tarefas. Zickler e Veloso (2008) apresentam uma variação da técnica *RRT - Rapidly Exploring Random Trees* de forma a melhorar a integração entre o desvio de obstáculos e a execução de estratégias de jogo para robôs móveis jogadores de futebol. Isso torna o planejamento de trajetórias e o rastreamento de objetos funções complementares que podem aumentar a autonomia dos robôs.

O planejamento de trajetórias pode ser estático ou dinâmico, dependendo da forma pela qual a informação sobre o obstáculo está disponível. Em um problema estático, toda a informação sobre o obstáculo é conhecida *a priori* e o movimento do robô é estabelecido a partir dessa informação. No planejamento dinâmico, somente uma informação parcial sobre os obstáculos está disponível. Para alcançar o alvo estabelecido, o robô planeja a trajetória baseado na informação disponível. Conforme o robô segue seu percurso, descobre mais informações sobre os obstáculos, o que possibilita um novo planejamento do percurso. Nas competições da *Robocup*, considera-se o planejamento dinâmico em um ambiente não estruturado, de forma que o robô não tem informação sobre todo o espaço de trabalho e a trajetória a ser rastreada é estabelecida em função da informação obtida *on-line* do sistema de visão computacional sobre a presença de

obstáculos em seu percurso. Para o rastreamento de um objeto em movimento, é necessário o conhecimento de alguns de seus parâmetros tais como posição, velocidade e aceleração, e para tal, podem ser utilizadas técnicas de identificação de sistemas para estimação de tais parâmetros.

O problema de estimação de posição de objetos móveis tem se tornado um importante problema na robótica móvel. Ao longo do últimos anos, vários robôs móveis têm sido implementados em ambientes povoados como edifícios de escritório (Arras e Vestli, 1998; Asoh et al., 1997; Horswill, 1993; Simmons et al., 1997), supermercados (Endres et al., 1998), hospitais (Engelberger, 1993) e museus (Burgard et al., 1999; Thrun et al., 1999). Nesse contexto, surgiram os robôs móveis autônomos com a proposta de realizar uma variedade de tarefas mais complexas. O conhecimento da posição e velocidade de objetos móveis pode ser utilizado de várias formas para melhorar significativamente o comportamento de tais sistemas. Por exemplo, essa informação permite que o robô adapte sua velocidade para a velocidade do objeto em movimento. Ser capaz de distinguir um objeto estático de um objeto móvel pode também melhorar o desempenho, uma vez que isso possibilita que o robô melhore sua habilidade de desviar-se de obstáculos em situações nas quais a trajetória do robô cruza com a trajetória de um objeto em movimento (Tadokoro et al., 1995).

Diversas técnicas têm sido empregadas no intuito de estimar a posição e velocidade de objetos móveis. Veja por exemplo (Takaba et al., 1996) em que é apresentado um método para compensação de incertezas do modelo utilizando redes neurais multi camadas incorporadas a um filtro de *Kalman*. Em (Vaidehi et al., 2001), a capacidade adaptativa do filtro de *Kalman* é melhorada com o auxílio de redes neurais para rastreamento de múltiplos objetos. (Zhoua e Aggarwalb, 2006) apresentam um método para rastreamento de objetos em movimento em ambientes externos, em que um rastreamento robusto é alcançado utilizando a fusão de características e múltiplas câmeras. A trajetória obtida de cada câmera é incorporada através da aplicação de um filtro de *Kalman* estendido.

Loebis et al. (2004) descrevem a implementação de um sistema de navegação inteligente baseado na integração de informações vindas de um sistema *GPS - Global Positioning System* e de sensores de um sistema de navegação inercial. A fusão dos dados do GPS e dos sensores é realizada através de um filtro de *Kalman* simples e de um filtro de *Kalman* estendido. É destacado o uso de técnicas de lógica *Fuzzy* para a adaptação das condições iniciais de ambos os filtros implementados devido à possíveis alterações nas características de ruído dos sensores.

Martins et al. (2008) elaboraram um controlador adaptativo para guiar um robô móvel do tipo monociclo. A estabilidade do sistema é analisada utilizando a teoria de Lyapunov. Gu e Veloso (2009) consideram tarefas nas quais os robôs agem sobre um alvo que é visualmente rastreado, tal como chutar uma bola ou empurrar um objeto. Os autores introduzem uma forma de incorporar o modelo de interação entre o alvo e o robô ao algoritmo de rastreamento de forma a aumentar o desempenho do rastreador. Sadhu et al. (2006) quantificam a robustez relativa de algumas variações de filtro de *Kalman* para rastreamento. Matía et al. (2006) apresentam uma nova forma de utilização de filtro de *Kalman* estendido associado com lógica *Fuzzy*. Sua principal diferença em comparação com outros filtros de *Kalman* comumente tratados na literatura que utilizam a lógica fuzzy é que, nesse filtro proposto, a lógica fuzzy está incluída na definição das variáveis, representando sua incerteza utilizando distribuições de

possibilidade, em vez de distribuições de Gauss.

Em (Liu et al., 2008) é apresentado o projeto de um controlador não linear para um robô móvel omnidirecional. O controlador consiste de um controle baseado na cinemática e outro baseado na dinâmica, os quais foram projetados usando o método de controle por linearização de trajetória baseado no modelo dinâmico não linear do robô. Purwin e Andrea (2006) descrevem um algoritmo para calcular trajetórias quase-ótimas com o mínimo tempo para veículos omnidirecionais com quatro rodas.

Desde sua introdução, as competições propostas pela *Robocup* tornaram-se de grande relevância em inteligência artificial, uma vez que possuem várias características encontradas em outros problemas reais de grande complexidade. Exemplos desses problemas são: sistemas de automação robótica que podem ser vistos como um grupo de robôs em uma tarefa de montagem (Mills e Ing, 1996; Sun e Mills, 2002), tarefas de transporte em grupo (Nouyan et al., 2006) e missões espaciais com vários robôs (Tambe, 1998; Huntsberger et al., 2004).

Durante uma partida de futebol de robôs, um dos pontos de maior influência na qualidade do jogo é o passe de bola entre os robôs. O passe de bola é uma habilidade elementar e frequentemente utilizada em partidas de futebol humano, porém, para uma partida de futebol de robôs se torna um tanto quanto complexa, uma vez que diversos fatores podem influenciar no jogo, como restrições cinemáticas do robô, atrito com a superfície do campo, atraso no recebimento de comandos, incertezas paramétricas e a própria dinâmica do robô. Em (Liu et al., 2004) é desenvolvido um estudo de técnicas para passe de bola entre robôs jogadores de futebol.

Identifica-se, através das competições da *Robocup*, que um controle preciso de trajetória para o robô é um dos pontos chaves para o bom desempenho das equipes. O controlador de trajetórias, deve garantir através de um controle em malha fechada que o caminho criado seja seguido (Kalmár-Nagy et al., 2004). Desde que esses robôs irão trabalhar em um ambiente dinâmico, o seguimento de trajetórias deve ser capaz de controlar o robô de forma a seguir o caminho previamente criado pelo planejador de trajetórias.

Nos últimos anos, a maioria das pesquisas sobre robôs móveis omnidirecionais concentram-se em projeto mecânico e análise dinâmica (Liu et al., 2008). A modelagem dinâmica e controladores precisos para seguimento de referência têm sido cuidadosamente estudados. Muitos modelos dinâmicos assumem que os motores são ideais e que o torque desenvolvido pode perfeitamente responder aos comandos enviados ao robô (Choi e Kim, 2001). Contudo, as limitações dinâmicas dos motores como também incertezas em seus parâmetros podem afetar o comportamento do robô, especialmente quando o robô é acelerado ou desacelerado. Em (Watanabe et al., 1998; Watanabe, 1998) é apresentado o desenvolvimento do modelo dinâmico de um robô omnidirecional, e várias estratégias de controle são analisadas. Em (Watanabe et al., 1998), controladores PI e PD baseados no controle de aceleração são desenvolvidos. Já (Watanabe, 1998) apresenta um controlador PID e um controlador *fuzzy*. Em (Huang et al., 2004), um controlador PID baseado no controle de aceleração é investigado. Tal controlador considera apenas o modelo cinemático do robô e, além disso, apenas uma trajetória em linha reta e uma circular são testadas.

2.2 Rodas Omnidirecionais

Rodas omnidirecionais têm se tornado populares em robôs móveis por permitirem movimento em linha reta de um ponto inicial até outro ponto sem ter que rotacionar primeiro. Além disso, movimentos de translação através de qualquer caminho desejado podem ser combinados com rotação de forma que o robô chegue ao destino com o ângulo correto.

As rodas omnidirecionais são todas baseadas no mesmo princípio. Enquanto a roda fornece tração na direção normal ao eixo do motor, essa pode deslizar sem atrito na direção do eixo do motor. A roda é construída usando rodas menores montadas ao longo da periferia da roda principal conforme Figura 2.3. Esse tipo de roda é empregado neste trabalho.

Figura 2.3: Modelo de roda omnidirecional utilizada pela equipe 5DPO. (Fonte: (Oliveira et al., 2008b)).

Duas ou mais rodas omnidirecionais são usadas para conduzir o robô. Cada roda provê tração na direção normal ao eixo do motor e paralela ao piso. As forças somadas geram movimentos de translação e rotação. As rodas podem mover o robô para qualquer direção, mas como elas estão localizadas em sua periferia, elas também podem rotacionar sua estrutura. Configurações populares de robôs da F180 usam três ou quatro rodas, veja por exemplo, as Figuras 2.4 e 2.5 respectivamente.

Qualquer robô com três ou mais rodas pode ser projetado para ser omnidirecional. A grande diferença entre a forma construtiva adotada pelas equipes competindo na liga SSL (Visser et al., 2008; Iocchi et al., 2009) é que algumas utilizam três rodas, enquanto outras utilizam quatro. Embora os sistemas baseados em três rodas sejam mais simples do que os de quatro, principalmente em termos mecânicos, os robôs com quatro rodas apresentam outras vantagens. Mantendo o mesmo tipo de motores, possuem maior aceleração, maior velocidade e melhor aderência ao piso. No entanto, essas vantagens são obtidas à custa de maiores custos de equipamentos e maior consumo energético. As vantagens em termos de aceleração e velocidade são também dependentes de fatores como o tipo de rodas utilizadas e os atritos presentes em todo o sistema. A solução ótima para utilização de robôs omnidirecionais de três ou quatro rodas depende de diversos fatores. Dessa forma, o ponto a partir do qual se torna vantajosa a utilização de quatro rodas não é claro e merece um estudo aprofundado (Oliveira et al., 2008a).

Existem duas razões principais que favorecem o uso de quatro rodas para as competições da *Robocup*. Uma delas é que mais motores significam mais torque, consequentemente mais aceleração, o que é altamente desejado em um jogo em que os robôs podem atravessar o campo

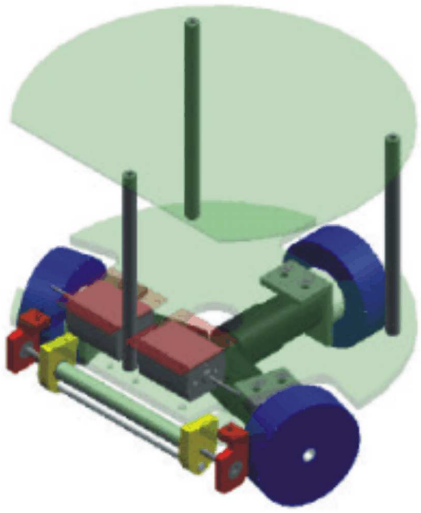


Figura 2.4: Robô F180 com três rodas omnidirecionais. (Fonte: (Carter et al., 2001)).

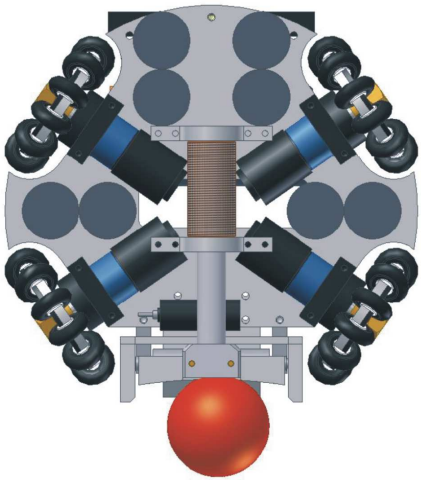


Figura 2.5: Robô F180 com quatro rodas omnidirecionais. (Fonte: (Rojas e Förster, 2006)).

em pouco mais de um segundo. A outra é o fato de que com três rodas, movimentos para algumas direções possuem significantemente menos aceleração do que em outras, o que leva a mais restrições quanto ao posicionamento das rodas. Essas restrições tornam-se ainda mais evidentes uma vez que as rodas não são as únicas partes a serem montadas na periferia do robô. Mecanismos de chute e drible também são necessários como pode ser visto nas Figuras 2.4 e 2.5.

Variações nos ângulos de disposição das rodas implicam em variação na distribuição de torque dos atuadores, o que pode restringir os torques máximos totais (somados entre todos os atuadores) para determinadas direções de translação. Esse aspecto será investigado neste trabalho e está apresentado na Seção 3.3.

2.3 Linearização por Realimentação

2.3.1 Introdução

Um dos métodos mais difundidos para o controle de um sistema não linear consiste em projetar um controlador linear para a aproximação desse sistema em torno de um ponto de operação e, a seguir, aplicá-lo diretamente ao sistema não linear (Franco, 2006b). A principal vantagem desse método é de poder utilizar qualquer uma das diversas técnicas de síntese de controladores da teoria de sistemas lineares, que é mais simples e mais conhecida que a teoria de sistemas não lineares. Por outro lado, uma grande desvantagem desse método é que a aproximação linear do sistema é válida em uma região limitada em torno do ponto de operação e, durante o seu funcionamento, o estado do sistema pode se afastar consideravelmente dessa região. Nesse caso, o sistema apresenta um comportamento diferente do esperado, o que pode reduzir a eficácia do controlador. A ideia da linearização por realimentação surgiu como uma maneira de contornar esse problema.

A linearização por realimentação é uma técnica difundida na área de controle não linear (El'youssef, 2007). Essa técnica tem por base gerar um sinal de controle que seja capaz de cancelar não linearidades presentes em um sistema que pretende-se controlar, ao qual, aplicando-se um mapeamento, conhecido como difeomorfismo[1] (veja definição 2.1 extraída de (Slotine e Li, 1991)), obtém-se um sistema linear.

Definição 2.1 (Difeomorfismo) *A aplicação $\phi : \Omega \to \mathbb{R}^n$, na qual Ω é um conjunto aberto de $\mathbb{R}^n$, é denominada um difeomorfismo se $\phi^{-1}(x)$ existe e se $\phi(x)$ e $\phi^{-1}(x)$ são diferenciáveis e contínuas em Ω. Se, adicionalmente, $\Omega = \mathbb{R}^n$, então $\phi(x)$ é um difeomorfismo global.*

No sistema linearizado é possível a aplicação das técnicas de controle linear, que são bem difundidas e consolidadas na literatura. O diagrama mostrado na Figura 2.6 ilustra o princípio de funcionamento dessa técnica.

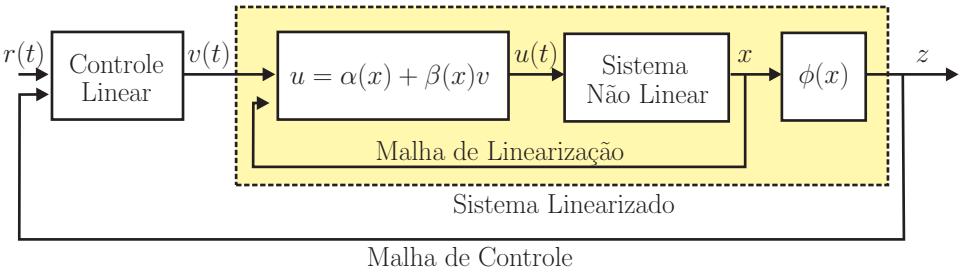


Figura 2.6: Diagrama mostrando duas malhas de realimentação. A malha interna lineariza o sistema e a malha externa aplica um controlador sobre o sistema linearizado.

Nesse método, existe uma primeira malha de realimentação que transforma o sistema não linear em um sistema linear e, então, uma segunda malha com um controlador linear que atua sobre a primeira malha. Evidentemente, em relação ao método mencionado anteriormente,

[1] Pode-se entender difeomorfismo como uma generalização do conceito de transformação de similaridades para sistemas lineares (Isidori, 1989).

mantém-se a vantagem de utilizar uma técnica de síntese linear para o projeto do controlador. Além disso, a desvantagem é eliminada, pois realiza-se uma transformação exata do sistema não linear em um sistema linear ao invés de uma simples aproximação.

Na Figura 2.6, $r(t)$ é a referência de controle, $u(t)$ representa a lei de controle linearizante, tratada aqui também como transformação de entrada, $\phi(x)$ representa o difeomorfismo, tratado também como transformação de estados, x são os estados do sistema não linear, z são os estados do sistema linearizado e $v(t)$ representa uma nova entrada de controle para o sistema linearizado, que é projetada utilizando as técnicas de controle linear.

A realimentação linearizante é projetada através do cálculo de uma transformação de estados (difeomorfismo) e uma transformação de entrada. O difeomorfismo é calculado para que o sistema não linear seja transformado em um sistema, também não linear, porém com as não linearidades diretamente relacionadas às entradas de controle. Então uma transformação de entrada é calculada para cancelar as não linearidades e prover uma nova entrada de controle para o sistema linearizado.

Na literatura existem duas abordagens da realimentação linearizante. A primeira, denominada linearização por realimentação clássica, desenvolvida no início dos anos 80 em trabalhos como (Hunt et al., 1983). Esse é atualmente um método bem conhecido, apresentado em muitos livros didáticos de controle não linear como (Isidori, 1989; Nijmeijer, 1990; Slotine e Li, 1991; Khalil, 2002). Nesse método, o sistema não linear é transformado em um sistema linear na forma canônica de Brunovsky, que corresponde a cadeias de integradores. Apesar de permitir o projeto de um controlador linear, essa forma apresenta um inconveniente, pois é extremamente sensível a variações de parâmetros, além de não ter nenhuma semelhança com o sistema original. Exemplos de utilização desse método são dados em (Ottoni e Lages, 2003; Serrano, 2007; Freitas, 2007; Rohr, 2008). A segunda abordagem, denominada linearização por realimentação robusta, foi proposta em (Guillard, 2000) com o objetivo de obter maior robustez para o sistema controlado. Nesse método, o sistema resultante, após a transformação, corresponde à aproximação linear do sistema original em torno de um ponto de operação. Essa característica traz vantagens, pois mantém informações do sistema original. Assim, o comportamento do sistema linearizado resultante é semelhante ao comportamento do sistema original, o que implica em maior robustez do sistema em malha fechada, pois, em caso de 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. Em (Franco, 2006b; El'youssef, 2007) podem ser vistos exemplos de aplicação dessa técnica.

Naturalmente, a aplicação de ambas as técnicas possuem restrições. Necessita-se 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 às condições necessárias e suficientes que garantam que ele seja linearizável pela realimentação, ao menos em uma região do espaço de estados. Outra restrição é a necessidade de acesso à informação de todos os estados, que representam grandezas utilizadas na modelagem dinâmica do sistema, para implementação da

lei de controle.

2.3.2 Formulação do Problema para Linearização por Realimentação

Seja o sistema não linear afim na entrada com n variáveis de estados e m entradas descrito pela equação de estados

$$\dot{x} = f(x) + g(x)u = f(x) + \sum_{i=1}^{m} g_i(x)u_i, \tag{2.1}$$

com $x \in \mathbb{R}^n$ representando o estado, $u \in \mathbb{R}^m$ a entrada de controle e $f(x)$, $g_1(x), \ldots,$ $g_m(x)$ campos vetoriais suaves definidos em um subconjunto aberto de $\mathbb{R}^n$. Assume-se que o estado x é conhecido e que o ponto de equilíbrio seja x_0.

O objetivo da linearização por realimentação é encontrar uma transformação de coordenadas dada por um difeomorfismo $\phi(x)$ tal que

$$z = \phi(x), \tag{2.2}$$

com $z \in \mathbb{R}^n$, e uma realimentação de estados

$$u(x,v) = \alpha(x) + \beta(x)v, \tag{2.3}$$

sendo $v \in \mathbb{R}^m$ a nova variável de controle, que permitam transformar o sistema (2.1) em um sistema linear na forma

$$\dot{z} = Az + Bv, \tag{2.4}$$

na qual (A,B) é um par controlável.

A linearização por realimentação é implementada conforme o diagrama de blocos mostrado na Figura 2.6. Assim, o sistema visto de v a z é linear e pode ser tratado como tal.

Inicialmente, é necessário definir as condições que o sistema (2.1) deve satisfazer para que possa ser linearizado por realimentação. Essas condições são apresentadas na Seção 2.3.3.

2.3.3 Condições para Linearização por Realimentação

Antes de seguir com a apresentação das condições necessárias e suficientes para linearização por realimentação, será introduzido o conceito de grau relativo para sistemas não lineares. De forma simplificada, o grau relativo é definido como o número de vezes que é necessário derivar a saída do sistema para atingir a entrada de controle (veja definição 2.2 extraída de (Slotine e Li, 1991)).

Definição 2.2 (Grau Relativo) *Seja o sistema não linear:*

$$\begin{aligned} \dot{x} &= f(x) + g(x)u, \\ y &= h(x), \end{aligned} \tag{2.5}$$

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 $f(x)$, $g_1(x), \ldots, g_m(x)$, $h(x)$ são campos vetoriais suaves em um subconjunto aberto de $\mathbb{R}^n$. Cada saída $h_j(x)$ tem grau relativo r_j, se:

$$\mathcal{L}_{g_i} h_j(x) = \mathcal{L}_{g_i}\mathcal{L}_f h_j(x) = \ldots = \mathcal{L}_{g_i}\mathcal{L}_f^{r_j-2} h_j(x) = 0 \qquad \forall\, i \in [1,m] \tag{2.6}$$

e se existe pelo menos um i tal que:

$$\mathcal{L}_{g_i}\mathcal{L}_f^{r_j-1} h_j(x) \neq 0. \tag{2.7}$$

O sistema não linear (2.5) tem grau relativo $r = r_1 + \ldots + r_p$.

De acordo com a definição 2.2, o sistema (2.1) tem grau relativo r em um ponto de equilíbrio, x_0, se for necessário derivar r vezes a saída y para que entrada u apareça no equacionamento.

A seguir defini-se distribuição e involutividade. Essas definições foram extraídas de (Franco, 2006b).

Definição 2.3 (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_pM.*

Definição 2.4 (Involutividade) *Uma distribuição D é chamada involutiva se $[X,Y] \in D$ sempre que X e Y são campos vetoriais de D.*

Sejam as distribuições $G_0 \subset G_1 \subset \ldots G_{n-1}$ associadas ao sistema não linear (2.1) e construídas de acordo com

$$G_i = \text{span}\{ad_f^k g_j : 0 \leq k \leq i, 1 \leq j \leq m\}, \qquad 0 \leq i \leq n-1,$$

em que $ad_f g = [f,g]$ representa o colchete de Lie, conforme definição A.4 apresentada no apêndice A.

Assume-se que as distribuições G_i 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.

Conforme (Isidori, 1989), se as distribuições $G_0, G_1, \cdots, G_{n-1}$ satisfazem as hipóteses **H1**, **H2** e **H3**, então existem funções escalares $\lambda_1(x), \cdots, \lambda_m(x)$ definidas em uma vizinhança U de x_0, de graus relativos $r_1, \cdots, r_m$, com $r_1 + \cdots + r_m = n$, tais que:

- para todo $i \in [1,m]$, todo $j \in [1,m]$ e todo $x \in U$

$$\mathcal{L}_{g_i}\lambda_j(x) = \mathcal{L}_{g_i}\mathcal{L}_f\lambda_j(x) = \cdots = \mathcal{L}_{g_i}\mathcal{L}_f^{r_j-2}\lambda_j(x) = 0.$$

- A matriz $E(x)$ dada por

$$E(x) = \begin{bmatrix} \mathcal{L}_{g_1}\mathcal{L}_f^{(r_1-1)}\lambda_1 & \cdots & \mathcal{L}_{g_m}\mathcal{L}_f^{(r_1-1)}\lambda_1 \\ \vdots & \ddots & \vdots \\ \mathcal{L}_{g_i}\mathcal{L}_f^{(r_m-1)}\lambda_m & \cdots & \mathcal{L}_{g_m}\mathcal{L}_f^{(r_m-1)}\lambda_m \end{bmatrix}. \tag{2.8}$$

é não singular para todo $x \in U$.

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

$$\lambda(x) = \begin{bmatrix} \lambda_1(x) \\ \vdots \\ \lambda_m(x) \end{bmatrix} \tag{2.9}$$

é escolhido como a saída do sistema (2.1), resultando no novo sistema

$$\begin{aligned} \dot{x} &= f(x) + g(x)u = f(x) + \sum_{i=1}^{m} g_i(x)u_i, \\ y &= \lambda(x). \end{aligned} \tag{2.10}$$

Esse sistema possui m entradas e m saídas, sendo por isso chamado um sistema quadrado.

Se as distribuições $G_0, G_1, \cdots, G_{n-1}$ satisfazem as hipóteses **H1**, **H2** e **H3**, então o sistema não linear (2.10) é linearizável por realimentação.

Tendo definido 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.

2.3.4 Linearização Clássica

Ao aplicar a técnica de realimentação linearizante clássica ao sistema (2.10), tem-se como objetivo obter uma transformação de estados (difeomorfismo) e uma transformação de entrada que linearizem esse sistema, colocando-o na forma canônica de Brunovsky (2.11):

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_c v_c, \\ y &= C_c x_c, \end{aligned} \tag{2.11}$$

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

$$A_c = \begin{bmatrix} A_{c_1} & 0_{r_1\times r_2} & \cdots & 0_{r_1\times r_m} \\ 0_{r_2\times r_1} & A_{c_2} & \cdots & 0_{r_2\times r_m} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{r_m\times r_1} & 0_{r_m\times r_2} & \cdots & A_{c_m} \end{bmatrix}, \tag{2.12}$$

$$B_c = \begin{bmatrix} B_{c_1} & 0_{r_1\times 1} & \cdots & 0_{r_1\times 1} \\ 0_{r_2\times 1} & B_{c_2} & \cdots & 0_{r_2\times 1} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{r_m\times 1} & 0_{r_m\times 1} & \cdots & B_{c_m} \end{bmatrix}, \tag{2.13}$$

$$C_c = \begin{bmatrix} C_{c_1} & 0_{1\times r_2} & \cdots & 0_{1\times r_m} \\ 0_{1\times r_1} & C_{c_2} & \cdots & 0_{1\times r_m} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{1\times r_1} & 0_{1\times r_2} & \cdots & C_{c_m} \end{bmatrix}, \tag{2.14}$$

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

Assume-se que existe um difeomorfismo $\phi_c(x) : \mathbb{R}^n \to \mathbb{R}^n$ válido numa região Ω para o sistema (2.10). A aplicação dessa transformação de estados sobre o sistema não linear deve transformá-lo em um sistema, também não linear, porém com as não linearidades diretamente relacionadas ao controle, como parte do sistema na forma de cascata de integradores.

Para transformar o sistema não linear (2.10) no sistema linear (2.11) é necessário determinar um difeomorfismo $\phi_c(x)$ e uma transformação de entrada $u_c(x)$ que cancele as não linearidades desse sistema e forneça uma nova entrada de controle.

Dessa forma, para se obter a linearização, deve-se derivar y na equação (2.10) no tempo até que a entrada do sistema apareça na equação resultante. Derivando a saída y no tempo, obtém-se

$$\dot{y} = \frac{\partial \lambda(x)}{\partial x}\dot{x}. \tag{2.15}$$

Substituindo $\dot{x}$ na equação (2.15), tem-se

$$\begin{aligned} \dot{y} &= \frac{\partial \lambda(x)}{\partial x}[f(x) + g(x)u], \\ \dot{y} &= \frac{\partial \lambda(x)}{\partial x}f(x) + \frac{\partial \lambda(x)}{\partial x}g(x)u. \end{aligned} \tag{2.16}$$

Utilizando a notação da derivada de Lie (veja definição A.3 no apêndice A), tem-se

$$\dot{y} = \mathcal{L}_f\lambda(x) + \mathcal{L}_g\lambda(x)u. \tag{2.17}$$

Estendendo essas derivadas para todas as saídas do sistema, a equação (2.17) pode ser escrita de forma mais geral, conforme mostrado na equação (2.18).

$$\dot{y}_j = \mathcal{L}_f\lambda_j(x) + \sum_{i=1}^{m}(\mathcal{L}_{g_i}\lambda_j(x))u, \tag{2.18}$$

sendo $\mathcal{L}_f\lambda_j(x)$ 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 da saída $\lambda_j(x)$ correspondente aos termos de $g_i(x)$, representadas por

$$\mathcal{L}_f\lambda_j(x) = \sum_{i=1}^{m} \frac{\partial\lambda_j(x)}{\partial x_i}\dot{x}_{if}, \tag{2.19}$$

$$\mathcal{L}_{g_i}\lambda_j(x) = \sum_{i=1}^{m} \frac{\partial\lambda_j(x)}{\partial x_i}\dot{x}_{ig}. \tag{2.20}$$

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

$$y_j^{(r_j)} = \mathcal{L}_f^{(r_j)}\lambda_j(x) + \sum_{i=1}^{m} (\mathcal{L}_{g_i}\mathcal{L}_f^{(r_j-1)}\lambda_j(x))u_i, \tag{2.21}$$

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,

$$\sum_{j=1}^{n} r_j = n, \tag{2.22}$$

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. Nesse caso tem-se dois subsistemas, um linearizável e outro denominado dinâmica nula. Neste trabalho, considera-se apenas sistemas com grau relativo igual ao número de estados, sendo, portanto, totalmente linearizáveis.

A equação (2.21) pode ser escrita na forma

$$\begin{bmatrix} y_1^{r_1} \\ \vdots \\ y_m^{r_m} \end{bmatrix} = \begin{bmatrix} \mathcal{L}_f^{r_1}\lambda_1(x) \\ \vdots \\ \mathcal{L}_f^{r_m}\lambda_m(x) \end{bmatrix} + E(x) \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}, \tag{2.23}$$

em que $E(x)$ é uma matriz $m \times m$ definida na equação (2.8) e repetida aqui para facilitar o entendimento.

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

A matriz $E(x)$ é chamada de matriz de desacoplamento do sistema MIMO (*Multiple-Input Multiple-Output*). Se essa matriz é não-singular, a transformação da entrada

$$\begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} = -E^{-1}(x) \begin{bmatrix} \mathcal{L}_f^{r_1}\lambda_1(x) \\ \vdots \\ \mathcal{L}_f^{r_m}\lambda_m(x) \end{bmatrix} + E^{-1}(x) \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix}, \tag{2.24}$$

leva a uma relação linear entre a saída y_j e a nova entrada v_j

$$\begin{bmatrix} y_1^{r_1} \\ \vdots \\ y_m^{r_m} \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix}. \tag{2.25}$$

Note que a relação entrada-saída acima é desacoplada e linear. Uma lei de controle na forma (2.24) é chamada de lei de controle desacoplada ou lei de controle não-interativa. Como resultado do desacoplamento, pode ser usado um sistema de uma entrada e uma saída (SISO) para cada canal $y_j - v_j$ da dinâmica desacoplada acima, de modo a obter controladores estáveis (Slotine e Li, 1991).

Utilizando a notação mostrada na equação (2.3), a equação (2.24) pode ser reescrita conforme (2.26).

$$u_c(x,v_c) = \alpha_c(x) + \beta_c(x)v_c. \tag{2.26}$$

Em que $\alpha_c(x)$ e $\beta_c(x)$ são dadas pelas equações (2.27) e (2.28) respectivamente.

$$\alpha_c(x) = -E^{-1}(x) \begin{bmatrix} \mathcal{L}_f^{r_1}\lambda_1(x) \\ \vdots \\ \mathcal{L}_f^{r_m}\lambda_m(x) \end{bmatrix}, \tag{2.27}$$

$$\beta_c(x) = E^{-1}(x). \tag{2.28}$$

Após a determinação da realimentação linearizante, o problema a ser solucionado é, agora, a determinação do difeomorfismo $\phi_c(x)$ que transforme, juntamente com (2.26), o sistema (2.10) no sistema (2.11). Sem perda de generalidade, será assumido a igualdade $\phi_c(x) = x_c(x)$. Conforme (Slotine e Li, 1991), o difeomorfismo pode ser encontrado como:

$$\phi_c(x) = \begin{bmatrix} \phi_{c_1}(x) \\ \vdots \\ \phi_{c_m}(x) \end{bmatrix}, \tag{2.29}$$

em que

$$\phi_{c_j}(x) = \begin{bmatrix} \lambda_j(x) \\ \mathcal{L}_f\lambda_j(x) \\ \vdots \\ \mathcal{L}_f^{r_j-1}\lambda_j(x) \end{bmatrix}. \tag{2.30}$$

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

$$\dot{x}_c(x) = \frac{\partial x_c(x)}{\partial x}\dot{x}. \tag{2.31}$$

Substituindo $\dot{x}$ na equação (2.31), tem-se

$$\begin{aligned} \dot{x}_c(x) &= \frac{\partial x_c(x)}{\partial x}[f(x) + g(x)u], \\ \dot{x}_c(x) &= \frac{\partial x_c(x)}{\partial x}f(x) + \frac{\partial x_c(x)}{\partial x}g(x)u. \end{aligned} \tag{2.32}$$

Substituindo (2.26) em (2.32), tem-se:

$$\begin{aligned} \dot{x}_c(x) &= \frac{\partial x_c(x)}{\partial x}f(x) + \frac{\partial x_c(x)}{\partial x}g(x)[\alpha_c(x) + \beta_c(x)v_c], \\ \dot{x}_c(x) &= \frac{\partial x_c(x)}{\partial x}f(x) + \frac{\partial x_c(x)}{\partial x}g(x)\alpha_c(x) + \frac{\partial x_c(x)}{\partial x}g(x)\beta_c(x)v_c. \end{aligned} \tag{2.33}$$

Utilizando a notação da derivada de Lie, tem-se

$$\dot{x}_c(x) = \mathcal{L}_f x_c(x) + \mathcal{L}_g x_c(x)\alpha_c(x) + \mathcal{L}_g x_c(x)\beta_c(x)v_c. \tag{2.34}$$

A equação (2.34) pode então ser reescrita conforme (2.35), que representa a forma canônica de Brunovsky.

$$\begin{aligned} \dot{x}_c(x) &= A_c x_c(x) + B_c v_c, \\ y &= C_c x_c(x), \end{aligned} \tag{2.35}$$

em que

$$A_c x_c(x) = \mathcal{L}_f x_c(x) + \mathcal{L}_g x_c(x)\alpha_c(x), \tag{2.36}$$

$$B_c = \mathcal{L}_g x_c(x)\beta_c(x). \tag{2.37}$$

Deve-se notar que (2.35) está representada em termos das variáveis de estado originais x. Para transformar para as novas variáveis de estado, basta utilizar $x = \phi_c^{-1}(x_c)$.

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, pois é mais complicado 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, consequentemente, o sistema em malha fechada não é, em geral, robusto.

A próxima seção, apresenta a técnica de linearização robusta, que pode ser empregada com o objetivo de contornar os problemas citados.

2.3.5 Linearização Robusta

Nesta Seção, apresenta-se a técnica de realimentação linearizante robusta. Essa técnica, proposta em (Guillard, 2000), tem por base uma transformação não linear, derivada da técnica de realimentação linearizante clássica, que lineariza o sistema não linear transformando-o em um sistema na forma canônica de Brunovsky. A linearização por realimentação robusta é realizada de maneira que o sistema resultante, após a transformação, corresponda à aproximação linear do sistema (2.10) em torno de um ponto de operação x_0. Em (Guillard, 2000) apresenta-se um resultado onde se prova por W-estabilidade (Bourles e Colledani, 1995; Bourles, 2000) que a linearização robusta é robusta para pequenas variações paramétricas, pois possui a característica de manter mais informações sobre o sistema. A técnica de linearização robusta, equivalentemente à clássica, visa encontrar uma transformação de estados $\phi_r(x) = x_r$ e uma transformação de entrada $u_r(x)$, porém que transformem o sistema não linear na sua aproximação linear em torno de um ponto de operação, ao invés de colocá-lo na forma canônica de Brunovsky como ocorre na técnica de linearização clássica.

O procedimento para desenvolvimento da realimentação robusta é uma extensão dos métodos apresentados na Seção 2.3.4.

Seja a aproximação linear do sistema (2.10) em torno do ponto de equilíbrio x_0, escolhido, sem perda de generalidade, como $x_0 = 0$, dada por

$$\begin{aligned} \dot{x}_r &= A_r x_r + B_r v_r, \\ y &= C_r x_r, \end{aligned} \tag{2.38}$$

com

$$A_r = \frac{\partial f(x)}{\partial x}\Big|_{x=0}, \tag{2.39}$$

$$B_r = g(0), \tag{2.40}$$

$$C_r = \frac{\partial \lambda(x)}{\partial x}\Big|_{x=0}, \tag{2.41}$$

no qual $x_r \in \mathbb{R}^n$ representa o estado e $v_r \in \mathbb{R}^m$ é a entrada de controle linear.

Supondo que o par (A_r,B_r) é controlável, então, segundo (Guillard, 2000) existem matrizes $T^{n\times n}$, $L^{m\times n}$ e $R^{m\times m}$ tais que as seguintes igualdades sejam verificadas:

$$\begin{aligned} x_c &= Tx_r, \\ A_c &= T(A_r - B_rRL)T^{-1}, \\ B_c &= TB_rR, \\ C_c &= C_rT^{-1}, \end{aligned} \tag{2.42}$$

sendo T e R matrizes não singulares.

Substituindo-se as relações descritas por (2.42) em (2.35) e pré-multiplicando-se por T^{-1} obtém-se:

$$\begin{aligned} \dot{x}_r &= A_rx_r + B_r(Rv_c - RLx_r), \\ y &= C_rx_r. \end{aligned} \tag{2.43}$$

Comparando-se a equação (2.38) com a equação (2.43), conclui-se que:

$$v_r = Rv_c - RLx_r. \tag{2.44}$$

A partir da equação (2.44), determina-se então, a lei de controle v_c, que transforma o sistema na forma de Brunovsky (2.35), na aproximação linear do sistema não linear em torno de um ponto de operação (2.38), conforme equação (2.45).

$$v_c = R^{-1}v_r + LT^{-1}x_c. \tag{2.45}$$

Pode-se unir o resultado da transformação não linear e da transformação linear para se obter a transformação de estados $\phi_r(x)$ e a transformação de entrada $u_r(x)$ referentes à técnica de realimentação linearizante robusta.

Da equação (2.42), tem-se que o difeomorfismo pode ser calculado como

$$\phi_r(x) = T^{-1}\phi_c(x). \tag{2.46}$$

A transformação de entrada $u_r(x,v_r)$ é calculada pela substituição da equação (2.45) na equação (2.26), resultando em:

$$u_r(x,v_r) = \alpha_r(x) + \beta_r(x)v_r, \tag{2.47}$$

em que

$$\alpha_r(x) = \alpha_c(x) + \beta_c(x)LT^{-1}\phi_c(x), \tag{2.48}$$

$$\beta_r(x) = \beta_c(x)R^{-1}. \tag{2.49}$$

Resta agora determinar as matrizes T, R e L de forma a garantir que a transformação de estados (2.46) e a transformação de entrada (2.47) transformem o sistema não linear em sua aproximação linear em torno de um ponto de operação.

Considere a aplicação da transformação de entrada (2.47) sobre o sistema não linear (2.10):

$$\dot{x} = f(x) + g(x)\alpha_r(x) + g(x)\beta_r(x)v_r. \tag{2.50}$$

Aplicando (2.39) e (2.40) sobre a equação (2.50), as matrizes A_r e B_r são encontradas como:

$$A_r = \frac{\partial f(x)}{\partial x}\Big|_{x=x_0} + \frac{\partial g(x)}{\partial x}\alpha_r(x)\Big|_{x=x_0} + \frac{\partial \alpha_r(x)}{\partial x}g(x)\Big|_{x=x_0}, \tag{2.51}$$

$$B_r = g(x)\beta_r(x)\Big|_{x=x_0}. \tag{2.52}$$

Para que o projeto das transformações de estado (2.46) e de entrada (2.47) cumpram o objetivo de transformar o sistema não linear (2.10) na sua aproximação linear em torno de um ponto de operação, é necessário que as equações (2.39) e (2.51) sejam iguais, assim como também as equações (2.40) e (2.52). Logo, comparando tais equações, é possível concluir que é necessário que sejam atendidas as seguintes condições:

$$\begin{aligned} \alpha_r(x_0) &= 0, \\ \frac{\partial \alpha_r(x)}{\partial x}\Big|_{x=x_0} &= 0, \\ \beta_r(x_0) &= I. \end{aligned} \tag{2.53}$$

O conjunto de equações (2.53) mostra que os termos $\alpha_r(x)$ e $\beta_r(x)$ não devem influenciar na linearização do sistema (2.10) em torno do ponto de operação x_0. O mesmo deve acontecer para o difeomorfismo $\phi_r(x)$, dessa forma é necessário que:

$$\frac{\partial \phi_r(x)}{\partial x}\Big|_{x=x_0} = I. \tag{2.54}$$

Relacionando as equações (2.46) e (2.54), obtém-se:

$$\frac{\partial \phi_r(x)}{\partial x}\Big|_{x=x_0} = T^{-1}\frac{\partial \phi_c(x)}{\partial x}\Big|_{x=x_0} = I, \tag{2.55}$$

de onde pode-se calcular a matriz T conforme equação (2.56):

$$T = \frac{\partial \phi_c(x)}{\partial x}\Big|_{x=x_0}. \tag{2.56}$$

Calcula-se agora a matriz R, relacionando as condições descritas na última equação do conjunto de equações (2.53) e a equação (2.49).

$$\beta_r(x_0) = \beta_c(x_0)R^{-1} = I. \tag{2.57}$$

De (2.57) e de (2.28) obtém-se:

$$R = \beta_c(x_0) = E^{-1}(x). \tag{2.58}$$

Por fim, calcula-se a matriz L relacionando a condição descrita pela primeira equação do conjunto de equações (2.53) e a equação (2.48), conforme segue:

$$\begin{aligned} \frac{\partial \alpha_r(x)}{\partial x}\Big|_{x=x_0} &= \frac{\partial}{\partial x}(\alpha_c(x) + \beta_c(x)LT^{-1}x_c)\Big|_{x=x_0} \\ &= \left[\frac{\partial \alpha_c(x)}{\partial x} + \frac{\partial \beta_c(x)}{\partial x}LT^{-1}x_c + \beta_c(x)LT^{-1}\frac{\partial x_c}{\partial x}\right]\Big|_{x=x_0} = 0. \end{aligned} \tag{2.59}$$

sabendo que $x_c = \phi_c(x)$ e que $\phi_c(x_0) = 0$ e substituindo-se a equação (2.56) no equacionamento (2.59), obtém-se:

$$\frac{\partial \alpha_c(x)}{\partial x}\Big|_{x=x_0} + L\beta_c(x)\Big|_{x=x_0} = 0, \tag{2.60}$$

o que resulta em

$$L = -\beta_c^{-1}(x_0)\frac{\partial \alpha_c(x)}{\partial x}\Big|_{x=x_0}. \tag{2.61}$$

Utilizando a equação (2.28), a equação (2.61) pode ser escrita como:

$$L = -E(x_0)\frac{\partial \alpha_c(x)}{\partial x}\Big|_{x=x_0}. \tag{2.62}$$

Todos os parâmetros necessários para projetar o controle de realimentação linearizante robusta foram determinados. Como pode-se observar pelo equacionamento dessa técnica, existe a necessidade de acesso a informação de todos os estados do sistema. Essa necessidade é uma desvantagem da técnica, porque em geral não é possível medir todos os estados. Porém, nas situações em que os estados não estão disponíveis, pode-se contornar essa dificuldade utilizando um observador de estados.

Portanto, conclui-se que esse método é uma extensão da linearização por realimentação clássica, a qual é combinada à transformação de base dada por (2.42). Com essa modificação, são eliminadas as desvantagens citadas anteriormente.

Evita-se a perda do sentido físico dos estados do sistema, pois o vetor de estados x_r é semelhante a x. Além disso, as não linearidades do sistema são substituídas por suas aproximações lineares, em lugar de serem simplesmente canceladas.

Essa linearização por realimentação é chamada robusta pois, como o sistema resultante após a transformação corresponde à aproximação linear do sistema original em torno de um ponto de operação, o comportamento do sistema linearizado é semelhante ao comportamento do sistema original. Intuitivamente, presume-se que isto implica em maior robustez do sistema em malha fechada, pois em caso de variações de parâmetros ou incertezas de modelo não estruturadas os dois sistemas apresentam respostas parecidas. As propriedades do controlador linear utilizado serão então provavelmente conservadas.

Capítulo 3

Uma Contribuição para o Posicionamento das Rodas do Robô F180

Este Capítulo apresenta o desenvolvimento do modelo dinâmico de um robô omnidirecional de quatro rodas. Esse modelo é constituído pelo modelo da base móvel do robô e pelo modelo dos motores de corrente contínua utilizados, tendo como estados as velocidades do robô nas direções x e y e a velocidade angular. Um estudo detalhado sobre o posicionamento das rodas é desenvolvido com o objetivo de melhorar o desempenho do robô.

3.1 Introdução

Aspectos de caráter mecânico, elétrico e a disposição física das partes que compõem um robô são características relevantes na escolha e na implementação de seus controladores. A capacidade de um robô locomover-se e as equações que regem seu movimento são influenciadas pela sua geometria e pela sua dinâmica. Aspectos de caráter geométrico são considerados no modelo cinemático, enquanto os de caráter dinâmico, no modelo dinâmico.

Um modelo cinemático descreve o movimento do robô em função das velocidades das rodas sem considerar as forças que atuam sobre o mesmo. Já um modelo dinâmico descreve as relações dinâmicas entre as coordenadas de postura e a orientação do robô e os torques desenvolvidos pelos seus atuadores. O modelo dinâmico pode ser formulado segundo o formalismo de Newton-Euler (Watanabe et al., 1998; Carter et al., 2001; Yamada et al., 2001; Kottke, 2005; Liu et al., 2008) ou o formalismo de Euler-Lagrange (Schroeder et al., 2005). De acordo com Yamada et al. (2001), enquanto as equações de Newton tratam cada corpo rígido separadamente e modelam explicitamente as restrições de cada corpo através das forças que atuam sobre esses, Lagrande e d'Alembert fornecem um procedimento sistemático para eliminar restrições mecânicas, através de um sistema de equações geral. Neste trabalho, foi adotado o formalismo de Newton-Euler, devido a uma vasta literatura sobre modelos dinâmicos de robôs semelhantes (Watanabe et al., 1998; Carter et al., 2001; Yamada et al., 2001; Oubbati, 2006; Franco, 2006a; Liu et al., 2008).

As expressões (3.1) e (3.2) apresentam as equações de Newton.

$$\sum F = ma, \tag{3.1}$$

$$\sum M_o = J_w \alpha, \tag{3.2}$$

em que $\sum F$ e $\sum M_o$ são respectivamente, as resultantes das forças e momentos que atuam sobre um corpo, m a sua massa, a a sua aceleração linear, J_w sua matriz de inércia e α sua aceleração angular.

3.2 Modelo Geométrico do Robô

Robôs móveis com rodas possuem boa manobrabilidade, o que faz com que eles sejam largamente utilizados em produção e na vida diária das pessoas (Cao et al., 2007b). Os robôs móveis omnidirecionais possuem movimentos com três graus de liberdade em um plano bi-dimensional, de forma a realizar movimentos de translação e rotação simultaneamente ao longo de uma direção arbitrária. Utilizando esse tipo de sistema, o robô economiza tempo, pois elimina a necessidade de rotacionar antes de se mover de um ponto A até um ponto B como acontece nos automóveis. Devido a essa maior agilidade, robôs omnidirecionais têm sido utilizados em várias situações, como em cadeiras de rodas (Wada, 2005, 2009).

Vários grupos de pesquisa têm dedicado atenção a esse tipo de robôs (Dickerson e Lapin, 1991; Watanabe et al., 1998; Jung et al., 2000; Williams et al., 2002; Ould-Khessal, 2005). Veja por exemplo (Watanabe et al., 1998) que apresenta a modelagem dinâmica de um robô móvel omnidirecional (RMO) de três rodas baseada em variáveis de estado, na qual assume-se que as rodas são simetricamente distribuídas na periferia do robô. Em (Williams et al., 2002), esse modelo é expandido para uma distribuição assimétrica de rodas, levando em consideração o deslizamento das rodas sobre o solo. Segundo Cao et al. (2007b), os RMO's de três rodas podem ter problemas de estabilidade devido a área triangular de contato com o solo, especialmente quando o centro de gravidade está distante do solo ou quando o tamanho do robô é pequeno. Consequentemente, é desejável o uso de robôs de quatro rodas quando a estabilidade é de grande interesse, principalmente em se tratando de competições como a *Robocup*, que é o caso estudado neste trabalho.

O modelo geométrico do robô estudado neste trabalho é mostrado na Figura 3.1. Esse modelo foi desenvolvido com base nos modelos apresentados em (Watanabe et al., 1998; Franco, 2006a; Rojas e Förster, 2006). Nessa figura, existem 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. No modelo geométrico, θ é o ângulo de rotação do robô em relação ao sistema global de referência, $\Psi = \theta + \gamma$ é o azimute do robô no sistema global de referência, sendo que γ é o ângulo entre X_m e f_R, ou seja, é o azimute no sistema móvel de referência e φ_i, $i = 1, 2, 3$ *e* 4 são os ângulos que os eixos das rodas fazem com o eixo X_m. Como o ângulo de rotação do robô é representado por $\theta = \Psi - \gamma$, percebe-se que mesmo que o azimute do robô mude arbitrariamente, o robô pode realizar movimentos de translação sem rotacionar (Watanabe et al., 1998).

Quando os quatro motores estão ativados, obtém-se quatro forças de tração $\mathbf{f}_1$, $\mathbf{f}_2$, $\mathbf{f}_3$ e $\mathbf{f}_4$ que criam uma força de translação e um binário de rotação. Cada força $\mathbf{f}_i$ é o torque do motor dividido pelo raio da roda. A força resultante $\mathbf{f}_R$ depende da posição exata em que as

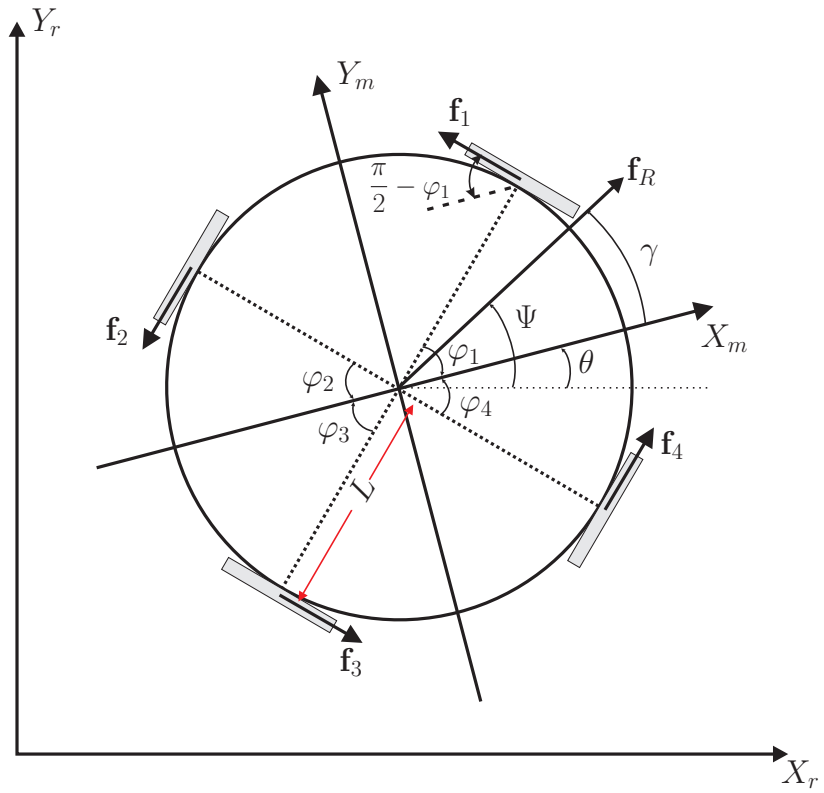


Figura 3.1: Modelo geométrico do robô relacionando o sistema de coordenadas móvel (X_m,Y_m) e o sistema de coordenadas global (X_r,Y_r). Veja texto na Seção 3.2 para descrição das variáveis.

rodas estão dispostas no robô e da direção de movimento do robô. A Seção 3.2.1 apresenta o desenvolvimento da matriz de acoplamento de forças, que é responsável por relacionar as forças e as acelerações desenvolvidas pelos atuadores.

3.2.1 Matriz de Acoplamento de Forças

Deseja-se analisar o movimento do robô ao longo das direções X_m e Y_m, dessa forma, considera-se a velocidade e aceleração instantâneas do robô em relação ao seu próprio sistema de referência, $S_m = [X_m \; Y_m]$.

Assumindo-se que o centro de massa do robô está localizado em seu centro geométrico e, utilizando-se a formulação de Newton-Euler (3.1), pode-se descrever sua aceleração linear a conforme equação (3.3).

$$a = \frac{1}{m}(\mathbf{f}_1 + \mathbf{f}_2 + \mathbf{f}_3 + \mathbf{f}_4), \tag{3.3}$$

em que m é a massa do robô. A aceleração angular α é encontrada a partir da equação de Newton (3.2) dada pela equação (3.4). Veja que α foi substituído por $\ddot{\theta}$ para adequar a notação com a utilizada no modelo geométrico apresentado na Figura 3.1 e

$$\ddot{\theta} = \frac{L}{J_w}(f_1 + f_2 + f_3 + f_4), \tag{3.4}$$

em que L é a distância do centro da roda até o centro do robô, J_w o momento de inércia do robô e f_i os módulos das forças $\mathbf{f}_i$, para $i = 1, \ldots, 4$.

A utilização dos módulos das forças é possível porque as forças são tangentes à periferia circular do robô e apontam para o mesmo sentido (Rojas e Förster, 2006). As forças f_1, f_2, f_3, f_4 podem ser positivas ou negativas de acordo com o sentido de rotação dos motores. A direção positiva é assumida como sendo aquela representada na Figura 3.1.

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

$$ma_x = -f_1 \sin\varphi_1 - f_2 \sin\varphi_2 + f_3 \sin\varphi_3 + f_4 \sin\varphi_4. \tag{3.5}$$

$$ma_y = f_1 \cos\varphi_1 - f_2 \cos\varphi_2 - f_3 \cos\varphi_3 + f_4 \cos\varphi_4. \tag{3.6}$$

Para um robô com distribuição de massa homogênea, $J_w = \frac{1}{2}mL^2$, para um robô com 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 $J_w = \sigma m L^2$, em que $0 < \sigma \leq 1$ representa a constante de distribuição de massa (Rojas e Förster, 2006).

Escrevendo as equações (3.4), (3.5) e (3.6) na forma matricial, tem-se:

$$\begin{bmatrix} a_x \\ a_y \\ \ddot{\theta} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} -\sin\varphi_1 & -\sin\varphi_2 & \sin\varphi_3 & \sin\varphi_4 \\ \cos\varphi_1 & -\cos\varphi_2 & -\cos\varphi_3 & \cos\varphi_4 \\ \frac{mL}{J_w} & \frac{mL}{J_w} & \frac{mL}{J_w} & \frac{mL}{J_w} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}. \tag{3.7}$$

Usando $J_w = \sigma m L^2$, a equação (3.7) pode ser escrita como

$$\begin{bmatrix} a_x \\ a_y \\ \ddot{\theta} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} -\sin\varphi_1 & -\sin\varphi_2 & \sin\varphi_3 & \sin\varphi_4 \\ \cos\varphi_1 & -\cos\varphi_2 & -\cos\varphi_3 & \cos\varphi_4 \\ \frac{1}{\sigma L} & \frac{1}{\sigma L} & \frac{1}{\sigma L} & \frac{1}{\sigma L} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}. \tag{3.8}$$

Pode-se ainda simplificar a equação (3.8) de forma a usar a mesma unidade para as acelerações linear e angular (m/s^2). Dessa forma, usa-se $L\ddot{\theta}$ no lugar de $\ddot{\theta}$. A nova expressão é mostrada na equação (3.9).

$$\begin{bmatrix} a_x \\ a_y \\ L\ddot{\theta} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} -\sin\varphi_1 & -\sin\varphi_2 & \sin\varphi_3 & \sin\varphi_4 \\ \cos\varphi_1 & -\cos\varphi_2 & -\cos\varphi_3 & \cos\varphi_4 \\ \frac{1}{\sigma} & \frac{1}{\sigma} & \frac{1}{\sigma} & \frac{1}{\sigma} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}. \tag{3.9}$$

Pela equação (3.9), pode-se notar que, conhecendo-se os torques associados aos motores, torna-se então simples encontrar as acelerações nas direções x e y, bem como a aceleração angular da periferia do robô.

Nota-se, que as forças f_i podem se cancelar, se por exemplo, $f_1 = f_3 = 1$ e $f_2 = f_4 = -1$, então o robô não se movimenta, pois as rodas estão trabalhando em sentidos opostos. Assume-se nesse caso que as rodas não deslizam e que todo torque é transferido para o robô.

É importante notar que a aceleração angular do robô depende de sua distribuição de massa σ. Dessa forma, esse é um ponto que deve ser considerado para melhorar o controle do robô. Neste trabalho é assumido que o robô possui distribuição de massa homogênea, ou seja, $J_w = \frac{1}{2}mL^2$. Dessa forma $\sigma = \frac{1}{2}$. Pela análise da equação (3.9), pode-se notar que a escolha adequada dos ângulos φ_i, $i = 1, 2, 3, 4$, pode melhorar de forma significativa o desempenho do robô, uma vez que é possível otimizar a distribuição das forças dos atuadores.

3.2.2 Modelagem Cinemática

Neste trabalho, será utilizado o modelo cinemático de postura. Para a formulação das equações cinemáticas, é necessário conhecer as relações entre as componentes de velocidade do robô em alguns pontos de referência, adotando, em cada referencial, um sistema de coordenadas cartesianas.

Na formulação do modelo cinemático foram feitas algumas considerações: só existe um ponto de contato da roda com o solo e a mesma não sofre nenhum deslizamento; o plano da roda permanece sempre na vertical e não existe nenhuma deformação das partes que compõem o robô. (Williams et al., 2002) formulam o modelo cinemático para um RMO de três rodas levando em consideração o deslizamento das rodas sobre o plano de contato, já (Cao et al., 2007b) estendem esse resultado para um RMO de quatro rodas.

Conforme citado na Seção 3.2, o modelo geométrico do robô possui dois sistemas de coordenadas. O sistema de coordenadas móvel S_m e o sistema de coordenadas global S_r.

A posição e a orientação do centro de massa do robô em relação ao sistema de coordenadas S_r podem ser representadas pela equação (3.10)

$$\xi_r = \begin{bmatrix} X_r \\ Y_r \\ \theta \end{bmatrix}, \tag{3.10}$$

em que ξ_r representa o vetor de postura do robô, sendo X_r e Y_r, respectivamente, a posição do robô sobre os eixos X_r e Y_r do sistema S_r e θ a sua orientação.

De forma análoga, pode-se escrever a equação (3.11) para a posição e orientação do centro de massa do robô em relação ao sistema de coordenadas S_m

$$\xi_m = \begin{bmatrix} X_m \\ Y_m \\ \theta \end{bmatrix}, \tag{3.11}$$

em que ξ_m representa o vetor de postura do robô, sendo X_m e Y_m, respectivamente a posição do robô sobre os eixos X_m e Y_m do sistema S_m e θ a sua orientação.

A relação das componentes de velocidade do centro de massa do robô entre o sistema de referência S_r e o sistema móvel S_m é dada pela equação (3.12)

$$\dot{\xi}_r = R(\theta)\dot{\xi}_m, \tag{3.12}$$

em que $\dot{\xi}_r = [\dot{X}_r \ \dot{Y}_r \ \dot{\theta}]^T$ e $\dot{\xi}_m = [\dot{X}_m \ \dot{Y}_m \ \dot{\theta}]^T$ são as componentes de velocidade do centro de massa do robô sobre os sistemas S_r e S_m respectivamente. $R(\theta)$ é a matriz ortogonal de rotação do sistema S_m em relação ao sistema S_r, dada pela equação (3.13).

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.13}$$

Nesse ponto, é necessário relacionar a velocidade do robô $\dot{\xi}_m$ com as velocidades impostas por cada uma das rodas. Para isso, agrupa-se as velocidades das rodas em um vetor $[v_1 \ v_2 \ v_3 \ v_4]^T$, em que v_i, $i = 1, \ldots, 4$, são as velocidades lineares das rodas.

Cada roda provê tração na direção normal ao eixo do robô e paralela ao piso. A soma das forças resulta em movimentos de translação e rotação. Para que o robô tenha uma velocidade nas direções X_m e Y_m com relação ao sistema de coordenadas $\dot{\xi}_m$, cada roda deve exercer parte de uma das duas componentes de velocidade (Rojas e Förster, 2006).

Se o robô move-se conforme o vetor $\dot{\xi}_m = [1 \ 0 \ 0]^T$, isto significa que ele está movendo-se na direção X_m sem rotacionar. Quando o robô move-se com velocidade 1 para a direita, as rodas giram com velocidade $-\sin\varphi_i$ e os roletes (pequenas rodas dispostas na periferia da roda principal conforme Figura 2.3) giram com velocidade $\cos\varphi_i$. A Figura 3.2 apresenta essa decomposição para a roda 1.

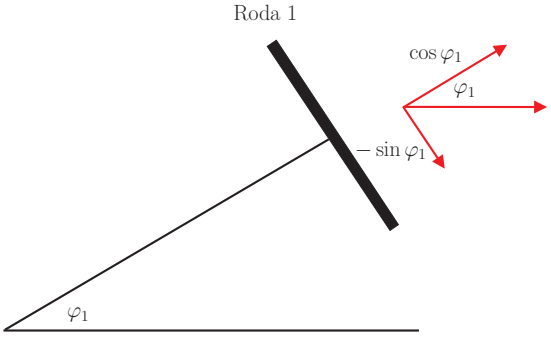


Figura 3.2: Rotação da roda e dos roletes quando o robô move-se para a direita com velocidade unitária. A roda move-se com velocidade $-\sin\varphi_1$ e os roletes com velocidade $\cos\varphi_1$. (Fonte: (Rojas e Förster, 2006)).

Utilizando-se a convenção que a direção de rotação positiva é aquela dada pela Figura 3.1, pode-se descrever a relação entre a velocidade do robô e as velocidades das rodas conforme equação (3.14).

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} -\sin\varphi_1 & \cos\varphi_1 & L \\ -\sin\varphi_2 & -\cos\varphi_2 & L \\ \sin\varphi_3 & -\cos\varphi_3 & L \\ \sin\varphi_4 & \cos\varphi_4 & L \end{bmatrix} \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}. \tag{3.14}$$

A equação (3.14) descreve a decomposição das velocidades linear e angular do robô no sistema de referência móvel em função das respectivas velocidades lineares das rodas; no entanto, é necessário representar essas velocidades em função das velocidades angulares dos motores de

forma que seja possível acrescentar a dinâmica dos atuadores. As velocidades lineares v_i, podem ser transformadas em velocidades angulares usando a relação, $v_i = \phi_i r$, sendo ϕ_i e r a velocidade angular da roda i e o raio da roda respectivamente. Como cada roda está acoplada ao motor através de uma caixa de redução, as velocidades angulares dos motores podem ser dadas por $\omega_{m_i} = \phi_i n$, sendo n o coeficiente de redução da caixa de redução. Dessa forma, a equação (3.14) pode então ser reescrita em função das velocidades angulares dos motores conforme (3.15).

$$\begin{bmatrix} \omega_{m_1} \\ \omega_{m_2} \\ \omega_{m_3} \\ \omega_{m_4} \end{bmatrix} = \frac{n}{r} D \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}. \tag{3.15}$$

em que D é dada por

$$D = \begin{bmatrix} -\sin\varphi_1 & \cos\varphi_1 & L \\ -\sin\varphi_2 & -\cos\varphi_2 & L \\ \sin\varphi_3 & -\cos\varphi_3 & L \\ \sin\varphi_4 & \cos\varphi_4 & L \end{bmatrix}. \tag{3.16}$$

A equação (3.15) é conhecida como modelo cinemático inverso. O modelo cinemático direto é mostrado na equação (3.17).

$$\begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} = \frac{r}{n} D^{+} \begin{bmatrix} \omega_{m_1} \\ \omega_{m_2} \\ \omega_{m_3} \\ \omega_{m_4} \end{bmatrix}. \tag{3.17}$$

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

Pode ser encontrada em (Angeles, 2003; Braunl, 2003; Siegwart e Nourbakhsh, 2004) uma abordagem mais ampla sobre cinemática de diversas classes de robôs móveis.

3.2.3 Modelagem Dinâmica

O modelo da base móvel do robô desenvolvido neste trabalho foi formulado segundo as leis de Newton, utilizando a equação (3.1) para a análise do robô em movimentos de translação e a equação (3.2) para a análise do robô em movimentos de rotação. Ao modelo da base móvel foi, então, incluído o modelo dinâmico dos atuadores, gerando o modelo dinâmico do robô. As forças que atuam no robô são geradas pelos torques dos atuadores, que sofre reações das forças de atrito viscoso do contato entre as partes mecânicas do motor. Modelo semelhante ao apresentado neste trabalho pode ser encontrado em (Cao et al., 2007b), no entanto, o modelo apresentado neste trabalho deixa os parâmetros e os ângulos das rodas explícitos nas matrizes do sistema, de forma que possa ser facilmente adaptado a outras configurações de robôs.

Modelo Dinâmico da Base Móvel

O modelo dinâmico do robô foi desenvolvido com base na Figura 3.1 e nas equações (3.1) e (3.2). Utilizando a mesma notação da Seção 3.2.2 e a equação (3.1), pode-se escrever

$$F_r = M_m \ddot{\xi}_r, \tag{3.18}$$

$$F_r = R(\theta) f_m, \tag{3.19}$$

em que, $F_r = [F_x \ F_y \ \sum M_o]^T$ representa as forças no sistema de coordenadas global, $f_m = [f_x \ f_y \ \sum M_o]^T$ representa as forças no sistema de coordenadas móvel e M_m é uma matriz diagonal contendo a massa e o momento de inércia do robô dada pela equação

$$M_m = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J_w \end{bmatrix}. \tag{3.20}$$

Substituindo-se a derivada temporal de (3.12) e a equação (3.19) em (3.18), tem-se:

$$M_m(R(\theta)^T \dot{R}(\theta)\dot{\xi}_m + \ddot{\xi}_m) = f_m. \tag{3.21}$$

Substituindo-se (3.20), $R(\theta)^T$, $\dot{R}(\theta)$, $\dot{\xi}_m$, $\ddot{\xi}_m = [\ddot{X}_m \ \ddot{Y}_m \ \ddot{\theta}]^T$ e f_m em (3.21), tem-se:

$$\begin{bmatrix} f_x \\ f_y \\ \sum M_o \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J_w \end{bmatrix} \left(\begin{bmatrix} 0 & -\dot{\theta} & 0 \\ \dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} \right). \tag{3.22}$$

Assim, conforme (3.22), as propriedades dinâmicas do robô podem ser escritas como:

$$\begin{aligned} f_x &= m(\ddot{X}_m - \dot{\theta}\dot{Y}_m), \\ f_y &= m(\ddot{Y}_m + \dot{\theta}\dot{X}_m), \\ \sum M_o &= J_w \ddot{\theta}. \end{aligned} \tag{3.23}$$

Para encontrar o modelo dinâmico do robô, deve-se escrever f_x, f_y e $\sum M_o$ em função de f_i, $i = 1,2,3,4$, em que f_i representa a força referente à roda i, conforme Figura 3.1.

A partir da decomposição vetorial das forças f_i da Figura 3.1, pode-se escrever f_x, f_y e $\sum M_o$ conforme equação (3.24).

$$\begin{aligned} f_x &= -f_1 \sin\varphi_1 - f_2 \sin\varphi_2 + f_3 \sin\varphi_3 + f_4 \sin\varphi_4, \\ f_y &= f_1 \cos\varphi_1 - f_2 \cos\varphi_2 - f_3 \cos\varphi_3 + f_4 \cos\varphi_4, \\ \sum M_o &= (f_1 + f_2 + f_3 + f_4)L. \end{aligned} \tag{3.24}$$

Então, substituindo-se (3.24) em (3.23) e isolando-se as componentes de aceleração resultantes, o modelo dinâmico da base móvel do robô pode ser escrito:

$$\ddot{X}_m = \frac{-f_1 \sin\varphi_1 - f_2 \sin\varphi_2 + f_3 \sin\varphi_3 + f_4 \sin\varphi_4}{m} + \dot{\theta}\dot{Y}_m, \tag{3.25}$$

$$\ddot{Y}_m = \frac{f_1 \cos\varphi_1 - f_2 \cos\varphi_2 - f_3 \cos\varphi_3 + f_4 \cos\varphi_4}{m} - \dot{\theta}\dot{X}_m, \tag{3.26}$$

$$\ddot{\theta} = \frac{(f_1 + f_2 + f_3 + f_4)L}{J_w}. \tag{3.27}$$

As equações (3.25), (3.26) e (3.27) podem ser escritas na forma matricial como:

$$M_m \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} + M_\theta \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} = M_f \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}, \tag{3.28}$$

em que M_m é dada pela equação (3.20) e M_θ e M_f são dadas por

$$M_\theta = \begin{bmatrix} 0 & -\dot{\theta}m & 0 \\ \dot{\theta}m & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.29}$$

$$M_f = \begin{bmatrix} -\sin\varphi_1 & -\sin\varphi_2 & \sin\varphi_3 & \sin\varphi_4 \\ \cos\varphi_1 & -\cos\varphi_2 & -\cos\varphi_3 & \cos\varphi_4 \\ L & L & L & L \end{bmatrix}. \tag{3.30}$$

Modelo dos Atuadores

No robô estudado neste trabalho, os atuadores são motores de corrente contínua. O modelo dinâmico de um motor de corrente contínua pode ser descrito por uma equação elétrica e uma mecânica conforme equações (3.31) e (3.32), respectivamente:

$$u = L_a \frac{di_a}{dt} + R_a i_a + e, \tag{3.31}$$

$$J_o \dot{\omega}_m + b_o \omega_m + \frac{r}{n} f = \tau, \tag{3.32}$$

sendo

$$\tau = k_t i_a, \tag{3.33}$$

$$e = k_{em} \omega_m, \tag{3.34}$$

em que L_a, R_a, i_a, J_o, b_o, τ, ω_m e u representam a indutância, a resistência, a corrente de armadura, o momento de inércia, a constante de atrito viscoso, o torque, a velocidade e a tensão do motor, respectivamente, e r e n o raio da roda e a constante de redução, respectivamente.

Como a constante de tempo elétrica do motor é muito pequena em relação à constante de tempo mecânica (Cao et al., 2007b; Liu et al., 2008), pode-se desprezar a dinâmica do circuito elétrico do motor. Com isso, $\frac{di_a}{dt} = 0$, então a equação (3.31) pode ser escrita

$$u = R_a i_a + k_{em}\omega_m. \tag{3.35}$$

Substituindo-se (3.35) e (3.33) em (3.32), o modelo dinâmico do motor pode ser escrito

$$u = \frac{R_a}{k_t} J_o \dot{\omega_m} + \frac{R_a}{k_t} b_o \omega_m + \frac{R_a}{k_t}\frac{r}{n} f + k_{em}\omega_m. \tag{3.36}$$

Em função de f, a equação (3.36) pode ser escrita como:

$$f = \frac{k_t}{R_a}\frac{n}{r} u - \frac{n}{r} J_o \dot{\omega}_m - \frac{n}{r} b_o \omega_m - \frac{k_t}{R_a}\frac{n}{r} k_{em}\omega_m. \tag{3.37}$$

Os motores são acionados por uma largura de pulso $\rho = \frac{u}{V_{ref}}$, considerando modulação PWM, então o modelo dinâmico do motor pode ser dado por

$$f = \frac{k_t}{R_a}\frac{n}{r}\rho V_{ref} - \frac{n}{r} J_o \dot{\omega}_m - \frac{n}{r}\left(b_o + \frac{k_t}{R_a} k_{em}\right)\omega_m. \tag{3.38}$$

A equação (3.38) pode agora ser utilizada para representar a dinâmica dos atuadores do robô, dessa forma, para os quatro motores, pode-se escrevê-la na forma matricial

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = -\frac{n}{r} J_o \begin{bmatrix} \dot{\omega}_{m_1} \\ \dot{\omega}_{m_2} \\ \dot{\omega}_{m_3} \\ \dot{\omega}_{m_4} \end{bmatrix} - \frac{n}{r}\left(b_o + \frac{k_t}{R_a} k_{em}\right) \begin{bmatrix} \omega_{m_1} \\ \omega_{m_2} \\ \omega_{m_3} \\ \omega_{m_4} \end{bmatrix} + \frac{n}{r}\frac{k_t}{R_a} V_{ref} \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix}. \tag{3.39}$$

Modelo Dinâmico da Base Móvel com Inclusão dos Atuadores

Após o desenvolvimento do modelo matemático que representa a dinâmica da base móvel do robô e do modelo dinâmico dos atuadores, pode-se então uní-los para encontrar o modelo dinâmico da base móvel com inclusão dos atuadores.

Substituindo (3.15) e sua primeira derivada temporal em (3.39), tem-se:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = -\frac{n^2}{r^2} J_o D \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} - \frac{n^2}{r^2}\left(b_o + \frac{k_t}{R_a} k_{em}\right) D \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + \frac{n}{r}\frac{k_t}{R_a} V_{ref} \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix}. \tag{3.40}$$

Para simplificar as operações daqui em diante, define-se:

$$\beta = \frac{n}{r}\frac{k_t}{R_a} V_{ref}, \tag{3.41}$$

$$\chi = \frac{n^2}{r^2} J_o, \tag{3.42}$$

$$\psi = \frac{n^2}{r^2}\left(b_o + \frac{k_t}{R_a}k_{em}\right). \tag{3.43}$$

Dessa forma, a equação (3.40) pode ser escrita:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = -\chi D \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} - \psi D \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + \beta \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix}. \tag{3.44}$$

Substituindo-se (3.44) em (3.28) tem-se:

$$M_m \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} + M_\theta \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} = M_f \left(-\chi D \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} - \psi D \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + \beta \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} \right). \tag{3.45}$$

Substituindo-se (3.20), (3.29), (3.30) e (3.16) em (3.45), e após algumas manipulações matemáticas, tem-se

$$M_1 \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} = M_2 \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + M_3 \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix}, \tag{3.46}$$

$$\begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} = M_1^{-1} M_2 \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + M_1^{-1} M_3 \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix}, \tag{3.47}$$

em que as matrizes M_1, M_2 e M_3 são dadas pelas equações (3.48), (3.49) e (3.50) respectivamente, e M_1^{-1} é a inversa de M_1:

$$M_1 = \begin{bmatrix} -(\chi e_1 + m) & -\chi e_2 & -\chi e_3 \\ -\chi e_2 & -(\chi e_4 + m) & -\chi e_5 \\ -\chi e_3 & -\chi e_5 & -(4L^2\chi + J_w) \end{bmatrix}, \tag{3.48}$$

$$M_2 = \begin{bmatrix} \psi e_1 & (\psi e_2 - \dot{\theta} m) & \psi e_3 \\ (\psi e_2 + \dot{\theta} m) & \psi e_4 & \psi e_5 \\ \psi e_3 & \psi e_5 & 4L^2\psi \end{bmatrix}, \tag{3.49}$$

$$M_3 = \begin{bmatrix} \beta \sin\varphi_1 & \beta \sin\varphi_2 & -\beta \sin\varphi_3 & -\beta \sin\varphi_4 \\ -\beta \cos\varphi_1 & \beta \cos\varphi_2 & \beta \cos\varphi_3 & -\beta \cos\varphi_4 \\ -\beta L & -\beta L & -\beta L & -\beta L \end{bmatrix}, \tag{3.50}$$

sendo

$$
\begin{aligned}
e_1 &= \sin^2\varphi_1 + \sin^2\varphi_2 + \sin^2\varphi_3 + \sin^2\varphi_4, \\
e_2 &= -\sin\varphi_1\cos\varphi_1 + \sin\varphi_2\cos\varphi_2 - \sin\varphi_3\cos\varphi_3 + \sin\varphi_4\cos\varphi_4, \\
e_3 &= -L\sin\varphi_1 - L\sin\varphi_2 + L\sin\varphi_3 + L\sin\varphi_4, \\
e_4 &= \cos^2\varphi_1 + \cos^2\varphi_2 + \cos^2\varphi_3 + \cos^2\varphi_4, \\
e_5 &= L\cos\varphi_1 - L\cos\varphi_2 - L\cos\varphi_3 + L\cos\varphi_4.
\end{aligned}
$$

A equação (3.47) representa o modelo dinâmico da base móvel do robô com inclusão dos atuadores. Esse modelo pode então ser utilizado para realizar as simulações de um robô omnidirecional de quatro rodas da categoria F180. O ponto importante desse modelo é que ele é genérico para robôs de quatro rodas, ou seja, ele não foi desenvolvido para uma configuração de rodas específica; dessa forma, ele pode ser adequado para qualquer configuração, bastando substituir os valores dos ângulos φ_i, $i = 1, 2, 3, 4$.

3.3 Análise do Posicionamento das Rodas

Analisando a Figura 3.1 e as equações (3.9) e (3.17), pode-se perceber que a forma como as rodas são distribuídas na periferia do robô pode influenciar na distribuição de forças e velocidades dos atuadores. Será mostrado nessa seção como a escolha dos ângulos φ_1, φ_2, φ_3 e φ_4 pode melhorar o desempenho do robô. Por exemplo, pode-se construir um robô otimizado para atuar como goleiro, atacante ou zagueiro. Embora a escolha desses ângulos tenha grande influência no desempenho do robô, essa análise ainda é pouco abordada na literatura. Veja por exemplo (Hirata et al., 2007), que apresenta uma comparação entre duas configurações de distribuição das rodas de um robô omnidirecional de três rodas do tipo passivo. O objetivo desta seção é apresentar uma análise mais completa sobre a influência dos ângulos de distribuição das rodas, de forma a obter informações suficientes para otimizar sua escolha.

Na literatura, diversas configurações para a distribuição angular das rodas podem ser encontradas. Em algumas, essa distribuição é simétrica, ou seja, $\varphi_1 = \varphi_2 = \varphi_3 = \varphi_4$. Veja por exemplo (Rojas e Förster, 2006; Purwin e Andrea, 2006) que utilizam distribuição simétrica com ângulos iguais a 45°, (Jr. et al., 2010) em que os ângulos são de 33° e (Cao et al., 2007b) que utiliza uma configuração com ângulos de 30°. Em outras, no entanto, a distribuição não apresenta simetria. Em (Laue et al., 2010) é utilizada uma configuração com $\varphi_1 = \varphi_2 = 37°$ e $\varphi_3 = \varphi_4 = 45°$, em (Pranggonoh et al., 2010) os ângulos são $\varphi_1 = \varphi_2 = 35°$ e $\varphi_3 = \varphi_4 = 45°$ e em (Fialho et al., 2010; Sharbafi et al., 2010) os ângulos são $\varphi_1 = \varphi_2 = 33°$ e $\varphi_3 = \varphi_4 = 45°$.

Percebe-se que não existe uma regra para escolha desses ângulos, sendo que são geralmente escolhidos com base nas restrições de espaço físico para acomodar os sistemas de chute e drible (Fialho et al., 2010). A Figura 3.3 apresenta a vista inferior de um robô omnidirecional de quatro rodas com ângulos $\varphi_1 = \varphi_2 = 33°$ e $\varphi_3 = \varphi_4 = 45°$. Nessa figura é possível perceber claramente que o espaço necessário para acomodação dos sistemas de chute e drible obriga a utilização de ângulos menores nas rodas dianteiras, caracterizando-se em uma distribuição assimétrica.

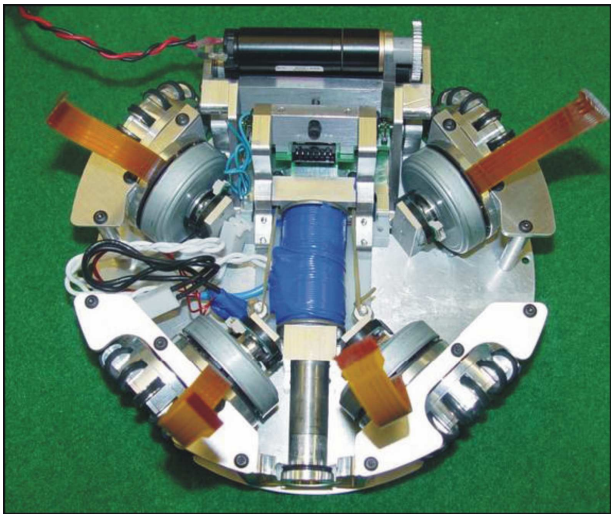


Figura 3.3: Vista inferior do robô CMDragons mostrando a assimetria na distribuição das rodas devido à necessidade de instalação dos sistemas de chute e drible. (Fonte: (Zickler et al., 2010)).

Apesar de algumas configurações não apresentarem os quatro ângulos iguais, ainda existe uma simetria em relação ao eixo Y_m mostrado na Figura 3.1, ou seja, percebe-se que nessas configurações $\varphi_1 = \varphi_2$ e $\varphi_3 = \varphi_4$. Dessa forma, neste trabalho será adotada essa igualdade entre os dois ângulos dianteiros e entre os dois ângulos traseiros. Então, por simplicidade, desse ponto em diante, define-se o ângulo dianteiro como $\varphi_d = \varphi_1 = \varphi_2$ e o traseiro como $\varphi_t = \varphi_3 = \varphi_4$. A Figura 3.4 apresenta o novo modelo geométrico do robô.

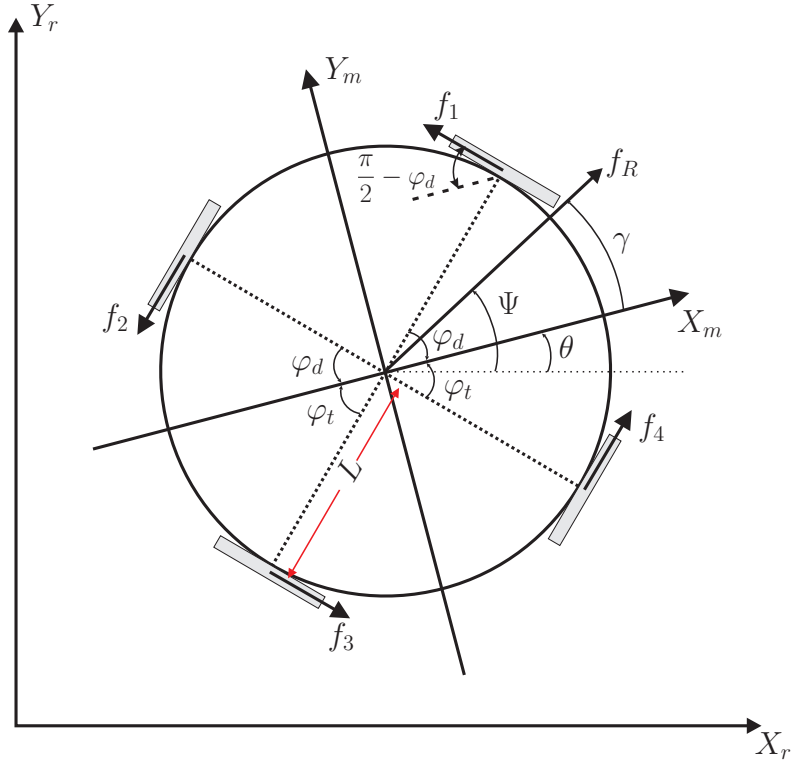


Figura 3.4: Modelo proposto neste trabalho. A geometria do robô apresenta simetria em relação ao eixo Y_m.

Para analisar a influência da distribuição angular das rodas no desempenho do robô, avalia-se a distribuição de aceleração para todas as direções de translação do robô, mantendo o ângulo de rotação θ igual a zero, ou seja, $\Psi = \gamma$. Nesse caso, varia-se Ψ de $0°$ até $360°$. A Figura 3.5 apresenta um esboço da variação da orientação do vetor de translação.

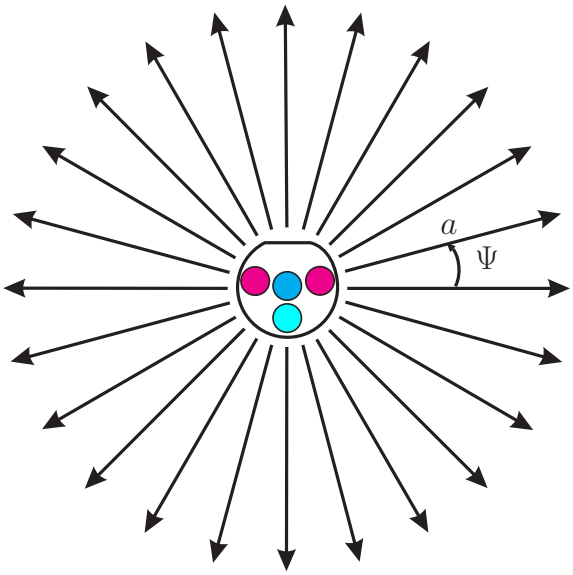


Figura 3.5: Variação da orientação do vetor de translação para $0° < \Psi < 360°$.

De forma a encontrar a contribuição de cada roda na locomoção do robô, assume-se um deslocamento em uma determinada direção Ψ com aceleração linear a unitária e aceleração angular $\ddot{\theta}$ igual a zero. Utiliza-se a primeira derivada da equação (3.14) para transformar a aceleração linear do robô nas respectivas acelerações das rodas, no entanto, de forma a simplificar a representação, utiliza-se a mesma unidade (m/s^2) para representar as acelerações lineares $\ddot{X}_m$ e $\ddot{Y}_m$ e a aceleração angular:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -\sin\varphi_d & \cos\varphi_d & 1 \\ -\sin\varphi_d & -\cos\varphi_d & 1 \\ \sin\varphi_t & -\cos\varphi_t & 1 \\ \sin\varphi_t & \cos\varphi_t & 1 \end{bmatrix} \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ L\ddot{\theta} \end{bmatrix}. \tag{3.51}$$

A decomposição vetorial da aceleração linear do robô em x e y é realizada tendo como referência a Figura 3.6, que representa o vetor aceleração para uma determinada direção de translação.

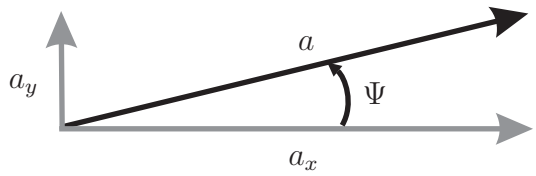


Figura 3.6: Decomposição do vetor de aceleração em suas componentes x e y.

Dessa forma, a decomposição da aceleração pode ser representada por:

$$\begin{aligned} a_x &= a\cos\Psi, \\ a_y &= a\sin\Psi, \end{aligned} \tag{3.52}$$

com Ψ dado em graus. Utilizando-se a notação apresentada na Seção 3.2.2, e tomando-se $a = 1$, as componentes de aceleração do robô em x e y podem ser representadas por:

$$\begin{aligned} \ddot{X}_m &= \cos\Psi, \\ \ddot{Y}_m &= \sin\Psi. \end{aligned} \tag{3.53}$$

Substituindo-se (3.53) em (3.51), tem-se:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \begin{bmatrix} -\sin\varphi_d & \cos\varphi_d & 1 \\ -\sin\varphi_d & -\cos\varphi_d & 1 \\ \sin\varphi_t & -\cos\varphi_t & 1 \\ \sin\varphi_t & \cos\varphi_t & 1 \end{bmatrix} \begin{bmatrix} \cos\Psi \\ \sin\Psi \\ 0 \end{bmatrix}. \tag{3.54}$$

Com base no diagrama mostrado na Figura 3.4, pode-se perceber que $\varphi_d = \varphi_t = 45°$ representa a distribuição com melhor simetria possível, uma vez que as rodas ficam exatamente no centro de cada quadrante. Essa simetria implica que a aceleração resultante seja a mais homogênea possível, uma vez que $\cos(45°) = \sin(45°)$. Além disso, $\cos(45°) = \sin(45°)$ garante que a contribuição total dos motores nas direções x e y sejam iguais. Dessa forma, a análise para a configuração com $\varphi_d = \varphi_t = 45°$ torna-se uma referência para as demais configurações. A Figura 3.7 apresenta a contribuição que cada motor deve realizar para que o robô locomova-se com aceleração unitária para direções de translação Ψ variando de 0° a 360°. Veja que as acelerações de cada motor estão normalizadas de forma a facilitar a análise.

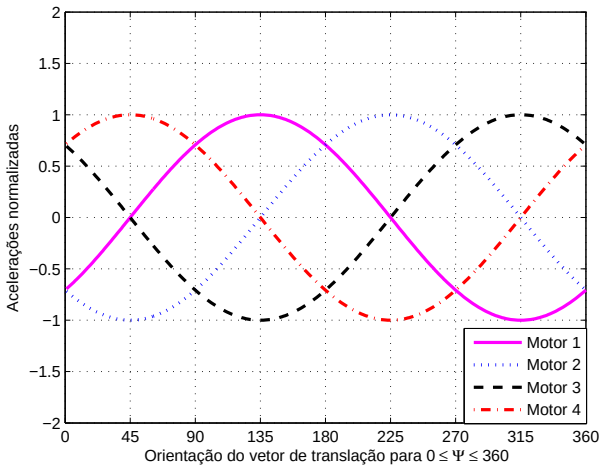


Figura 3.7: Contribuição de cada motor na distribuição de acelerações para $\varphi_d = \varphi_t = 45°$.

Analisando-se a Figura 3.7, é possível saber quantos motores estão efetivamente contribuindo para que o robô locomova-se para uma determinada direção. Veja por exemplo a translação para

$\Psi = 45°$. Nesse caso, apenas os motores 2 e 4 estão contribuindo para a locomoção do robô, uma vez que o robô está se movendo sobre o eixo dos motores 1 e 3, fazendo com que esses motores não sejam acionados. Já para $\Psi = 90°$, todos os motores estão sendo acionados e desenvolvem uma aceleração de $\sin(45°) = 0{,}7071$ cada, gerando uma contribuição total de 2,83 motores nessa direção.

Para realizar uma análise mais completa sobre a contribuição total dos motores para cada direção de translação utiliza-se a Figura 3.8 que apresenta a soma das contribuições de cada motor para atingir uma aceleração de 1 m/s^2 para todas as direções de translação conforme diagrama mostrado na Figura 3.5. Nessa figura, foi acrescentada uma curva que representa a distribuição média, ou seja, a soma das contribuições para todas as direções dividida por 360. Essa curva possui mesma área que a curva da soma das contribuições dos motores, então, se a diferença entre elas for pequena, significa que a distribuição está mais próxima da média, dessa forma, ela serve como uma referência para encontrar uma configuração para o robô que mais se aproxima da distribuição média. Vale ressaltar que quanto maior o número de motores, mais uniforme será a distribuição. É claro que na prática não se pode colocar um número grande de motores, tornando a investigação da posição das rodas ainda mais relevante.

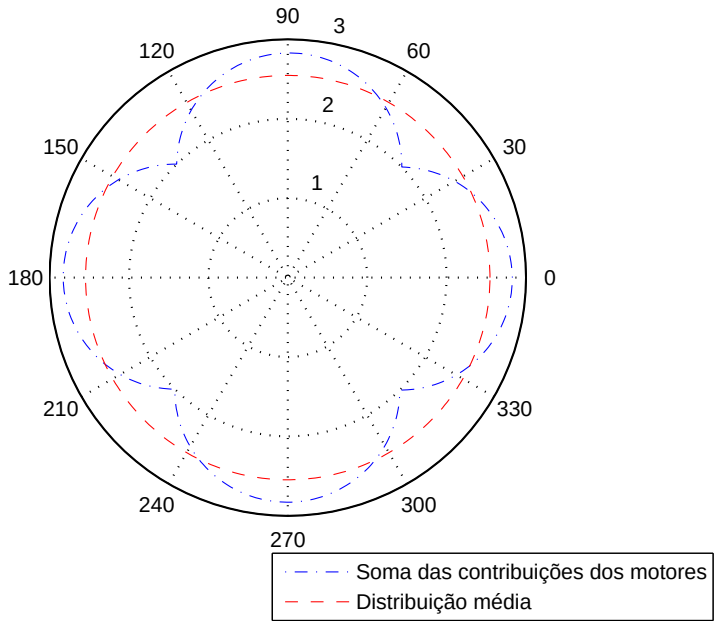


Figura 3.8: Contribuição total dos motores na distribuição da aceleração em função do ângulo de translação para $\varphi_d = \varphi_t = 45°$ para atingir uma aceleração de 1 m/s^2.

Veja, na Figura 3.8, que dependendo da direção de translação existem mais ou menos motores contribuindo, no entanto, não significa que toda aceleração disponível será entregue para o robô, uma vez que deve-se realizar a soma vetorial das acelerações em cada direção e nesse caso algumas componentes podem se cancelar. Como um exemplo, examina-se a locomoção do robô para a direção de 90° mostrada na Figura 3.8. Conforme equação (3.54), locomovendo-se na direção de 90°, cada motor realiza uma contribuição de $a_i = \sin(45°) = 0{,}7071$, $i = 1, \ldots, 4$, ou seja, cada motor exerce 70,71 % de sua aceleração máxima. Para saber a aceleração exercida pelo

robô na direção desejada, deve-se calcular a aceleração resultante gerada por cada motor e então somá-las. A Figura 3.9 apresenta a aceleração exercida pelo motor 1.

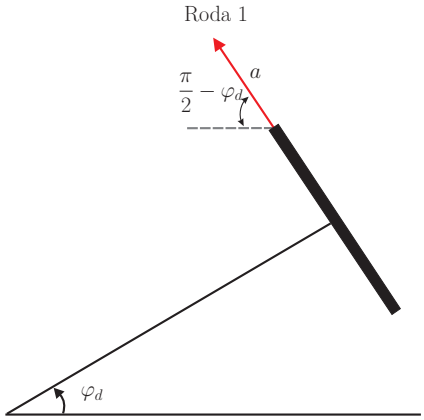


Figura 3.9: Aceleração exercida pelo motor 1.

Como deseja-se saber a aceleração resultante para a direção de 90°, basta calcular a componente de cada motor na direção Y_m mostrada na Figura 3.4. Essa componente pode ser calculada por:

$$a_{i_y} = a_i \sin\left(\frac{\pi}{2} - \varphi_d\right), \tag{3.55}$$

em que $i = 1, \ldots, 4$ representa os motores. Para a configuração analisada, cada motor contribui com $a_{i_y} = 0{,}5$, dessa forma, a contribuição total na direção de 90° é igual a 2 motores.

As componentes de aceleração nas direções X_m e Y_m e a aceleração angular do robô podem ser calculadas diretamente através da utilização das equações (3.54) e (3.9), substituindo φ_1 e φ_2 por φ_d e φ_3 e φ_4 por φ_t. A Figura 3.10 apresenta essas componentes.

A Figura 3.10 apresenta as componentes vetoriais de aceleração; no entanto, interessa-nos saber a aceleração resultante em função do ângulo de translação Ψ. Essa aceleração resultante tem a mesma direção da força $\mathbf{f}_R$ mostrada na Figura 3.4 e pode ser calculada transformando as componentes encontradas pela equação (3.9) em coordenadas polares. A Figura 3.11 apresenta uma comparação entre a soma das contribuições dos motores e a aceleração resultante em função do ângulo de translação.

Veja que para algumas direções de translação, a contribuição total dos motores é diferente da aceleração resultante, ou seja, aquela aceleração que efetivamente é entregue ao robô. Isso acontece, pois, algumas componentes de aceleração dos motores são canceladas dependendo da direção de translação. Como exemplo, volta-se ao caso analisado anteriormente para o robô locomovendo-se na direção de 90°. Nesse caso, a componente em x do motor 1 cancela-se com a componente do motor 2 e a componente do motor 3 cancela-se com a do motor 4.

Com base nas simulações mostradas até o momento, pode-se perceber que variando os ângulos φ_d e φ_t é possível alterar a forma como a aceleração é distribuída. Essa variação angular pode ser utilizada para melhorar o desempenho do robô conforme o objetivo do projeto. Dessa forma, pode-se elaborar algum tipo de indicador para que se possa avaliar a qualidade da configuração escolhida. O indicador utilizado neste trabalho é o erro quadrático médio da aceleração

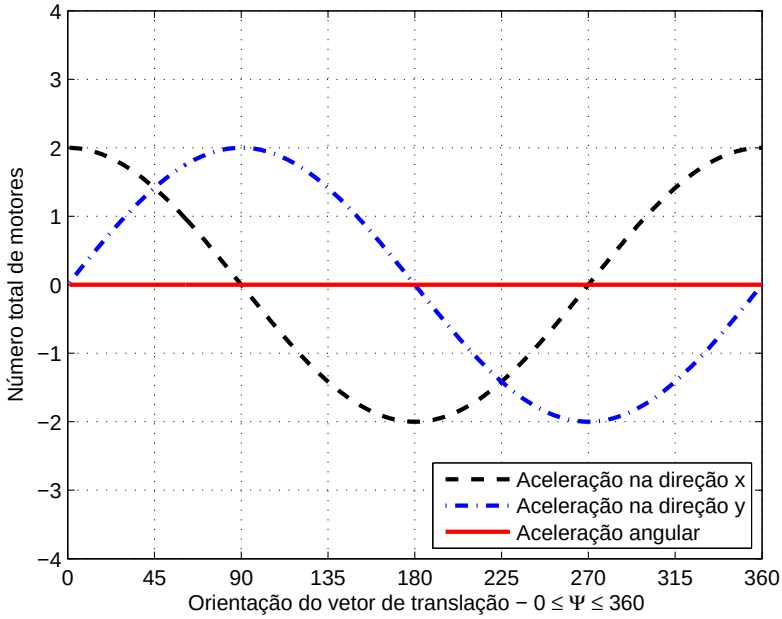


Figura 3.10: Componentes de aceleração em x e y e aceleração angular para $\varphi_d = \varphi_t = 45°$.

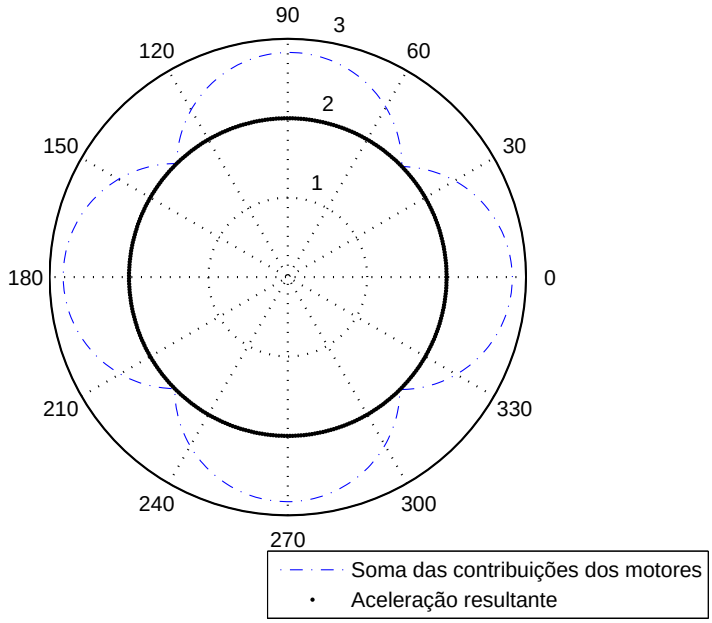


Figura 3.11: Comparação entre a soma das contribuições dos motores e aceleração resultante em função do ângulo de translação para $\varphi_d = \varphi_t = 45°$.

resultante:

$$EQA = \frac{1}{n}\sum_{i=1}^{n}(\hat{a} - a)^2, \tag{3.56}$$

em que $\hat{a}$ é a aceleração resultante real, a é a aceleração resultante ideal, encontrada para um

robô simétrico com $\varphi_d = \varphi_t = 45°$ mostrada na Figura 3.11 e n é o número de elementos do vetor de acelerações.

O índice EQA ao ser minimizado leva a uma configuração de motores com maior uniformidade de aceleração resultante em todas as direções de translação. Para avaliar esse índice, será verificado como φ_d e φ_t influenciam na distribuição de aceleração.

De acordo com a Figura 3.4, φ_d e φ_t podem variar no intervalo de 0 a 90 graus. Assim, variando φ_d e φ_t de grau em grau, são 91 possíveis ângulos. Dessa forma, para cada valor de φ_d, existem 91 possibilidades de escolha para φ_t, resultando em 8281 possíveis configurações para o robô. Pode-se obter um gráfico tridimensional relacionando todas as configurações possíveis ao EQA que elas geram. Esse gráfico é mostrado na Figura 3.12.

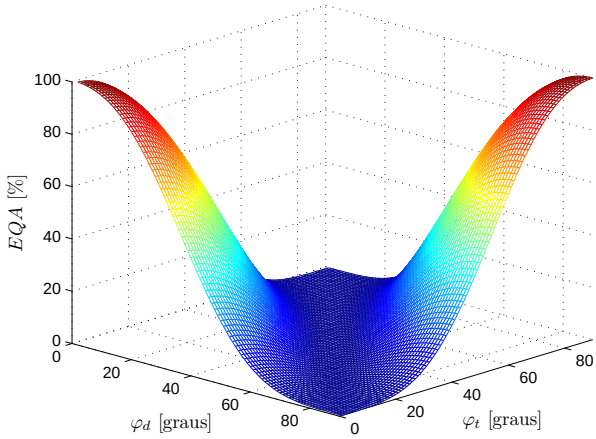


Figura 3.12: Erro médio quadrático da aceleração resultante encontrado em função da configuração.

A Figura 3.12 apresenta as 8281 configurações possíveis. A região inferior (mais escura) no gráfico representa as configurações que apresentam pequenos erros, dessa forma, escolhendo um dos ângulos (dianteiro ou traseiro) é possível encontrar o outro ângulo que represente o menor erro. Pela Figura 3.12, observa-se a existência de um mínimo global para o EQA.

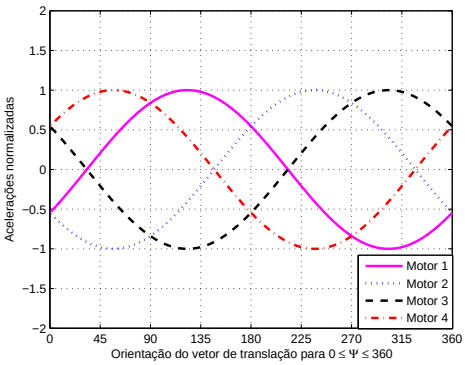
Conforme foi citado no início desta Seção, não existe na literatura uma definição de qual configuração utilizar. Com base nessa carência, será realizada uma análise das configurações utilizadas por algumas equipes de forma a verificar a possibilidade de melhoria no desempenho do robô.

3.3.1 Estudos de Caso

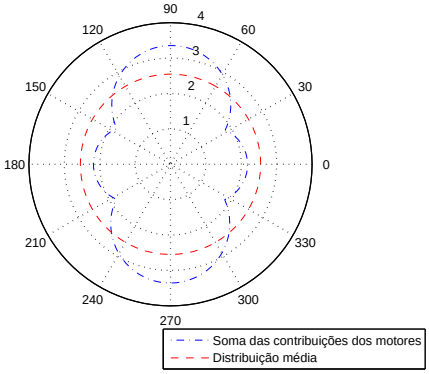
A seguir são apresentados alguns estudos de caso de forma a investigar a influência dos ângulos φ_d e φ_t na distribuição de acelerações do robô. Esses casos englobam o caso simétrico em que $\varphi_d = \varphi_t$ e o caso assimétrico em que $\varphi_d \neq \varphi_t$.

Caso Simétrico - $\varphi_d = 33°$ **e** $\varphi_t = 33°$

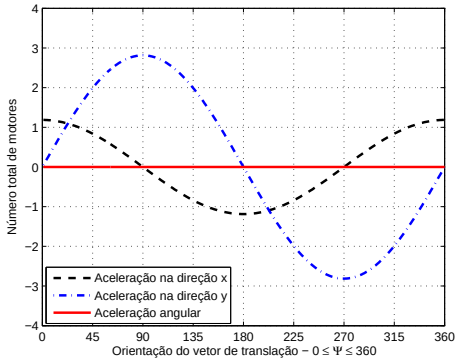
A configuração a ser analisada nesse estudo de caso é a que apresenta $\varphi_d = \varphi_t = 33°$ utilizada em (Jr. et al., 2010). A Figura 3.13 apresenta os gráficos para essa configuração.



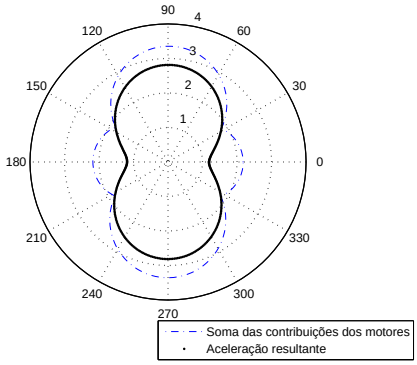
(a) Contribuição de cada motor na distribuição de acelerações para $\varphi_d = \varphi_t = 33°$.



(b) Contribuição total dos motores na distribuição da aceleração em função do ângulo de translação para $\varphi_d = \varphi_t = 33°$.



(c) Componentes de aceleração em x e y e aceleração angular para $\varphi_d = \varphi_t = 33°$.



(d) Comparação entre a soma das contribuições dos motores e aceleração resultante em função do ângulo de translação para $\varphi_d = \varphi_t = 33°$.

Figura 3.13: Análise da configuração simétrica com $\varphi_d = \varphi_t = 33°$.

Pode-se notar na Figura 3.13 que o robô desenvolve acelerações diferentes daquelas mostradas anteriormente para o caso $\varphi_d = \varphi_t = 45°$. Isso ocorre porque nessa configuração, mesmo havendo simetria, tem-se que $\cos(33°) \neq \sin(33°)$. Essa diferença faz com que a soma vetorial das componentes de aceleração sejam diferentes. Veja por exemplo a Figura 3.13d que apresenta a aceleração resultante que o robô é capaz de empregar em cada direção. Para $\Psi = 90°$, o robô possui uma aceleração resultante equivalente a 2,82 motores, já na direção $\Psi = 0°$, essa aceleração é de apenas 1,19 motores. Essa configuração otimiza a locomoção do robô na direção y em detrimento à locomoção na direção x.

Essa característica de variação da distribuição de aceleração em função do posicionamento das rodas pode ser utilizada para melhorar o desempenho do robô em determinadas direções, como exemplo, na direção x para um goleiro ou na direção y para um atacante. No entanto, se o projeto precisa de um robô com distribuição de aceleração homogênea, pode-se escolher os

ângulos de disposição das rodas com base na Figura 3.12.

Assumindo que a escolha do ângulo dianteiro φ_d foi devida à necessidade de acomodação dos sistemas de chute e drible na parte dianteira do robô, é possível escolher o ângulo traseiro φ_t de forma que EQA seja minimizado. Para isso, realiza-se uma busca na Figura 3.12 para encontrar φ_t que gere o menor erro.

Tendo definido φ_d, o gráfico tridimensional mostrado na Figura 3.12 pode ser reduzido a um bidimensional, uma vez que apenas o ângulo traseiro precisa ser variado. A Figura 3.14 apresenta o gráfico de EQA em função de φ_t para $\varphi_d = 33°$.

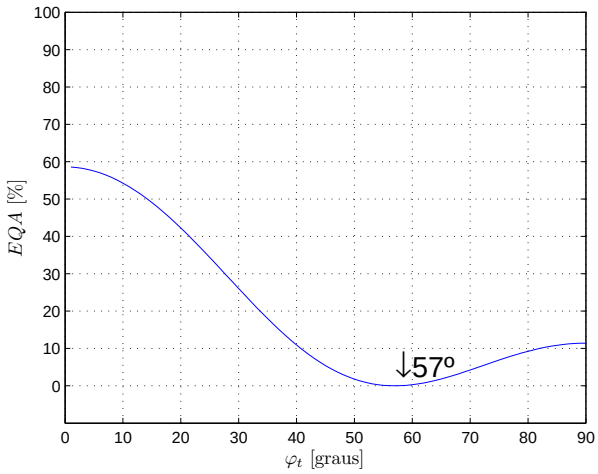
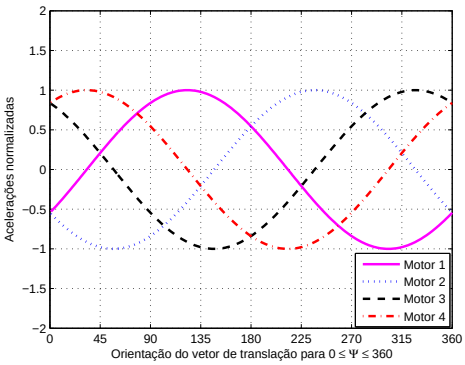


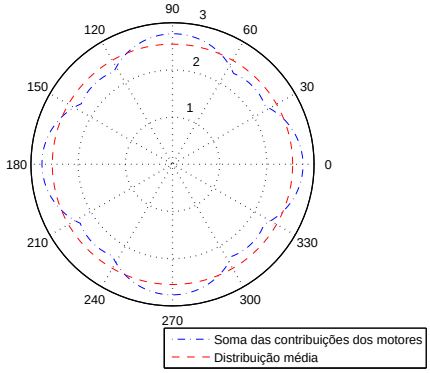
Figura 3.14: Erro médio quadrático da aceleração em função de φ_t para $\varphi_d = 33°$. Indicação do valor mínimo.

Esse gráfico apresenta um mínimo global em $\varphi_t = 57°$, isso significa que esse é o ângulo que proporciona o menor erro quando utilizado $\varphi_d = 33°$. Veja que $\cos(33°) = \sin(57°)$ e $\cos(57°) = \sin(33°)$ o que implica que a contribuição total dos motores nas direções x e y sejam iguais, conforme pode ser visto na Figura 3.15b. Além disso, essa correção na configuração do robô proporciona uma aceleração resultante idêntica à da configuração com $\varphi_d = \varphi_t = 45°$ mostrada na Figura 3.11, conforme Figura 3.15d. Veja, na Figura 3.14 que $EQA = 0$.

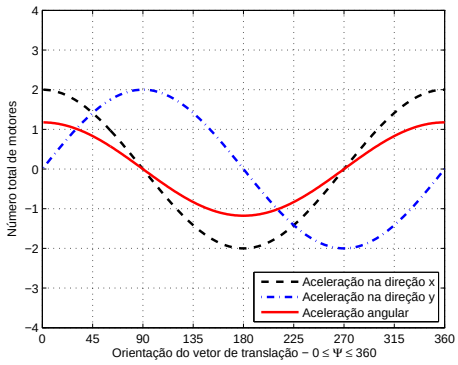
É importante notar que essa correção torna o robô assimétrico, fazendo com que seja gerada uma aceleração angular para determinadas direções de translação conforme pode ser visto na Figura 3.15c. Essa aceleração angular, por sua vez, criará um momento de rotação que deverá ser corrigido pelo sistema de controle do robô. Comparando os resultados mostrados na Figura 3.15d com os da Figura 3.11, percebe-se que para $\varphi_d = \varphi_t = 45°$ a maior contribuição é de 2,83 motores, enquanto que para $\varphi_d = 33°$ e $\varphi_t = 57°$ a maior contribuição é de 2,77, no entanto, a aceleração resultante é a mesma nos dois casos. Isso mostra que a configuração $\varphi_d = 33°$ e $\varphi_t = 57°$ exige menos dos motores, o que implica em um menor consumo energético instantâneo. Essa redução no consumo torna-se ainda mais evidente quando se compara com o caso em que $\varphi_d = \varphi_t = 33°$ que apresenta uma contribuição máxima de 3,35 motores, conforme Figura 3.13d. Essa redução no consumo é um ponto importante e deve ser levado em consideração durante o projeto de robôs da categoria F180, uma vez que devido a limitações de tamanho, é necessário que se tenha baterias pequenas. Além disso, a redução no consumo pode significar aumento de



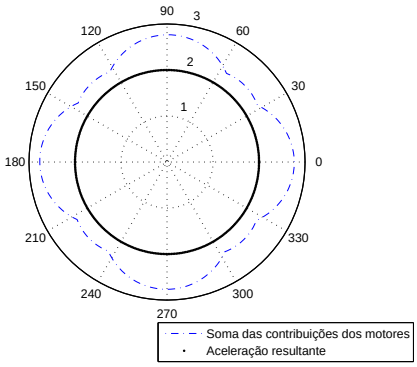
(a) Contribuição de cada motor na distribuição de acelerações para $\varphi_d = 33°$ e $\varphi_t = 57°$.



(b) Contribuição total dos motores na distribuição da aceleração em função do ângulo de translação para $\varphi_d = 33°$ e $\varphi_t = 57°$.



(c) Componentes de aceleração em x e y e aceleração angular para $\varphi_d = 33°$ e $\varphi_t = 57°$.



(d) Comparação entre a soma das contribuições dos motores e aceleração resultante em função do ângulo de translação para $\varphi_d = 33°$ e $\varphi_t = 57°$.

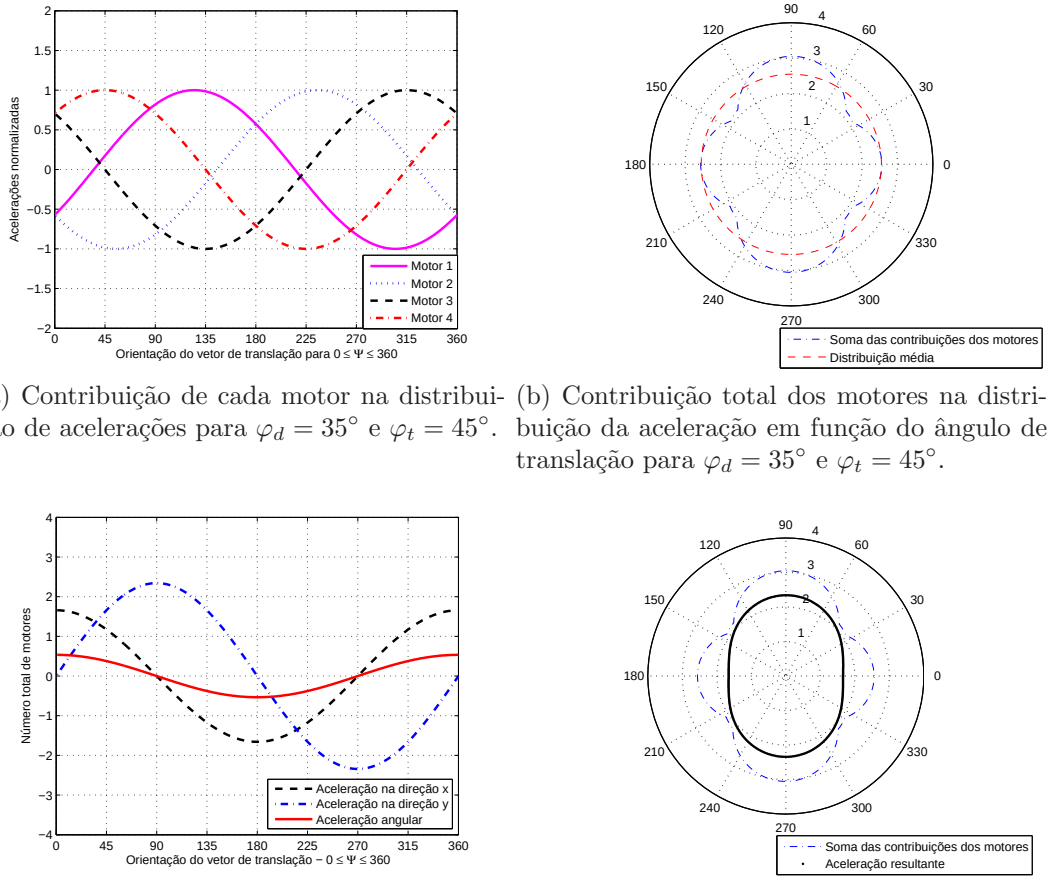
Figura 3.15: Análise da configuração com $\varphi_d = 33°$ e $\varphi_t = 57°$.

autonomia para esses robôs.

Caso Assimétrico - $\varphi_d = 35°$ e $\varphi_t = 45°$

Após o desenvolvimento de um estudo de caso para uma configuração simétrica, prossegue-se com a análise de uma configuração assimétrica. No início desta Seção, foram apresentados alguns exemplos de configurações utilizadas pelas principais equipes competindo na liga *SSL*. A configuração em que $\varphi_d = 33°$ e $\varphi_t = 45°$ utilizada em (Fialho et al., 2010; Sharbafi et al., 2010), terá como resultados os mesmos apresentados na Figura 3.15 uma vez que o ângulo dianteiro é o mesmo. Assim, a configuração apresentada em (Pranggonoh et al., 2010), na qual $\varphi_d = 35°$ e $\varphi_t = 45°$ será utilizada nesse estudo de caso. A Figura 3.16 apresenta as simulações para esse caso.

Veja que na Figura 3.16c existe uma componente de aceleração angular devido a falta de simetria. Na Figura 3.16d é possível ver que a aceleração resultante apresenta um máximo de 2,34 motores na direção y e de 1,66 na direção x, enquanto que a contribuição total dos motores apresenta um máximo de 3,05 motores na direção y e de 2,56 na direção x.



(a) Contribuição de cada motor na distribuição de acelerações para $\varphi_d = 35°$ e $\varphi_t = 45°$.

(b) Contribuição total dos motores na distribuição da aceleração em função do ângulo de translação para $\varphi_d = 35°$ e $\varphi_t = 45°$.

(c) Componentes de aceleração em x e y e aceleração angular para $\varphi_d = 35°$ e $\varphi_t = 45°$.

(d) Comparação entre a soma das contribuições dos motores e aceleração resultante em função do ângulo de translação para $\varphi_d = 35°$ e $\varphi_t = 45°$.

Figura 3.16: Análise da configuração com $\varphi_d = 35°$ e $\varphi_t = 45°$.

Aplicando a mesma análise realizada no estudo de caso anterior, a Figura 3.17 apresenta o gráfico do EQA em função de φ_t para $\varphi_d = 35°$. Esse gráfico apresenta um mínimo global em $\varphi_t = 55°$, isso significa que esse é o ângulo que proporciona o menor erro quando utilizado $\varphi_d = 35°$. Da mesma forma que no exemplo anterior, tem-se que $\cos(35°) = \sin(55°)$ e $\cos(55°) = \sin(35°)$ o que implica que a contribuição total dos motores nas direções x e y sejam iguais, conforme pode ser visto na Figura 3.18b. Além disso, essa correção na configuração do robô proporciona uma aceleração resultante idêntica à da configuração com $\varphi_d = \varphi_t = 45°$ mostrada na Figura 3.11, conforme Figura 3.18d. Novamente, percebe-se uma redução na exigência dos motores, uma vez que essa configuração corrigida apresenta uma contribuição total máxima de 2,78 motores, contra os 3,05 motores do caso original. Veja, na Figura 3.17, que novamente o $EQA = 0$.

Foram realizados testes computacionais para todos os valores de $0 < \varphi_d < 90$, sendo que $\varphi_d + \varphi_t = 90°$ apresenta o menor valor de EQA. Com base nessas evidências computacionais, formulou-se a Conjectura 3.1.

Conjectura 3.1 *Sejam os ângulos φ_1, φ_2, φ_3 e φ_4 os ângulos que os eixos das rodas fazem*

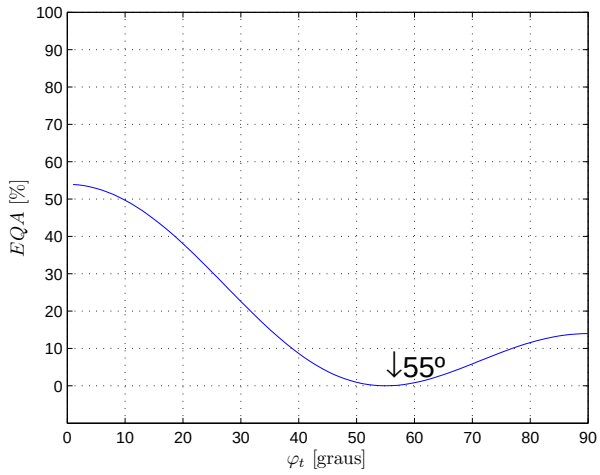
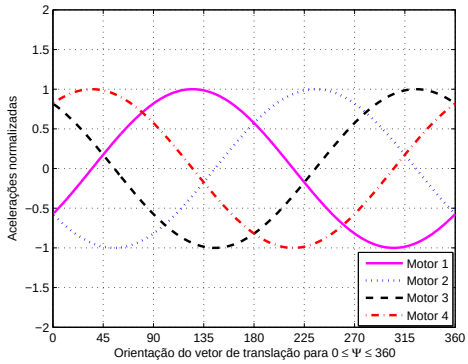
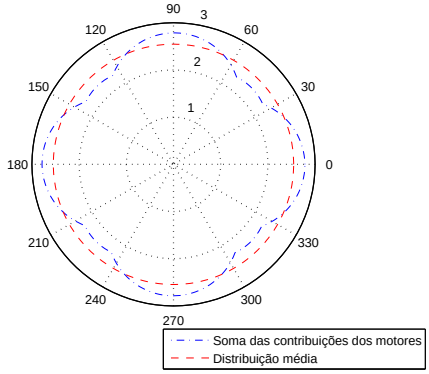


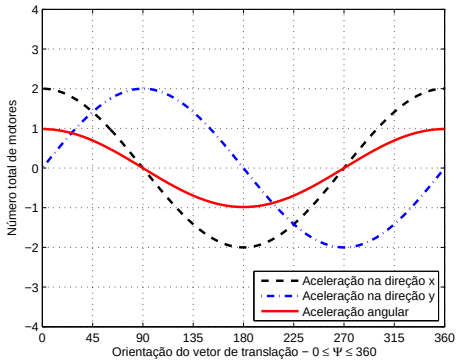
Figura 3.17: Erro médio quadrático da aceleração em função de φ_t para $\varphi_d = 35°$. Indicação do valor mínimo.



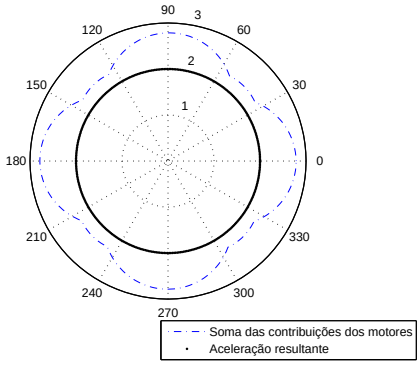
(a) Contribuição de cada motor na distribuição de acelerações para $\varphi_d = 35°$ e $\varphi_t = 55°$.



(b) Contribuição total dos motores na distribuição da aceleração em função do ângulo de translação para $\varphi_d = 35°$ e $\varphi_t = 55°$.



(c) Componentes de aceleração em x e y e aceleração angular para $\varphi_d = 35°$ e $\varphi_t = 55°$.



(d) Comparação entre a soma das contribuições dos motores e aceleração resultante em função do ângulo de translação para $\varphi_d = 35°$ e $\varphi_t = 55°$.

Figura 3.18: Análise da configuração com $\varphi_d = 35°$ e $\varphi_t = 55°$.

com o eixo X_m. Ao arbitrar um valor idêntico para os ângulos dianteiros $\varphi_d = \varphi_1 = \varphi_2$, obtém-se o menor valor de EQA para $\varphi_t = 90° - \varphi_d$, sendo φ_t um valor idêntico para os ângulos traseiros $\varphi_t = \varphi_3 = \varphi_4$.

3.4 Conclusões

Neste Capítulo, os modelos cinemático e dinâmico de um robô móvel omnidirecional de quatro rodas foram desenvolvidos. O modelo dinâmico foi desenvolvido de forma genérica, ou seja, seus parâmetros e os ângulos de disposição das rodas foram deixados como incógnitas do modelo. Dessa forma, sua aplicação para robôs com configurações diferentes torna-se simples. Foi apresentado um estudo sobre a escolha da disposição física das rodas na periferia do robô. Foi mostrado que mesmo que existam restrições de espaço físico que obriguem a escolha do ângulo dianteiro de disposição das rodas, ainda assim é possível projetar um robô que possa ter uma aceleração resultante igual ao do caso em que as rodas estão dispostas simetricamente a 45°. Esse resultado está expresso na Conjectura 3.1. O estudo apresentado neste trabalho torna-se então relevante para equipes que pretendem projetar ou melhorar robôs existentes.

Capítulo 4

Controle Robusto do Robô F180 Utilizando Linearização por Realimentação

Neste capítulo, controladores para seguimento de referência são aplicados ao modelo matemático do robô F180 desenvolvido no Capítulo 3. Primeiramente, realiza-se a linearização do modelo utilizando-se as técnicas apresentadas na Seção 2.3. Em seguida, a topologia de controle e condições para síntese de controladores por realimentação de estados na forma de desigualdades matriciais lineares (LMIs) são propostas. Por fim, experimentos numéricos são realizados para comparar o desempenho dos controladores.

4.1 Introdução

Na Seção 2.3, foram apresentados dois métodos para linearização de um sistema por meio da realimentação de 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 respectivo valor nominal. De maneira intuitiva, espera-se que ao utilizar a linearização por realimentação robusta, o sistema em malha fechada apresente certa robustez, pois o controle linear é projetado para um sistema similar àquele no qual o controle é aplicado e que responde de forma semelhante às variações paramétricas. Por esse motivo, os resultados apresentados neste capítulo são baseados em uma linearização por realimentação robusta.

Após realizar a linearização do sistema, é possível utilizar algum método de controle linear para projetar um controle para o sistema linearizado; no entanto, é necessário a definição de uma topologia de controle que atenda aos requisitos do projeto. Em uma partida de futebol, deseja-se que o robô movimente-se de forma a capturar a bola ao mesmo tempo em que desvia-se de outros robôs. Para realizar essa tarefa, uma trajetória descrevendo o caminho a ser seguido é gerada e, a partir dessa trajetória, velocidades de referência são enviadas para o robô (Franco, 2006a). Dessa forma, a topologia escolhida neste trabalho é a de seguimento de referência apresentada em (Chen, 1999) e denominada rastreamento robusto, do inglês *Robust Tracking*.

4.2 Linearização do Modelo do Robô F180

Neste trabalho, os ângulos de disposição das rodas são definidos como sendo $\varphi_d = \varphi_t = 45°$, com base nas análises apresentadas na Seção 3.3. Substituindo esses ângulos no modelo não linear dado pela equação (3.47), tem-se:

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

em que:

$$\begin{aligned} a_1 &= 2\frac{n^2\left(b_oR_a + K_tK_{em}\right)}{R_a\left(r^2m + 2n^2J_o\right)}, \\ a_2 &= \frac{r^2m\dot{\theta}}{r^2m + 2n^2J_o}, \\ a_3 &= 4\frac{n^2L^2\left(b_oR_a + K_tK_{em}\right)}{R_a\left(4n^2J_oL^2 + r^2J_w\right)}, \\ b_1 &= \frac{nrK_tV_{ref}}{\sqrt{2}R_a(r^2m + 2n^2J_o)}, \\ b_2 &= \frac{nLK_tV_{ref}r}{R_a(J_wr^2 + 4J_on^2L^2)}. \end{aligned}$$

A saída do sistema é escolhida como sendo os próprios estados, assim:

$$y = \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}. \tag{4.2}$$

O modelo dado por (4.1)-(4.2) não atende a um dos requisitos para utilizar a técnica de linearização por realimentação, uma vez que esse não possui o mesmo número de entradas e saídas. Dessa forma, é necessário realizar uma transformação no sistema de maneira que esse tenha o mesmo número de entradas e de saídas. O sistema (4.1) pode então ser escrito como

$$\begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -a_1 & a_2\dot{\theta} & 0 \\ -a_2\dot{\theta} & -a_1 & 0 \\ 0 & 0 & -a_3 \end{bmatrix} \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_1 & 0 \\ 0 & 0 & b_2 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}, \tag{4.3}$$

em que

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix}. \tag{4.4}$$

A equação (4.4) está na forma $b = Ax$, em que $A \in \mathbb{R}^{3\times 4}$. Dessa forma há mais variáveis do que equações e x é sub-especificado, ou seja, muitas escolhas de x levam a um mesmo b. Esse tipo de sistema de equações é conhecido como sistema subdeterminado e permite infinitas soluções. Neste trabalho, é utilizada a solução particular que minimiza a norma do vetor x, $\|x\|$, dada por:

$$x = A^T(AA^T)^{-1}b. \tag{4.5}$$

Dado um vetor de entradas $[\mu_1 \ \mu_2 \ \mu_3]^T$, a equação (4.5) pode então ser utilizada para encontrar o vetor de entradas $[\rho_1 \ \rho_2 \ \rho_3 \ \rho_4]^T$ que possui a menor norma. Como ρ_i representa as tensões aplicadas ao motores do robô, o uso de um vetor que possui menor norma implica em um menor consumo energético pelo robô.

Após a transformação, o novo sistema formado pelas equações (4.3) e (4.2) possui o mesmo número de entradas e de saídas e pode então ser submetido à linearização por realimentação. Esse sistema satisfaz as hipóteses **H1**, **H2** e **H3** apresentadas na Seção 2.3.3 e, 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, de graus relativos $r_1 = r_2 = r_3 = 1$ respectivamente, com $r_1 + r_2 + r_3 = 3$, tais que o sistema dado por

$$\underbrace{\begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} -a_1 & a_2\dot{\theta} & 0 \\ -a_2\dot{\theta} & -a_1 & 0 \\ 0 & 0 & -a_3 \end{bmatrix} \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_1 & 0 \\ 0 & 0 & b_2 \end{bmatrix}}_{g(x)} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}, \tag{4.6}$$

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

é linearizável por realimentação (Isidori, 1989).

Uma vez garantidas as condições necessárias e suficientes para a linearização por realimentação, prossegue-se com a aplicação da técnica. Foi mostrado na Seção 2.3 que a linearização por realimentação robusta é uma extensão da linearização clássica e, dessa forma, é necessário primeiramente que a técnica de linearização clássica seja aplicada, para então aplicar a técnica de linearização robusta.

4.2.1 Linearização por Realimentação Clássica do Robô F180

Conforme apresentado na Seção 2.3.4, a linearização por realimentação clássica do sistema (4.6) é realizada utilizando um difeomorfismo

$$x_c = \phi_c(x), \tag{4.7}$$

e a realimentação de estados

$$u_c(x,v_c) = \alpha_c(x) + \beta_c(x)v_c, \tag{4.8}$$

sendo

$$\alpha_c(x) = \begin{bmatrix} \dfrac{\sqrt{2}\left(2n^2\dot{X}_m b_o R_a + 2n^2\dot{X}_m K_t K_{em} - r^2 m\dot{\theta}\dot{Y}_m R_a\right)}{nrK_tV_{ref}} \\ \dfrac{\sqrt{2}\left(r^2 m\dot{\theta}\dot{X}_m R_a + 2n^2\dot{Y}_m b_o R_a + 2n^2\dot{Y}_m K_t K_{em}\right)}{nrK_tV_{ref}} \\ 4\dfrac{nL\left(b_o R_a + K_t K_{em}\right)\dot{\theta}}{rK_tV_{ref}} \end{bmatrix}, \tag{4.9}$$

$$\beta_c(x) = \begin{bmatrix} \dfrac{\sqrt{2}R_a\left(r^2 m + 2n^2 J_o\right)}{nrK_tV_{ref}} & 0 & 0 \\ 0 & \dfrac{\sqrt{2}R_a\left(r^2 m + 2n^2 J_o\right)}{nrK_tV_{ref}} & 0 \\ 0 & 0 & \dfrac{R_a\left(4n^2 J_o L^2 + r^2 J_w\right)}{nrK_tV_{ref}L} \end{bmatrix}, \tag{4.10}$$

$$\phi_c(x) = \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}, \tag{4.11}$$

obtidos de (2.29), (2.26), (2.27) e (2.28) respectivamente.

Obtém-se então o sistema linearizado

$$\dot{x}_c = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A_c} x_c + \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{B_c} v_c. \tag{4.12}$$

que está na forma canônica de Brunovsky.

4.2.2 Linearização por Realimentação Robusta do Robô F180

Conforme apresentado na Seção 2.3.5, a linearização por realimentação robusta do sistema (4.6) é realizada utilizando um difeomorfismo

$$x_r = \phi_r(x), \tag{4.13}$$

e a realimentação de estados

$$u_r(x,v_r) = \alpha_r(x) + \beta_r(x)v_r, \tag{4.14}$$

sendo

$$\alpha_r(x) = \alpha_c(x) + \beta_c(x)LT^{-1}\phi_c(x), \tag{4.15}$$

$$\beta_r(x) = \beta_c(x)R^{-1}, \tag{4.16}$$

$$\phi_r(x) = \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}, \tag{4.17}$$

com

$$L = \begin{bmatrix} -2\dfrac{n^2\left(b_oR_a + K_tK_{em}\right)}{R_a\left(r^2m + 2n^2J_o\right)} & 0 & 0 \\ 0 & -2\dfrac{n^2\left(b_oR_a + K_tK_{em}\right)}{R_a\left(r^2m + 2n^2J_o\right)} & 0 \\ 0 & 0 & -4\dfrac{L^2n^2\left(b_oR_a + K_tK_{em}\right)}{R_a\left(4n^2J_oL^2 + r^2J_w\right)} \end{bmatrix}, \tag{4.18}$$

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.19}$$

$$R = \begin{bmatrix} \dfrac{\sqrt{2}R_a\left(r^2m + 2n^2J_o\right)}{nrK_tV_{ref}} & 0 & 0 \\ 0 & \dfrac{\sqrt{2}R_a\left(r^2m + 2n^2J_o\right)}{nrK_tV_{ref}} & 0 \\ 0 & 0 & \dfrac{R_a\left(4n^2J_oL^2 + r^2J_w\right)}{nrK_tV_{ref}L} \end{bmatrix}. \tag{4.20}$$

O sistema linearizado é dado por

$$\begin{aligned} \dot{x}_r &= \underbrace{\begin{bmatrix} -2\dfrac{n^2\left(b_oR_a + K_tK_{em}\right)}{R_a\left(r^2m + 2n^2J_o\right)} & 0 & 0 \\ 0 & -2\dfrac{n^2\left(b_oR_a + K_tK_{em}\right)}{R_a\left(r^2m + 2n^2J_o\right)} & 0 \\ 0 & 0 & -4\dfrac{L^2n^2\left(b_oR_a + K_tK_{em}\right)}{R_a\left(4n^2J_oL^2 + r^2J_w\right)} \end{bmatrix}}_{A_r} x_r \\ &+ \underbrace{\begin{bmatrix} \dfrac{1}{2}\dfrac{nrK_tV_{ref}\sqrt{2}}{R_a\left(r^2m + 2n^2J_o\right)} & 0 & 0 \\ 0 & \dfrac{1}{2}\dfrac{nrK_tV_{ref}\sqrt{2}}{R_a\left(r^2m + 2n^2J_o\right)} & 0 \\ 0 & 0 & \dfrac{nrK_tV_{ref}L}{R_a\left(4n^2J_oL^2 + r^2J_w\right)} \end{bmatrix}}_{B_r} v_r, \end{aligned} \tag{4.21}$$

que é equivalente à aproximação linear do sistema (4.6) em torno do ponto de equilíbrio $x_0 = 0$.

4.3 Topologia de Controle

Tendo finalizado o projeto da realimentação linearizante, é necessário projetar o sinal de controle v_r que seja capaz de estabilizar o sistema linearizado representado pela formulação (2.38). Como o projeto de v_r é feito para o sistema linearizado em torno de um ponto de equilíbrio, é possível utilizar as ferramentas clássicas de controle linear.

A maioria dos sistemas de controle podem ser formulados conforme Figura 4.1, na qual a planta e o sinal de referência $r(t)$ são conhecidos (Chen, 1999). A entrada $u(t)$ é chamada de sinal de controle e a saída $y(t)$ de sinal controlado. O problema é projetar um sistema de forma que a saída do sistema siga o sinal de referência com menor erro possível. Exitem dois tipos de controle. Quando o sinal de controle depende apenas do sinal de referência e é independente do sinal controlado, o controle é chamado controle em malha aberta. Mas se o sinal de controle depende tanto da referência quanto do sinal controlado, o controle é chamado controle em malha fechada.

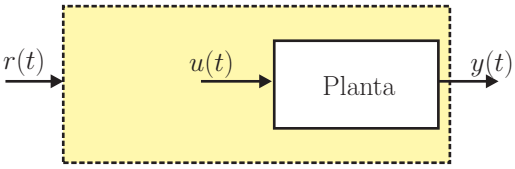


Figura 4.1: Descrição simplificada de um sistema de controle.

O controle em malha aberta geralmente não é satisfatório se existem incertezas paramétricas e/ou ruídos ou distúrbios agindo sobre o sistema. Um controle em malha fechada corretamente projetado, por outro lado, pode reduzir os efeitos das variações paramétricas e suprimir os ruídos e distúrbios. O modelo desenvolvido para representar um sistema pode mudar devido a mudança de carga, mudanças no ambiente, não linearidades ou envelhecimento dos componentes. Assim, variações nos parâmetros de um processo podem ocorrer em aplicações reais. Consequentemente, o controle em malha fechada é mais largamente utilizado.

Considere o sistema dado pela equação (2.38) com o par (A_r,B_r) controlável. Então a estrutura de controle mostrada na Figura 4.2 pode ser utilizada para fazer 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 que $r(t)$ é a referência a ser seguida.

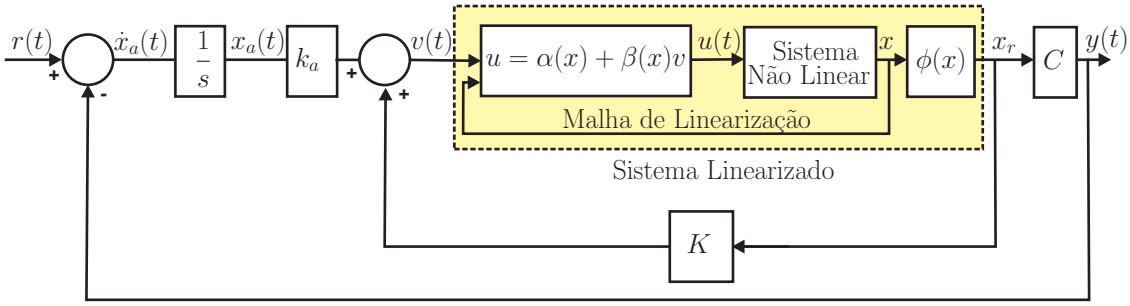


Figura 4.2: Controle de seguimento de referência com integração do sinal de erro.

A inclusão de um integrador ao sistema original, conforme apresentado na Figura 4.2, au-

menta a ordem do sistema e assegura erro nulo em regime permanente para uma entrada em degrau. Essa característica é importante do ponto de vista do futebol de robôs, uma vez que, durante o jogo, é geralmente requisitado aos robôs que interceptem a bola e deem passe para outro robô. Essa ação não seria possível se houvesse algum erro em regime permanente. Deve-se notar que essa topologia é semelhante a um controlador do tipo Proporcional Integral (PI).

Essa estrutura de controle é representada no espaço de estados como

$$\underbrace{\begin{bmatrix} \dot{x}_r \\ \dot{x}_a \end{bmatrix}}_{\dot{\tilde{x}}} = \left(\underbrace{\begin{bmatrix} A_r & 0 \\ -C_r & 0 \end{bmatrix}}_{\tilde{A}} + \underbrace{\begin{bmatrix} B_r \\ 0 \end{bmatrix}}_{\tilde{B}} \underbrace{\begin{bmatrix} K & k_a \end{bmatrix}}_{\tilde{K}} \right) \underbrace{\begin{bmatrix} x_r \\ x_a \end{bmatrix}}_{\tilde{x}} + \underbrace{\begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix}}_{E} r, \tag{4.22}$$

em que K e k_a são os ganhos de realimentação de estados e os ganhos de integração, respectivamente. Sendo o par $(\tilde{A},\tilde{B})$ controlável, então é possível alocar os polos de $(\tilde{A}+\tilde{B}\tilde{K})$ utilizando alguma ferramenta de sistemas lineares, tais como a alocação de polos.

Um dos problemas fundamentais na teoria de controle é o projeto de leis de controle capazes de alocar os polos de malha fechada de um sistema em uma região desejada. Dentre as diferentes formas para definição de especificações de desempenho, uma das mais populares é aquela que impõe alguma restrição para a alocação dos polos do sistema em malha fechada, como tempo de assentamento ou máximo sobressinal (Chilali et al., 1999). Grande parte da literatura relativa a alocação de polos, foca no problema de alocação exata, no qual é requerido que os polos de malha fechada estejam posicionados sobre ou próximos a locais previstos. Na prática, a alocação exata de polos 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 para sistemas discretos no tempo (Haddad e Bernstein, 1992).

Na literatura, o problema de alocação regional de polos tem sido considerado para uma variedade de regiões, incluindo faixas verticais e horizontais, setores, círculos e hipérboles (Chilali et al., 1999).

Neste trabalho, o problema de alocação de polos é considerado para uma região circular conforme Figura 4.3.

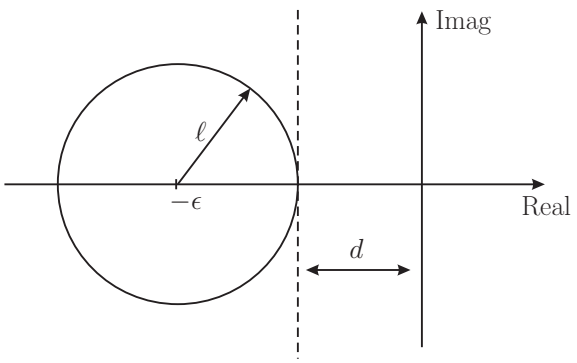


Figura 4.3: Região circular para alocação de polos.

Note que essa especificação para os polos do sistema em malha fechada garante que toda a dinâmica esteja limitada por exponenciais com decaimento limitado por $\exp(-\epsilon+\ell)t$ e frequên-

cias (parte imaginária dos polos) menores ou iguais a ℓ. Em particular, para sistemas de segunda ordem, essa 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.

O projeto de controle resume-se então, em calcular a matriz $\tilde{K} = [K\ k_a]$ dada pela equação (4.22), tal que o sistema realimentado dado por

$$\dot{\tilde{x}} = (\tilde{A} + \tilde{B}\tilde{K})\tilde{x} + Er, \tag{4.23}$$

possua todos os polos de malha fechada no interior de uma região circular de raio ℓ e centro em $(-\epsilon,0)$, conforme mostrado na Figura 4.3.

4.4 Condições LMI para Alocação Robusta de Polos

Após a definição da estrutura de controle a ser utilizada para o seguimento de referência, passa-se ao estudo de técnicas para síntese dos controladores lineares.

Considere um sistema linear incerto contínuo no tempo descrito por

$$\dot{x}(t) = \tilde{A}(\delta)x(t) + \tilde{B}(\delta)u(t), \tag{4.24}$$

com $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $\tilde{A}(\delta) \in \mathbb{R}^{n\times n}$, $\tilde{B}(\delta) \in \mathbb{R}^m$. Suponha que as matrizes $\tilde{A}(\delta)$ e $\tilde{B}(\delta)$ pertençam a um politopo $\mathcal{A}$ dado por

$$\mathcal{A} = \left\{ (\tilde{A},\tilde{B})(\delta) \in \mathbb{R}^{n\times(n\times m)} | (\tilde{A},\tilde{B})(\delta) = \sum_{j=1}^{N} \delta_j (\tilde{A},\tilde{B})_j ; \sum_{j=1}^{N} \delta_j = 1; \delta_j \geq 0 \right\}, \tag{4.25}$$

sendo $(\tilde{A},\tilde{B})_j$, $j = 1,\ldots,N$ os vértices desse politopo.

O objetivo aqui é determinar um ganho $\tilde{K}$ para a lei de realimentação de estados

$$u(t) = \tilde{K}x(t), \tag{4.26}$$

de forma a garantir que o sistema incerto de malha fechada

$$A_{cl}(\delta) = \tilde{A}(\delta) + \tilde{B}(\delta)\tilde{K} \tag{4.27}$$

tenha todos os polos de malha fechada no interior de um círculo de raio ℓ e centro $(-\epsilon,0)$, mostrado na Figura 4.3 para todo $(\tilde{A},\tilde{B})(\delta) \in \mathcal{A}$.

Para um sistema autônomo precisamente conhecido, uma condição necessária e suficiente para que todos os polos de $A_{cl}(\delta)$ localizem-se dentro da região circular mostrada na Figura 4.3 é dada pela existência de uma matriz simétrica definida positiva W de forma que (veja (Haddad e Bernstein, 1992))

$$(A_{cl}(\delta) + d\mathbf{I})W + W(A_{cl}(\delta) + d\mathbf{I})^T + \frac{1}{\ell}(A_{cl}(\delta) + d\mathbf{I})W(A_{cl}(\delta) + d\mathbf{I})^T < 0, \tag{4.28}$$

com

$$d = \epsilon - \ell. \tag{4.29}$$

A desigualdade (4.28) é obtida através da função de Lyapunov modificada

$$AW + WA^T + \frac{1}{\ell}AWA^T < 0, \tag{4.30}$$

com a matriz dinâmica de malha aberta A sendo substituída por $A_{cl}(\delta) + d\mathbf{I}$. Neste caso, $d > 0$ garante que todos os polos de malha fechada possuam parte real menor que $-d$.

Se $A_{cl}(\delta)$ pertence a um politopo incerto $\mathcal{A}$ descrito por seus vértices, uma condição suficiente para garantir a alocação desejada de polos para todo $A_{cl}(\delta) \in \mathcal{A}$ é dada pela existência de uma mesma matriz $W = W^T > 0$ satisfazendo (4.28) nos vértices de $\mathcal{A}$, o qual é uma abordagem baseada na estabilidade quadrática.

Lema 4.1 (Estabilidade Quadrática) *Se existir uma matriz $Z \in \mathbb{R}^{m\times n}$ e uma matriz simétrica definida positiva $W \in \mathbb{R}^{n\times n}$ tais que*

$$\begin{bmatrix} (A_jW + WA_j^T + B_jZ + Z^TB_j^T + 2dW) & A_jW + B_jZ + dW \\ WA_j^T + Z^TB_j^T + dW & -\ell W \end{bmatrix} < 0; \qquad j = 1,\ldots,N, \tag{4.31}$$

então $K = ZW^{-1}$ garante a alocação de polos para o sistema incerto de malha fechada na região especificada na Figura 4.3 para todo $(\tilde{A},\tilde{B})(\delta) \in \mathcal{A}$.

A seguir apresenta-se a prova para o Lema 4.1, a qual foi retirada de (Montagner et al., 2006).

Prova: Multiplicando-se a equação (4.31) por δ_j, $\delta_j \geq 0$, $\sum_{j=1}^{N} \delta_j = 1$, e somando-se em $j = 1,\ldots,N$, obtém-se

$$\begin{bmatrix} (A(\delta)W + WA(\delta)^T + B(\delta)Z + Z^TB(\delta)^T + 2dW) & A(\delta)W + B(\delta)Z + dW \\ WA(\delta)^T + Z^TB(\delta)^T + dW & -\ell W \end{bmatrix} < 0. \tag{4.32}$$

Aplicando-se o complemento de Schur (veja Apêndice B.2) na equação (4.32), tem-se

$$\begin{aligned} (A(\delta)W &+ WA(\delta)^T + B(\delta)Z + Z^TB(\delta)^T + 2dW) \\ &- (A(\delta)W + B(\delta)Z + dW)\left(\frac{-W^{-1}}{\ell}\right)(WA(\delta)^T + Z^TB(\delta)^T + dW) < 0. \end{aligned} \tag{4.33}$$

Substituindo $Z = KW$ e após algumas manipulações matemáticas

$$\begin{aligned} (A(\delta) &+ B(\delta)K + d\mathbf{I})W + W(A(\delta) + B(\delta)K + d\mathbf{I})^T \\ &+ \frac{1}{\ell}(A(\delta) + B(\delta)K + d\mathbf{I})W(A(\delta) + B(\delta)K + d\mathbf{I})^T < 0. \end{aligned} \tag{4.34}$$

Com $A_{cl}(\delta) = A(\delta) + B(\delta)K$, a equação (4.34) pode então ser escrita como

$$(A_{cl}(\delta) + d\mathbf{I})W + W(A_{cl}(\delta) + d\mathbf{I})^T + \frac{1}{\ell}(A_{cl}(\delta) + d\mathbf{I})W(A_{cl}(\delta) + d\mathbf{I})^T < 0,$$

garantindo a alocação de polos no círculo indicado na Figura 4.3, conforme equação (4.28), para qualquer valor de $\delta \in \mathbb{R}^N$ satisfazendo $\sum_{j=1}^{N} \delta_j = 1$, ou seja, para qualquer sistema pertencente ao politopo $\mathcal{A}$ dado em (4.25). ∎

Na literatura, várias extensões da condição de estabilidade quadrática têm sido utilizadas para controle robusto e projeto de filtros, incluindo especificações para alocação de polos (Chilali et al., 1999; Garcia e Bernussou, 1995; Palhares e Peres, 1999). Contudo, estabilizabilidade quadrática pode levar a resultados conservadores no sentido de que exitem especificações para alocação de polos que não podem ser alcançadas por meio de um ganho de realimentação de estados baseado na estabilização quadrática.

Uma condição para a análise da estabilidade robusta de sistemas incertos discretos no tempo foi proposta em (Oliveira et al., 1999). A vantagem dessa condição em relação ao resultado baseado na estabilizabilidade quadrática está na utilização de uma matriz de Lyapunov dependente de parâmetros. Uma extensão desse resultado para a alocação robusta de polos através da realimentação de estados é apresentada em (Leite e Peres, 2005), que produz resultados menos conservadores do que os do Lema 4.1. Note que o uso de uma função de Lyapunov dependente de parâmetros $W(\delta) = W(\delta)^T > 0$, de modo que

$$(A_{cl}(\delta) + d\mathbf{I})W(\delta) + W(\delta)(A_{cl}(\delta) + d\mathbf{I})^T + \frac{1}{\ell}(A_{cl}(\delta) + d\mathbf{I})W(\delta)(A_{cl}(\delta) + d\mathbf{I})^T < 0, \qquad (4.35)$$

pode fornecer resultados menos conservadores do que os obtidos com $W(\delta) = W$ fixa.

Lema 4.2 (Espaço Aumentado) *Se existirem N matrizes simétricas definidas positivas $P_j \in \mathbb{R}^{n\times n}$ e matrizes $L \in \mathbb{R}^{m\times n}$ tais que*

$$\begin{bmatrix} -\ell P_j & A_jG + B_jL + \epsilon G \\ G^TA_j^T + L^TB_j^T + \epsilon G^T & \ell(P_j - G - G^T) \end{bmatrix} < 0; \qquad j = 1,\ldots,N, \qquad (4.36)$$

então $K = LG^{-1}$ é um ganho de realimentação de estados robusto e $P(\delta)$, dado por

$$P(\delta) = \sum_{j=1}^{N} \delta_j P_j; \qquad \delta_j \geq 0; \qquad \sum_{j=1}^{N} \delta_j = 1, \qquad (4.37)$$

é uma matriz de Lyapunov dependente de parâmetro que garante ao sistema incerto de malha fechada a alocação de polos especificada na Figura 4.3 para todo $(\tilde{A},\tilde{B})(\delta) \in \mathcal{A}$.

A prova para o Lema 4.2, retirada de (Leite e Peres, 2005), é apresentada a seguir.

Prova: Note que, com $L = KG$, (4.36) é equivalente a

$$\begin{bmatrix} -P_j & (A_j + B_jK + \epsilon\mathbf{I})G/\ell \\ G^T(A_j + B_jK + \epsilon\mathbf{I})^T/\ell & P_j - G - G^T \end{bmatrix} < 0; \qquad j = 1, \ldots, N. \tag{4.38}$$

Multiplicando-se a equação (4.38) por δ_j, $\delta_j \geq 0$, $\sum_{j=1}^{N} \delta_j = 1$, e somando-se em $j = 1, \ldots, N$, obtém-se

$$\begin{bmatrix} -P(\delta) & (A(\delta) + B(\delta)K + \epsilon\mathbf{I})G/\ell \\ G^T(A(\delta) + B(\delta)K + \epsilon\mathbf{I})^T/\ell & P(\delta) - G - G^T \end{bmatrix} < 0. \tag{4.39}$$

Multiplicada à esquerda por

$$T = [\mathbf{I} \;\; (A(\delta) + B(\delta)K + \epsilon\mathbf{I})/\ell], \tag{4.40}$$

e à direita por T^T (veja (Oliveira et al., 1999) para detalhes), implica em

$$\frac{1}{\ell^2}(A(\delta) + B(\delta)K + \epsilon\mathbf{I})P(\delta)(A(\delta) + B(\delta)K + \epsilon\mathbf{I})^T - P(\delta) < 0, \tag{4.41}$$

que, por sua vez, com $\epsilon = \ell + d$, escreve-se

$$\begin{aligned} \frac{1}{\ell^2}(A(\delta) \;&+\; B(\delta)K + d\mathbf{I})P(\delta)(A(\delta) + B(\delta)K + d\mathbf{I})^T \\ &+\; \frac{1}{\ell}(A(\delta) + B(\delta)K + d\mathbf{I})P(\delta) + \frac{1}{\ell}P(\delta)(A(\delta) + B(\delta)K + d\mathbf{I})^T < 0. \end{aligned} \tag{4.42}$$

Substituindo-se $W(\delta) = \dfrac{P(\delta)}{\ell}$, tem-se

$$\begin{aligned} (A(\delta) \;&+\; B(\delta)K + d\mathbf{I})W(\delta) + W(\delta)(A(\delta) + B(\delta)K + d\mathbf{I})^T \\ &+\; \frac{1}{\ell}(A(\delta) + B(\delta)K + d\mathbf{I})W(\delta)(A(\delta) + B(\delta)K + d\mathbf{I})^T < 0. \end{aligned} \tag{4.43}$$

Com $A_{cl}(\delta) = A(\delta) + B(\delta)K$, a equação (4.43) pode então ser escrita como

$$(A_{cl}(\delta) + d\mathbf{I})W(\delta) + W(\delta)(A_{cl}(\delta) + d\mathbf{I})^T + \frac{1}{\ell}(A_{cl}(\delta) + d\mathbf{I})W(\delta)(A_{cl}(\delta) + d\mathbf{I})^T < 0,$$

garantindo, com a função de Lyapunov dependente de parâmetros $W(\delta)$, a alocação de polos no círculo indicado na Figura 4.3, conforme equação (4.35), para valor de $\delta \in \mathbb{R}^N$ satisfazendo $\sum_{j=1}^{N} \delta_j = 1$, ou seja, para qualquer sistema pertencente ao politopo $\mathcal{A}$ dado em (4.25). ■

Os Lemas 4.1 e 4.2 apresentam condições suficientes, formuladas em termos de desigualdades matriciais lineares, para a obtenção de um ganho fixo que garante a alocação robusta de polos. A condição apresentada no Lema 4.2 fornece resultados menos conservadores que os obtidos através do Lema 4.1 (baseados na estabilizabilidade quadrática), devido à utilização de uma matriz de Lyapunov dependente de parâmetros.

4.5 Experimentos Numéricos

Nesta seção são apresentados os experimentos numéricos realizados utilizando as matrizes A_r e B_r dadas pelo modelo linearizado pela técnica de realimentação linearizante robusta (4.21) substituídas no sistema aumentado apresentado na equação (4.22). Os parâmetros nominais para esse modelo, extraídos de (Franco, 2006a), são apresentados na Tabela 4.1.

Tabela 4.1: Parâmetros nominais do modelo dinâmico. (Fonte: (Franco, 2006a)).

Parâmetro	Valor	Unid	Descrição
K_t	0,0059	$N \times m/A$	Constante de torque dos atuadores
n	$19:1$	-	Fator de redução dos atuadores para as rodas
V_{ref}	12	V	Tensão de referência
R_a	1,71	Ohm	Resistência de armadura dos atuadores
r	0,024	m	Raio da roda
J_o	$3{,}88 \times 10^{-7}$	kgm^2	Momento de inércia dos atuadores
J_w	2,125	kgm^2	Momento de inércia do robô
b_o	$2{,}4 \times 10^{-6}$	$N \times m \times s/rad$	Constante de atrito viscoso dos atuadores
K_{em}	0,0059	$V \times s/rad$	Constante de força contra eletromotriz dos atuadores
L	0,09	m	Distância entre a roda e o centro do robô
m	3,4	Kg	Massa do robô
φ_d	$45°$	graus	Ângulo dianteiro do robô
φ_t	$45°$	graus	Ângulo traseiro do robô

Após a substituição dos parâmetros, as matrizes $\tilde{A}$ e $\tilde{B}$ do modelo nominal aumentado para o robô F180 são dadas por (4.44). Veja que, conforme a topologia de controle adotada, esse modelo possui a inclusão de três integrados, sendo um para cada erro de seguimento de referência das variáveis $\dot{X}_m$, $\dot{Y}_m$ e $\dot{\theta}$. Além disso, o par $(\tilde{A},\tilde{B})$ é controlável.

$$\tilde{A} = \begin{bmatrix} -7{,}3398 & 0 & 0 & 0 & 0 & 0 \\ 0 & -7{,}3398 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0{,}2167 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \tilde{B} = \begin{bmatrix} 5{,}9638 & 0 & 0 \\ 0 & 5{,}9638 & 0 \\ 0 & 0 & 1{,}3831 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.44}$$

Aplica-se então, as técnicas apresentadas nos Lemas 4.1 e 4.2 no projeto de uma lei de realimentação de estados para o sistema (4.44). Apesar das incertezas poderem estar presentes em quaisquer dos parâmetros do robô, apenas os parâmetros associados aos atuadores do robô, K_t, constante de torque, K_{em}, constante de força contra eletromotriz e b_o, coeficiente de atrito viscoso, foram considerados incertos, isto é, os valores desses parâmetros podem variar em torno de seus valores nominais mostrados na Tabela 4.1. Assume-se que esses parâmetros possam variar em $\pm 20\%$. A razão para a escolha desses parâmetros está na forma como eles aparecem no modelo matemático, ou seja, eles aparecem de forma linear. A escolha de parâmetros apresentados de forma não linear dificultaria a geração do politopo de incertezas.

Note que os parâmetros K_{em} e b_o afetam apenas a matriz dinâmica, enquanto que K_t afeta também a matriz de entrada conforme equação (4.21). O intervalo numérico para variação paramétrica é dado por

$$\begin{aligned} 0{,}0047 \leq \quad K_t \quad &\leq 0{,}0071, \\ 0{,}0047 \leq \quad K_{em} \quad &\leq 0{,}0071, \\ 0{,}1920 \times 10^{-5} \leq \quad b_o \quad &\leq 0{,}2880 \times 10^{-5}. \end{aligned} \tag{4.45}$$

Realizando um combinação dos valores extremos dos parâmetros considerados incertos dados pela equação (4.45) e aplicando ao sistema aumentado (4.22), é produzido um sistema incerto formado por oito vértices dado pelas matrizes

$$\tilde{A}_1 = \begin{bmatrix} -4{,}8213 & 0 & 0 & 0 & 0 & 0 \\ 0 & -4{,}8213 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0{,}1423 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad \tilde{B}_1 = \begin{bmatrix} 4{,}7710 & 0 & 0 \\ 0 & 4{,}7710 & 0 \\ 0 & 0 & 1{,}1065 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.46}$$

$$\tilde{A}_2 = \begin{bmatrix} -5{,}1309 & 0 & 0 & 0 & 0 & 0 \\ 0 & -5{,}1309 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0{,}1515 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad \tilde{B}_2 = \begin{bmatrix} 4{,}7710 & 0 & 0 \\ 0 & 4{,}7710 & 0 \\ 0 & 0 & 1{,}1065 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.47}$$

$$\tilde{A}_3 = \begin{bmatrix} -6{,}9223 & 0 & 0 & 0 & 0 & 0 \\ 0 & -6{,}9223 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0{,}2043 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad \tilde{B}_3 = \begin{bmatrix} 4{,}7710 & 0 & 0 \\ 0 & 4{,}7710 & 0 \\ 0 & 0 & 1{,}1065 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.48}$$

$$\tilde{A}_4 = \begin{bmatrix} -7{,}2320 & 0 & 0 & 0 & 0 & 0 \\ 0 & -7{,}2320 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0{,}2135 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad \tilde{B}_4 = \begin{bmatrix} 4{,}7710 & 0 & 0 \\ 0 & 4{,}7710 & 0 \\ 0 & 0 & 1{,}1065 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.49}$$

$$\tilde{A}_5 = \begin{bmatrix} -6{,}9223 & 0 & 0 & 0 & 0 & 0 \\ 0 & -6{,}9223 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0{,}2043 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad \tilde{B}_5 = \begin{bmatrix} 7{,}1566 & 0 & 0 \\ 0 & 7{,}1566 & 0 \\ 0 & 0 & 1{,}6597 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.50}$$

$$\tilde{A}_6 = \begin{bmatrix} -7{,}2320 & 0 & 0 & 0 & 0 & 0 \\ 0 & -7{,}2320 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0{,}2135 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \tilde{B}_6 = \begin{bmatrix} 7{,}1566 & 0 & 0 \\ 0 & 7{,}1566 & 0 \\ 0 & 0 & 1{,}6597 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.51)$$

$$\tilde{A}_7 = \begin{bmatrix} -10{,}0739 & 0 & 0 & 0 & 0 & 0 \\ 0 & -10{,}0739 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0{,}2974 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \tilde{B}_7 = \begin{bmatrix} 7{,}1566 & 0 & 0 \\ 0 & 7{,}1566 & 0 \\ 0 & 0 & 1{,}6597 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.52)$$

$$\tilde{A}_8 = \begin{bmatrix} -10{,}3835 & 0 & 0 & 0 & 0 & 0 \\ 0 & -10{,}3835 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0{,}3065 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \tilde{B}_8 = \begin{bmatrix} 7{,}1566 & 0 & 0 \\ 0 & 7{,}1566 & 0 \\ 0 & 0 & 1{,}6597 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.53)$$

Após obter-se o sistema nominal e os oito vértices que formam o politopo de incertezas $\mathcal{A}$, passa-se ao projeto dos controladores. A intenção é selecionar controladores com desempenho semelhante no caso nominal e avaliar o desempenho na presença de incertezas paramétricas.

4.5.1 Síntese dos Controladores

Nesta seção, são apresentados os controladores obtidos pela utilização dos Lemas 4.1 e 4.2. Como referência para comparação, primeiramente, é realizada a síntese de um controlador utilizando-se a técnica de alocação exata de polos e, para tal, foi utilizado o comando `Place` do MATLAB . Note que o sistema nominal (4.44) é BIBO estável, com polos em 0, $-7{,}3398$ e $-0{,}2167$. Além disso, o sistema possui polos lentos (próximos à origem). Deseja-se alocar os polos do sistema de malha fechada de forma a garantir uma dinâmica rápida o suficiente e ao mesmo tempo limitar a intensidade da ação de controle, evitando variações bruscas que poderiam causar saturação no sinal de controle. O conjunto de polos escolhido é mostrado na equação (4.54).

$$P_d = \begin{bmatrix} -14+7i & -14-7i & -15 & -25 & -30 & -33 \end{bmatrix}. \quad (4.54)$$

O ganho obtido pela aplicação do comando `Place` ao sistema nominal (4.44) é dado por

$$K_p = \begin{bmatrix} -4{,}0515 & -4{,}3280 & -6{,}3858 & 50{,}8298 & 84{,}0753 & 126{,}8835 \\ -2{,}4933 & -4{,}5817 & 2{,}3486 & 31{,}0759 & 66{,}7305 & -9{,}2637 \\ 5{,}8113 & -9{,}7887 & -46{,}7184 & -65{,}0259 & 178{,}1385 & 592{,}2449 \end{bmatrix}. \quad (4.55)$$

Apesar de fornecer uma especificação exata para alocação de polos do sistema nominal, o ganho (4.55) não é capaz de lidar adequadamente com as incertezas dos parâmetros. A Figura 4.4 mostra os polos de malha fechada para o sistema incerto $\tilde{A}(\delta) + \tilde{B}(\delta)K_p$ fora de suas especificações nominais.

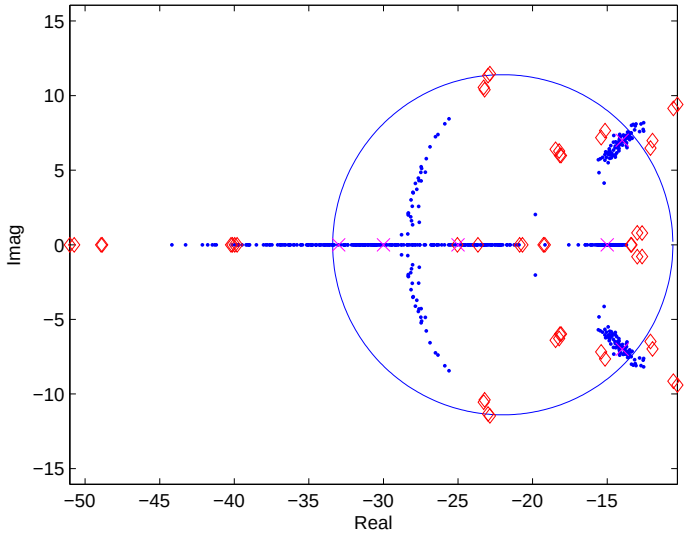


Figura 4.4: Polos de $\tilde{A}(\delta) + \tilde{B}(\delta)K_p$ com K_p dado por (4.55) e os vértices do sistema incerto por (4.46)-(4.53). Os polos dos vértices do sistema de malha fechada são representados por $\diamond$ e os polos do sistema nominal por $\times$.

Um círculo centrado em $(-22{,}0)$ com raio $\ell = 11{,}4$ foi incluído, uma vez que esse é o menor círculo atingido pelo controle baseado em estabilidade quadrática apresentado pelo Lema 4.1. Essa região circular foi escolhida de forma que o círculo englobe os autovalores em (4.54).

O teste de factibilidade utilizando os Lemas 4.1 e 4.2 geram os ganhos mostrados nas equações (4.56) e (4.57) respectivamente.

$$K_{L1} = \begin{bmatrix} -4{,}6884 & 0{,}0000 & 0{,}0000 & 50{,}1373 & 0{,}0000 & 0{,}0000 \\ 0{,}0000 & -4{,}6884 & 0{,}0000 & 0{,}0000 & 50{,}1373 & 0{,}0000 \\ 0{,}0000 & 0{,}0000 & -24{,}6676 & 0{,}0000 & 0{,}0000 & 204{,}4187 \end{bmatrix}. \tag{4.56}$$

$$K_{L2} = \begin{bmatrix} -4{,}7319 & 0{,}0000 & 0{,}0000 & 51{,}2458 & 0{,}0000 & 0{,}0000 \\ 0{,}0000 & -4{,}7319 & 0{,}0000 & 0{,}0000 & 51{,}2458 & 0{,}0000 \\ 0{,}0000 & 0{,}0000 & -26{,}1235 & 0{,}0000 & 0{,}0000 & 231{,}2998 \end{bmatrix}. \tag{4.57}$$

Veja que a diferença entre os valores dos ganhos dados por (4.56) e (4.57) é pequena. Isso sugere que a alocação dos polos do sistema incerto seja semelhante nos dois casos. As Figuras 4.5 e 4.6 mostram a alocação dos polos do sistema em malha fechada para ganhos dados por (4.56) e (4.57) respectivamente.

Analisando as Figuras 4.5 e 4.6, fica claro que tanto o ganho robusto obtido pelo Lema 4.1 quanto aquele obtido pelo Lema 4.2 conseguem alocar os polos do sistema incerto de malha fechada dentro da região circular centrada em $(-22{,}0)$ e com raio $\ell = 11{,}4$. No entanto, se a especificação para alocação dos polos é mais rigorosa, o Lema 4.1 torna-se infactível. Para

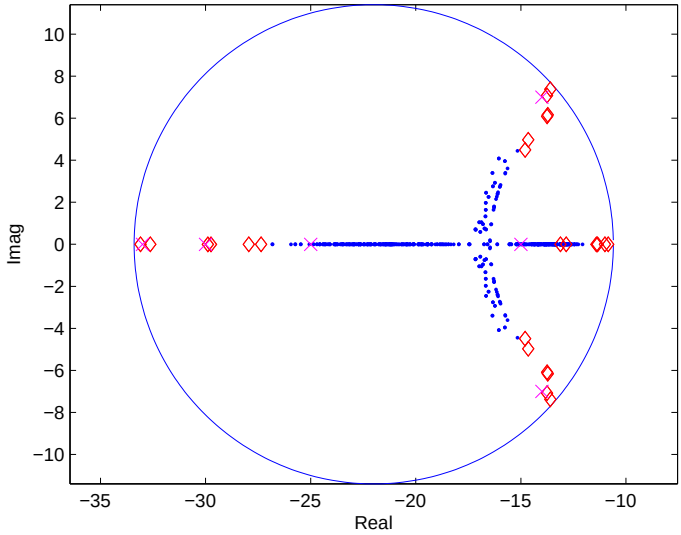


Figura 4.5: Polos de $\tilde{A}(\delta) + \tilde{B}(\delta)K_{L1}$ com $\epsilon = 22$, $\ell = 11{,}4$, K_{L1} dado por (4.56) e os vértices do sistema incerto por (4.46)-(4.53). Os polos dos vértices do sistema de malha fechada são representados por $\diamond$ e os polos do sistema nominal por $\times$.

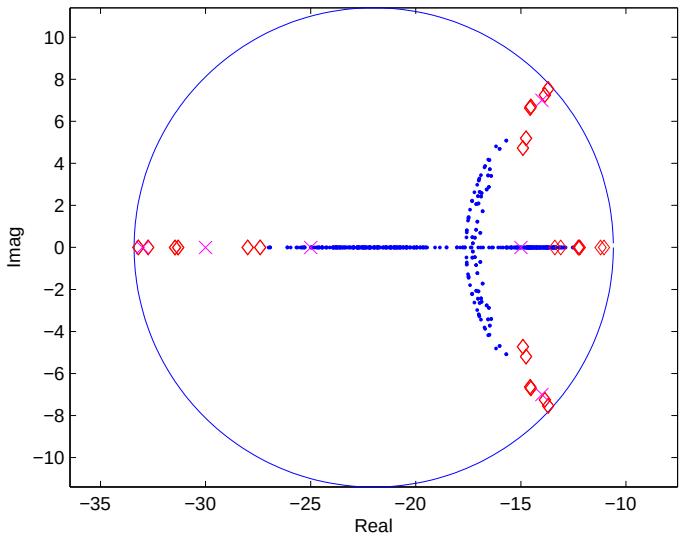


Figura 4.6: Polos de $\tilde{A}(\delta) + \tilde{B}(\delta)K_{L2}$ com $\epsilon = 22$, $\ell = 11{,}4$, K_{L2} dado por (4.57) e os vértices do sistema incerto por (4.46)-(4.53). Os polos dos vértices do sistema de malha fechada são representados por $\diamond$ e os polos do sistema nominal por $\times$.

mostrar essa limitação do método, escolhe-se $\ell = 11{,}2$, que representa o menor raio que garante que o controle apresentado pelo Lema 4.2 seja factível. A equação (4.58) apresenta o novo valor para o ganho obtido pelo Lema 4.2 e a Figura 4.7 apresenta os polos de malha fechada do

sistema incerto utilizando o ganho dado por (4.58).

$$\bar{K}_{L2} = \begin{bmatrix} -4{,}7049 & 0{,}0000 & 0{,}0000 & 50{,}4576 & 0{,}0000 & 0{,}0000 \\ 0{,}0000 & -4{,}7049 & 0{,}0000 & 0{,}0000 & 50{,}4576 & 0{,}0000 \\ 0{,}0000 & 0{,}0000 & -26{,}1942 & 0{,}0000 & 0{,}0000 & 232{,}6620 \end{bmatrix}. \tag{4.58}$$

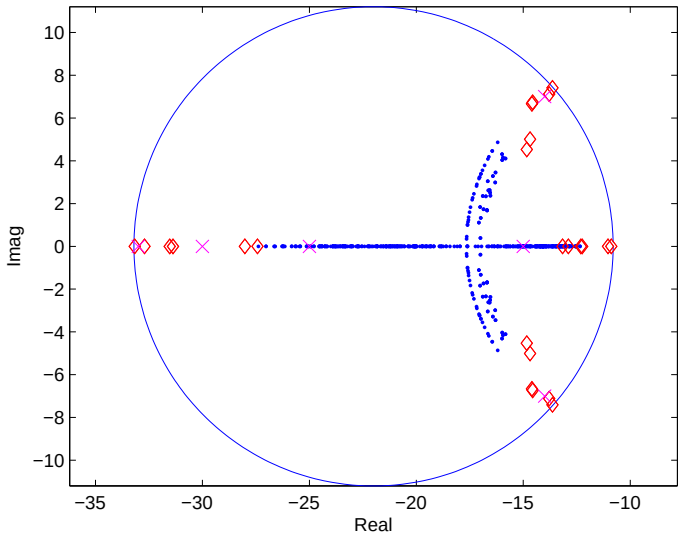
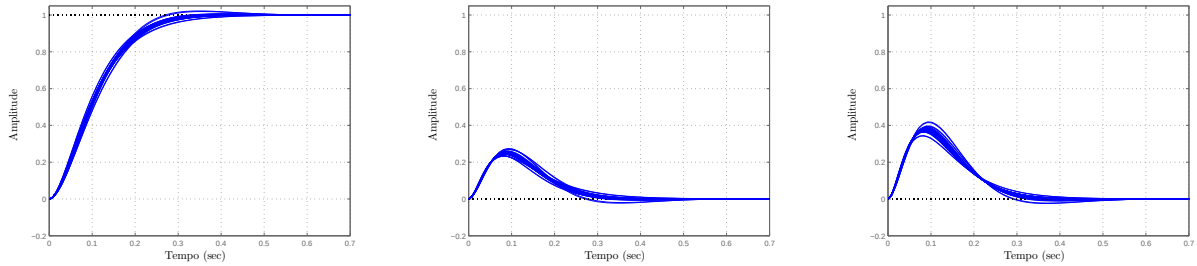


Figura 4.7: Polos de $\tilde{A}(\delta) + \tilde{B}(\delta)\bar{K}_{L2}$ com $\epsilon = 22$, $\ell = 11{,}2$, $\bar{K}_{l2}$ dado por (4.57) e os vértices do sistema incerto por (4.46)-(4.53). Os polos dos vértices do sistema de malha fechada são representados por $\diamond$ e os polos do sistema nominal por $\times$.

Os Lemas 4.1 e 4.2 garantem que a alocação de polos na região especificada é preservada para qualquer valor de parâmetros no intervalo mostrado em (4.45) se uma especificação conservadora é utilizada, por outro lado, para uma especificação menos conservadora, apenas o Lema 4.2 é factível. No entanto, é importante notar que, o ganho dado por (4.58) apresenta diferenças pequenas em relação ao ganho dado por (4.57). Isso mostra que, para o problema apresentado neste trabalho, a síntese de controladores baseada na estabilidade quadrática e no sistema aumentado com função de Lyapunov dependente de parâmetros apresentam resultados semelhantes de forma que ambas podem ser utilizadas para o intervalo de incertezas paramétricas dado por (4.45). Observe também que, se a alocação é realizada utilizando o comando `Place` do MATLAB , mesmo uma pequena mudança no valor de algum parâmetro poderia causar o deslocamento dos polos de malha fechada para fora da região desejada, conforme mostrado na Figura 4.4.

Conforme citado anteriormente nesta seção, os polos de malha fechada do sistema utilizando o comando `Place` do MATLAB foram especificados para que o sistema possua uma dinâmica rápida e ao mesmo tempo para limitar a intensidade da ação de controle. São mostradas na Figura 4.8 as respostas dos estados $\dot{X}_m$, $\dot{Y}_m$ e $\dot{\theta}$ para uma entrada em degrau unitário aplicada em r para diversos valores dos parâmetros incertos dentro dos intervalos (4.45) para o controlador dado por (4.55). Veja que a resposta temporal mostra que o sistema em malha fechada é acoplado, diferentemente do sistema nominal dado por (4.21), que é desacoplado. Além disso, as

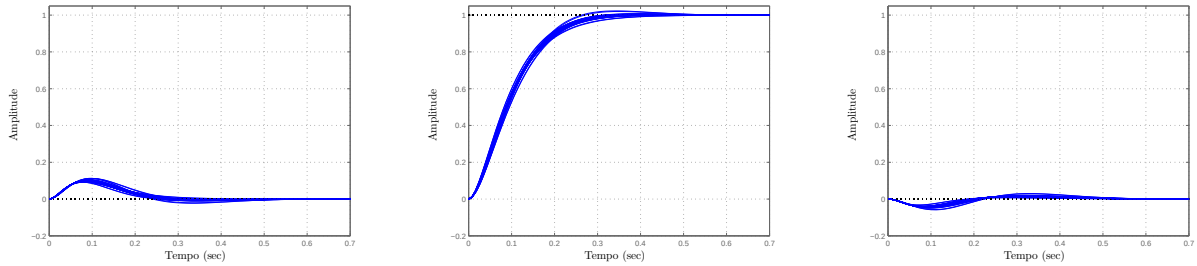
respostas dos estados $\dot{X}_m$, $\dot{Y}_m$ e $\dot{\theta}$ referentes às entradas r_1, r_2 e r_3 respectivamente, apresentam um sobressinal devido às incertezas do sistema.



(a) Resposta de $\dot{X}_m$ para o degrau unitário aplicado em r_1.

(b) Resposta de $\dot{X}_m$ para o degrau unitário aplicado em r_2.

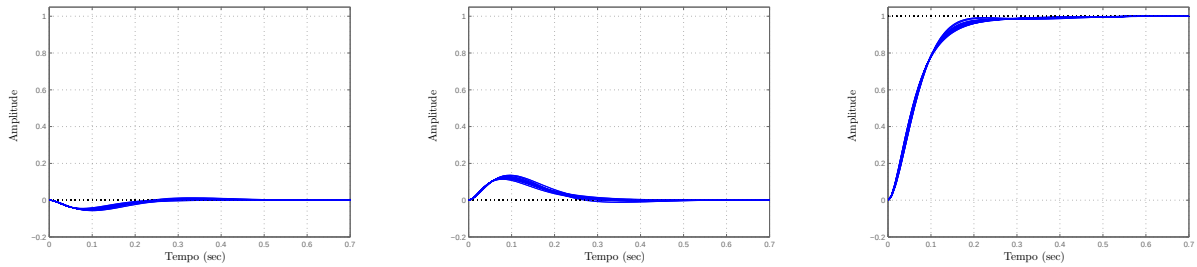
(c) Resposta de $\dot{X}_m$ para o degrau unitário aplicado em r_3.



(d) Resposta de $\dot{Y}_m$ para o degrau unitário aplicado em r_1.

(e) Resposta de $\dot{Y}_m$ para o degrau unitário aplicado em r_2.

(f) Resposta de $\dot{Y}_m$ para o degrau unitário aplicado em r_3.



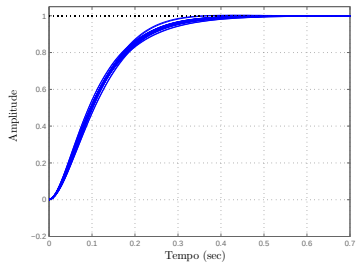
(g) Resposta de $\dot{\theta}$ para o degrau unitário aplicado em r_1.

(h) Resposta de $\dot{\theta}$ para o degrau unitário aplicado em r_2.

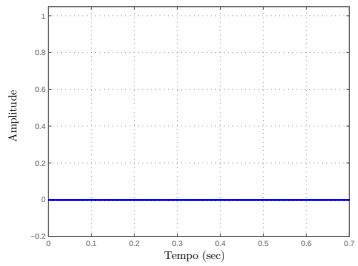
(i) Resposta de $\dot{\theta}$ para o degrau unitário aplicado em r_3.

Figura 4.8: Resposta temporal dos estados $\dot{X}_m$, $\dot{Y}_m$ e $\dot{\theta}$ para uma entrada em degrau unitário aplicada em r para diversos valores dos parâmetros incertos dentro do intervalo (4.45) para o controlador utilizando o comando `Place` dado por (4.55).

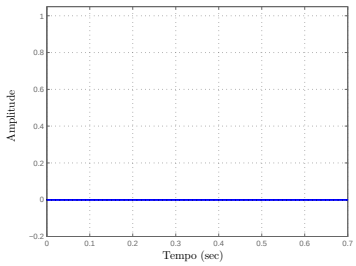
As Figuras 4.9 e 4.10 apresentam as respostas temporais para o sistema em malha fechada utilizando os ganhos (4.56) e (4.57) respectivamente. Note que o sistema agora é desacoplado e não apresenta sobressinal em ambos os casos. Isso mostra que esses controladores exigirão menores intensidades dos sinais de controle do que o controlador dado por (4.55).



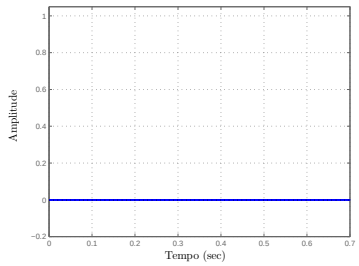
(a) Resposta de $\dot{X}_m$ para o degrau unitário aplicado em r_1.



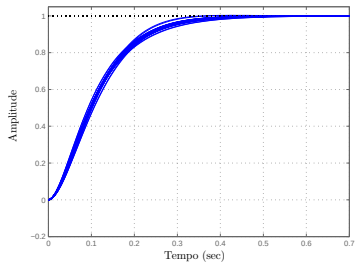
(b) Resposta de $\dot{X}_m$ para o degrau unitário aplicado em r_2.



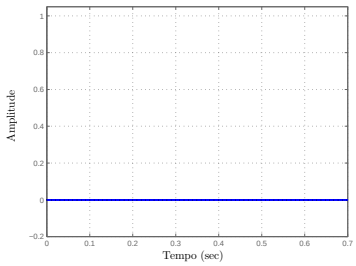
(c) Resposta de $\dot{X}_m$ para o degrau unitário aplicado em r_3.



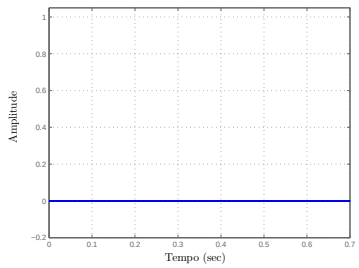
(d) Resposta de $\dot{Y}_m$ para o degrau unitário aplicado em r_1.



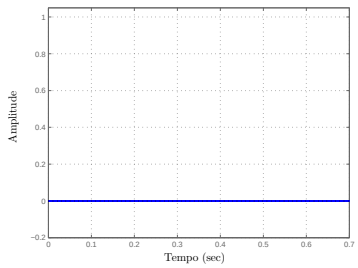
(e) Resposta de $\dot{Y}_m$ para o degrau unitário aplicado em r_2.



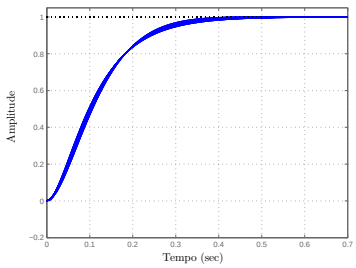
(f) Resposta de $\dot{Y}_m$ para o degrau unitário aplicado em r_3.



(g) Resposta de $\dot{\theta}$ para o degrau unitário aplicado em r_1.

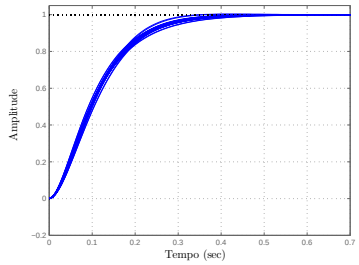


(h) Resposta de $\dot{\theta}$ para o degrau unitário aplicado em r_2.

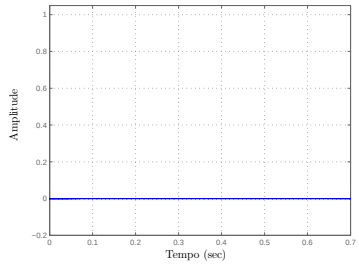


(i) Resposta de $\dot{\theta}$ para o degrau unitário aplicado em r_3.

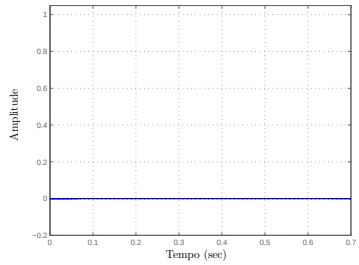
Figura 4.9: Resposta temporal dos estados $\dot{X}_m$, $\dot{Y}_m$ e $\dot{\theta}$ para uma entrada em degrau unitário aplicada em r para diversos valores dos parâmetros incertos dentro do intervalo (4.45) para o controlador utilizando o Lema 4.1 dado por (4.56).



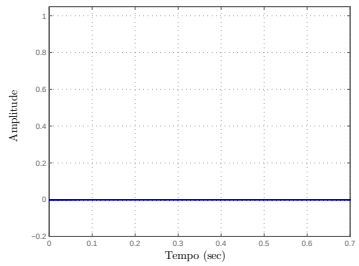
(a) Resposta de $\dot{X}_m$ para o degrau unitário aplicado em r_1.



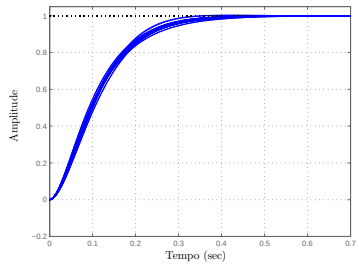
(b) Resposta de $\dot{X}_m$ para o degrau unitário aplicado em r_2.



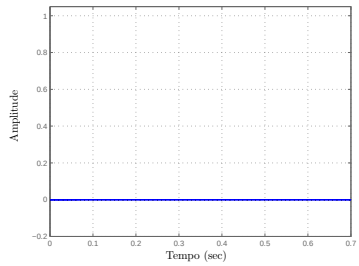
(c) Resposta de $\dot{X}_m$ para o degrau unitário aplicado em r_3.



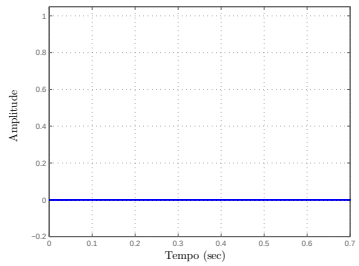
(d) Resposta de $\dot{Y}_m$ para o degrau unitário aplicado em r_1.



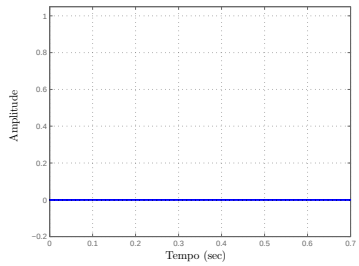
(e) Resposta de $\dot{Y}_m$ para o degrau unitário aplicado em r_2.



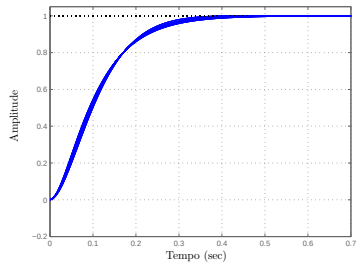
(f) Resposta de $\dot{Y}_m$ para o degrau unitário aplicado em r_3.



(g) Resposta de $\dot{\theta}$ para o degrau unitário aplicado em r_1.



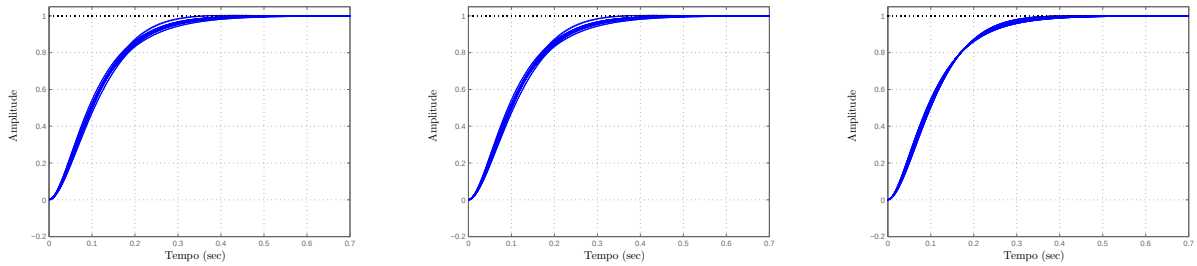
(h) Resposta de $\dot{\theta}$ para o degrau unitário aplicado em r_2.



(i) Resposta de $\dot{\theta}$ para o degrau unitário aplicado em r_3.

Figura 4.10: Resposta temporal dos estados $\dot{X}_m$, $\dot{Y}_m$ e $\dot{\theta}$ para uma entrada em degrau unitário aplicada em r para diversos valores dos parâmetros incertos dentro do intervalo (4.45) para o controlador utilizando o Lema 4.2 dado por (4.57).

A Figura 4.11 apresenta a resposta dos estados $\dot{X}_m$, $\dot{Y}_m$ e $\dot{\theta}$ referentes às entradas r_1, r_2 e r_3, respectivamente, para o controlador dado por (4.58). Como o sistema é desacoplado, as demais respostas são omitidas. Perceba que o resultado é semelhante ao apresentado na Figura 4.10.



(a) Resposta de $\dot{X}_m$ para o degrau unitário aplicado em r_1.

(b) Resposta de $\dot{Y}_m$ para o degrau unitário aplicado em r_2.

(c) Resposta de $\dot{\theta}$ para o degrau unitário aplicado em r_3.

Figura 4.11: Resposta temporal dos estados $\dot{X}_m$, $\dot{Y}_m$ e $\dot{\theta}$ para uma entrada em degrau unitário aplicada em r_1, r_2 e r_3 respectivamente, para diversos valores dos parâmetros incertos dentro do intervalo (4.45) para o controlador utilizando o Lema 4.2 dado por (4.58).

Após a análise das respostas temporais do sistema em malha fechada utilizando os controladores propostos, propõem-se alguns estudos de caso para seguimento de referência. Esses casos têm como objetivo demostrar a diferença de desempenho dos controladores na presença de incertezas paramétricas. O critério de desempenho adotado nesse trabalho é o valor máximo absoluto do erro de seguimento das velocidades passadas como referência para o sistema:

$$|e_v| = \sqrt{e_{vx}^2 + e_{vy}^2}, \tag{4.59}$$

com

$$e_{vx} = v_{rx} - v_x, \tag{4.60}$$

$$e_{vy} = v_{ry} - v_y, \tag{4.61}$$

e v_{rx}, v_{ry}, v_x e v_x as velocidades de referência e as velocidades encontradas nas direções x e y respectivamente.

De forma a quantificar a diferença entre os erros dos controladores, é também calculado o valor relativo percentual entre o máximo erro encontrado com o `Place` em relação ao máximo erro encontrado com os controladores dados pelos Lemas 4.1 e 4.2, dado pela equação (4.62). Neste caso, sempre que esta quantidade for positiva, o controlador utilizando `Place` apresenta maior erro e o valor indicado representa quantos por cento esse erro foi maior:

$$V_{rp} = \left(\frac{M_{ep}}{M_{el}} - 1\right) \times 100\ \%, \tag{4.62}$$

em que M_{ep} representa o máximo valor de erro encontrado com `Place` e M_{el} o máximo valor de erro encontrado com os controladores dados pelos Lemas 4.1 ou 4.2.

4.5.2 Estudos de Caso

Um robô móvel omnidirecional pode realizar movimentos de rotação e translação em torno de seu centro de gravidade independentemente no plano bi-dimensional. Para verificar essa propriedade, chamada de propriedade holonômica, trajetórias serão definidas realizando movimentos de translação e movimentos de translação e rotação simultâneos. É assumido que o valor inicial do sistema de coordenadas móvel S_m seja igual ao sistema de coordenadas global S_r e o valor inicial das variáveis de estado é dado por $\tilde{x} = [0\ 0\ 0\ 0\ 0\ 0]^T$.

Qualquer combinação de valores dentro do intervalo dado por (4.45) pode ser utilizada para gerar um sistema incerto para fins de simulações, então, existe um número infinito de possíveis combinações. Assim, de forma a simplificar as simulações utilizou-se apenas os vértices do politopo de incertezas $\mathcal{A}$ e, somente os piores casos dentre essas simulações serão apresentados devido ao grande número de gráficos. Ressalta-se contudo, que situações mais críticas que as apresentadas pelos vértices do politopo de incertezas podem ocorrer em alguma combinação no interior desse politopo. Oito casos são analisados: trajetória reta, trajetória reta com rotação, trajetória quadrada, trajetória quadrada com rotação, trajetória em forma de oito, trajetória em forma de oito com rotação, trajetória circular e trajetória circular com rotação.

Caso 1 - Trajetória Reta

Nesse estudo de caso, é assumido que o robô esteja movimentando-se em linha reta na direção $\Psi = 45°$ com ângulo de rotação $\theta = 0°$ durante 8 segundos. Embora a velocidade nas direções x e y seja setada para 0,3 m/s, uma onda quadrada é definida para cada componente de velocidade de forma a melhor excitar a dinâmica do sistema.

As Figuras 4.12, 4.13, 4.14, 4.15, 4.16 e 4.17 apresentam os resultados para essa trajetória e a tabela 4.2 os erros máximos para cada controlador e o valor relativo percentual do controlador utilizando `Place` em relação aos controladores utilizando os Lemas 4.1 e 4.2.

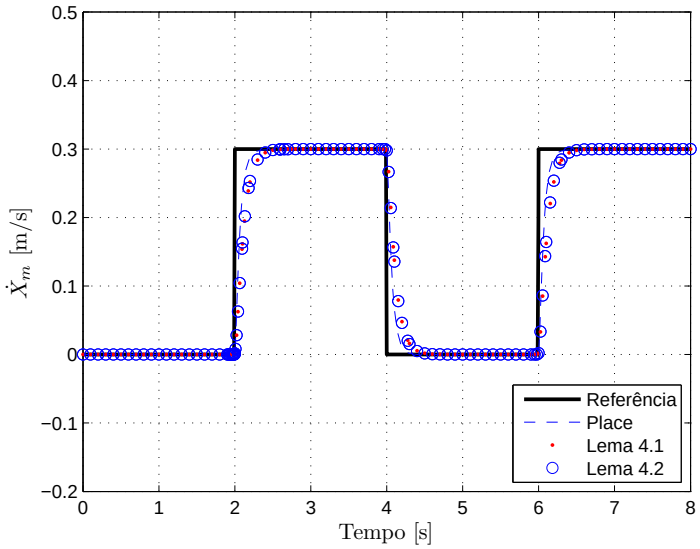


Figura 4.12: Velocidade $\dot{X}_m$ para trajetória reta sem rotação.

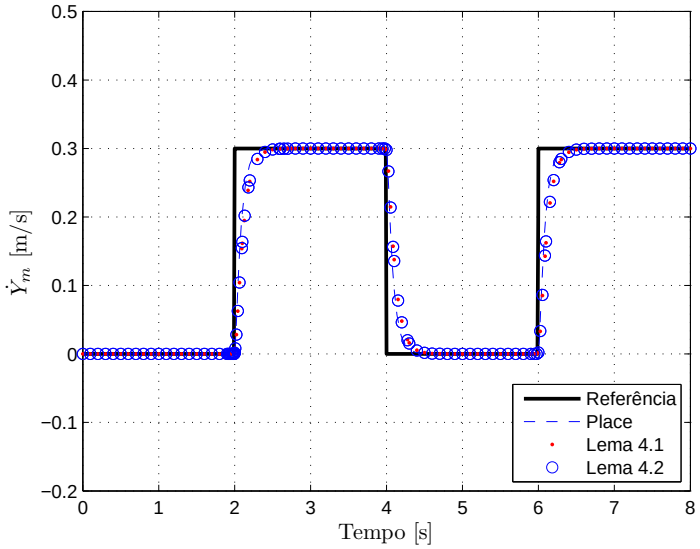


Figura 4.13: Velocidade $\dot{Y}_m$ para trajetória reta sem rotação.

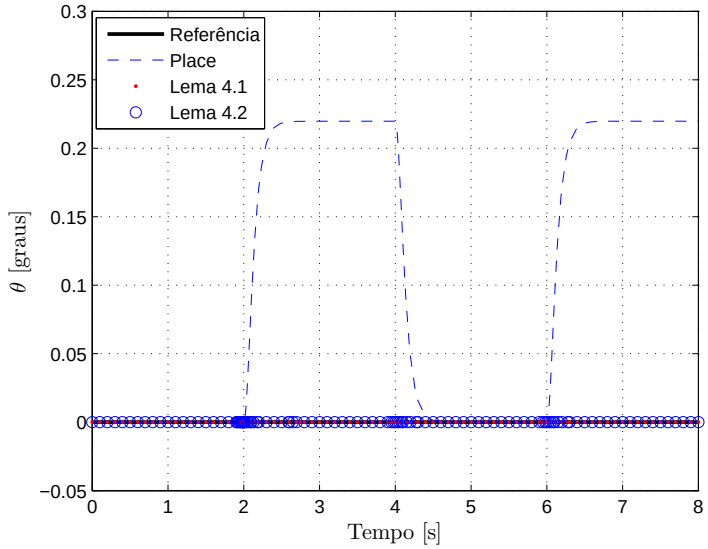


Figura 4.14: Deslocamento angular θ para trajetória reta sem rotação.

Tabela 4.2: Erros apresentados pelos controladores para uma trajetória reta sem rotação.

	Place	Lema 4.1	Lema 4.2
Erro Máximo	0,4180	0,4178	0,4140
Valor Relativo Percentual	-	0,0589	0,9764

Observe nas Figuras 4.12 e 4.13 que, para essa trajetória, o comportamento dos controladores é semelhante. A Figura 4.14 mostra o deslocamento angular do robô, veja que o resultado encontrado utilizando o comando `Place` apresenta uma rotação de aproximadamente 0,22° nos intervalos de 2 a 4 segundos e de 6 a 8 segundos. No entanto, essa rotação é muito pequena

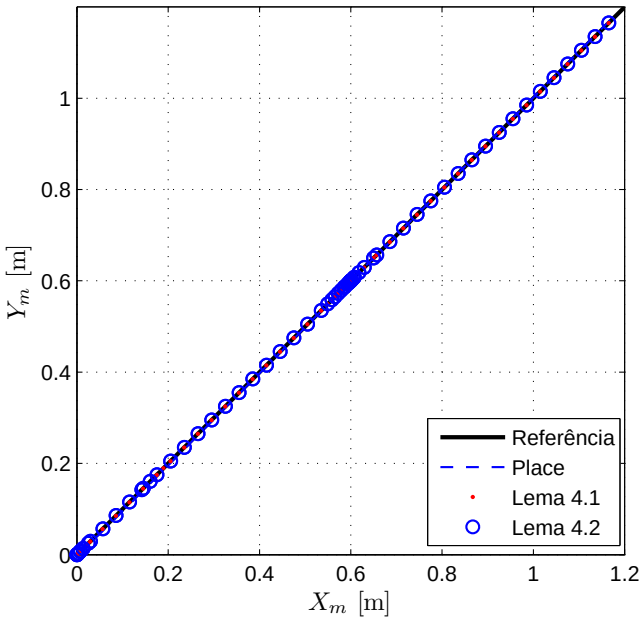


Figura 4.15: Trajetória reta sem rotação.

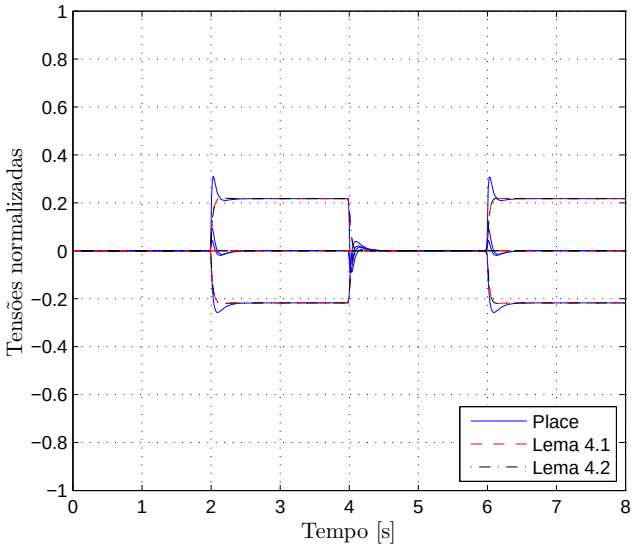


Figura 4.16: Tensões para trajetória reta sem rotação.

e não deverá causar problemas ao sistema. Percebe-se pela Figura 4.15 que os controladores conseguem manter o robô sobre a trajetória de referência, no entanto, a Figura 4.16 mostra que houve um esforço maior dos atuadores no caso do `Place`. Isso mostra que, mesmo os controladores tendo apresentado resultados semelhantes no seguimento de referência, os Lemas 4.1 e 4.2 apresentam menor consumo de energia. Observando a Figura 4.17 e a tabela 4.2, verifica-se que o desempenho dos controladores é praticamente o mesmo.

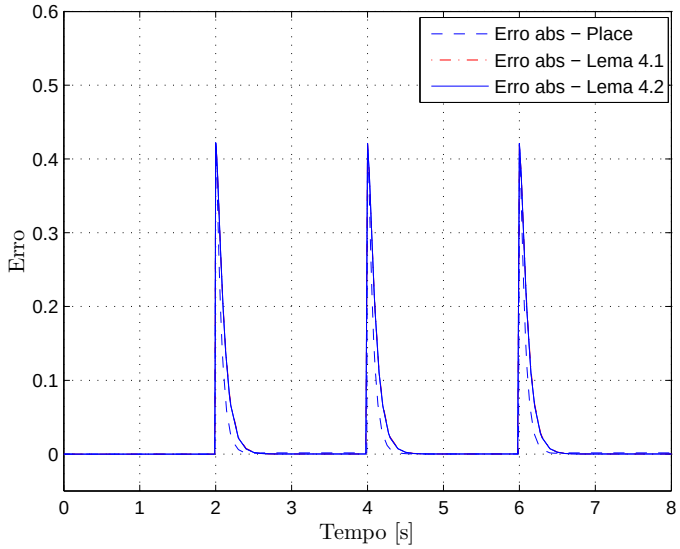


Figura 4.17: Erro absoluto para trajetória reta sem rotação.

Caso 2 - Trajetória Reta com Rotação

A trajetória reta do estudo de caso anterior é repetida nesse caso, porém é acrescentado um deslocamento angular durante o movimento do robô de forma que esse realize movimentos de translação e rotação simultâneos. É definido então que o robô movimente-se com ângulo de rotação $\theta = 0°$ no intervalo de 0 a 2 segundos e com uma variação uniforme do ângulo de rotação de $\theta = 0°$ até $\theta = 135°$ no intervalo de 2 a 8 segundos.

Os resultados das simulações são apresentados nas Figuras 4.18, 4.19, 4.20, 4.21, 4.22 e 4.23 e na tabela 4.3.

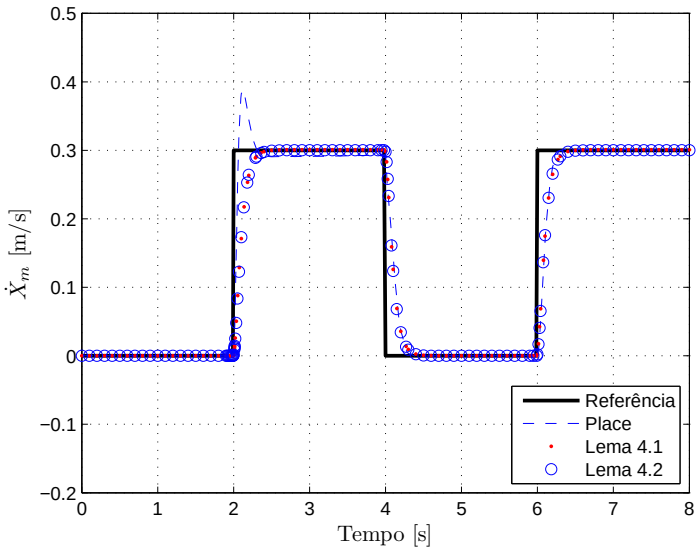


Figura 4.18: Velocidade $\dot{X}_m$ para trajetória reta com rotação.

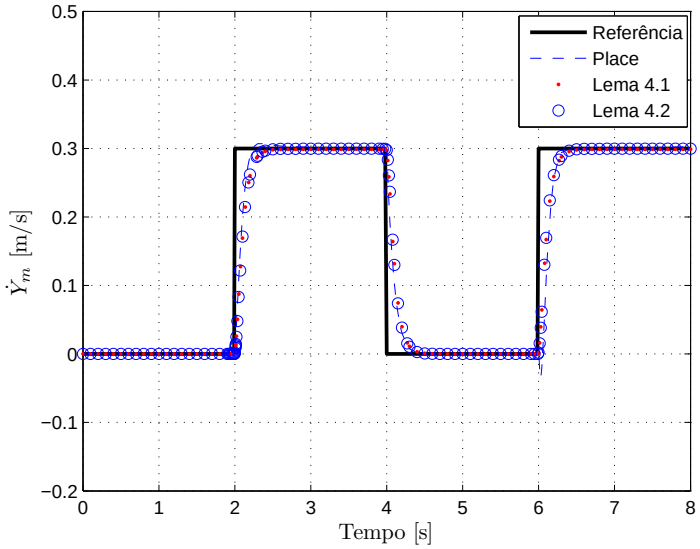


Figura 4.19: Velocidade $\dot{Y}_m$ para trajetória reta com rotação.

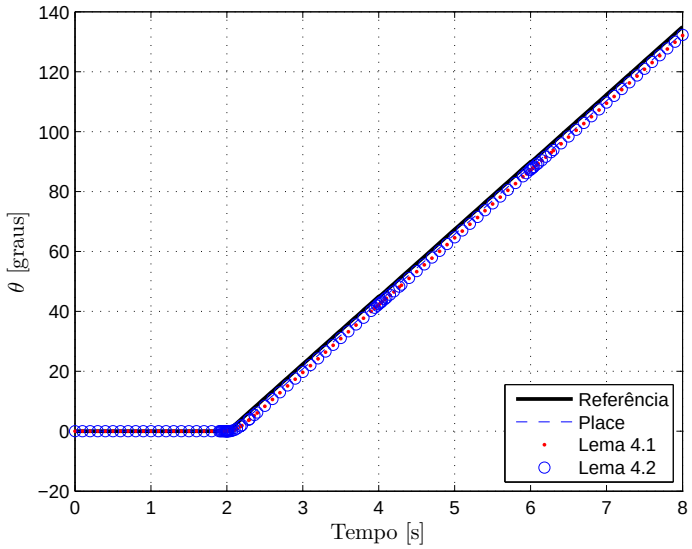


Figura 4.20: Deslocamento angular θ para trajetória reta com rotação.

Tabela 4.3: Erros apresentados pelos controladores para uma trajetória reta com rotação.

	Place	Lema 4.1	Lema 4.2
Erro Máximo	0,4386	0,4220	0,4219
Valor Relativo Percentual	-	3,9531	3,9628

Observe nas Figuras 4.18 e 4.19 que, com a inclusão de um movimento de rotação, o comportamento dos controladores apresenta uma diferença de desempenho. O controlador utilizando `Place` apresenta sobressinal no seguimento das velocidades justamente no instante em que o robô começa a rotacionar. Esse sobressinal é grande o suficiente para gerar saturação, eviden-

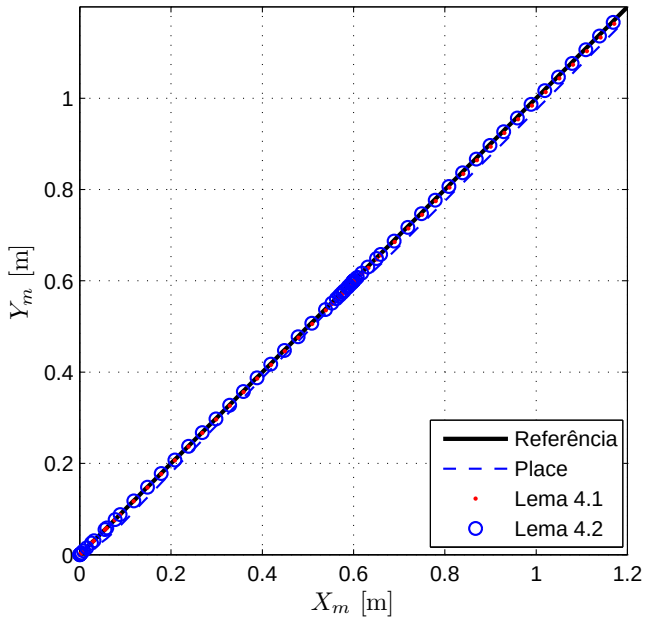


Figura 4.21: Trajetória reta com rotação.

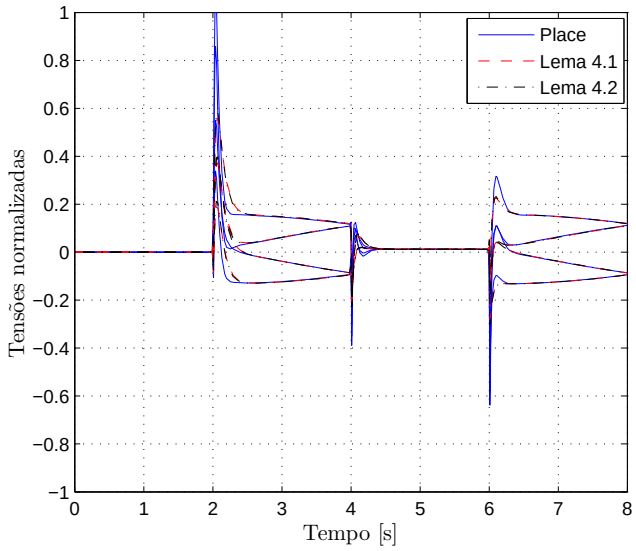


Figura 4.22: Tensões para trajetória reta com rotação.

ciando a necessidade de um maior esforço dos atuadores para realizar a mesma tarefa, como pode ser visto na Figura 4.22. Além disso, o controlador utilizando `Place` apresenta um desvio no seguimento de trajetória mostrado na Figura 4.21. A tabela 4.3 mostra que o desempenho dos Lemas 4.1 e 4.2 é claramente melhor do que o do `Place` .

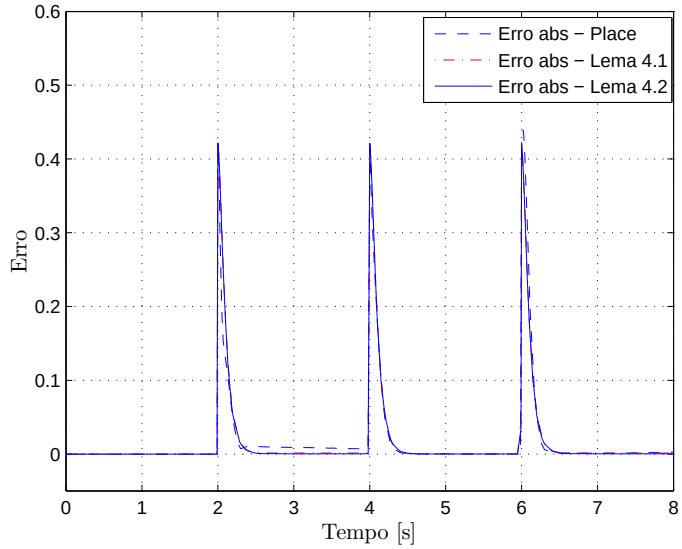


Figura 4.23: Erro absoluto para trajetória reta com rotação.

Caso 3 - Trajetória Quadrada

Neste estudo de caso, uma trajetória quadrada com 1 metro de aresta foi gerada. A velocidade nas direções x e y foi definida como sendo 0,5 m/s. Foi assumido que o robô deve deslocar-se no intervalo de 0 a 2 segundos na direção $\Psi = 0°$, de 2 a 4 segundos na direção $\Psi = 90°$, de 4 a 6 segundos na direção $\Psi = 180°$ e de 6 a 8 segundos na direção $\Psi = 270°$ e com um ângulo de rotação $\theta = 0°$ em todo o trajeto.

Os resultados das simulações são apresentados nas Figuras 4.24, 4.25, 4.26, 4.27, 4.28 e 4.29 e na Tabela 4.4.

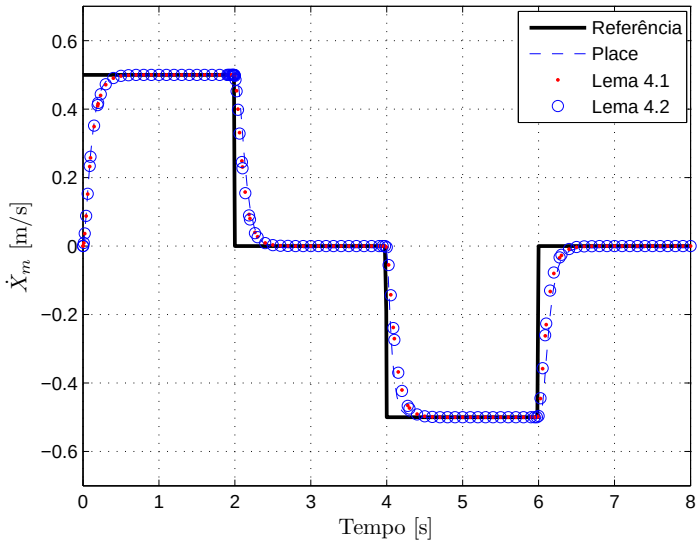


Figura 4.24: Velocidade $\dot{X}_m$ para trajetória quadrada sem rotação.

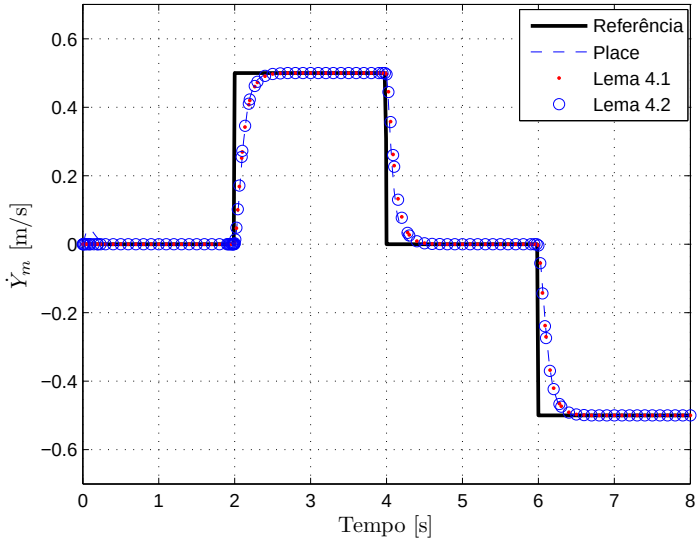


Figura 4.25: Velocidade $\dot{Y}_m$ para trajetória quadrada sem rotação.

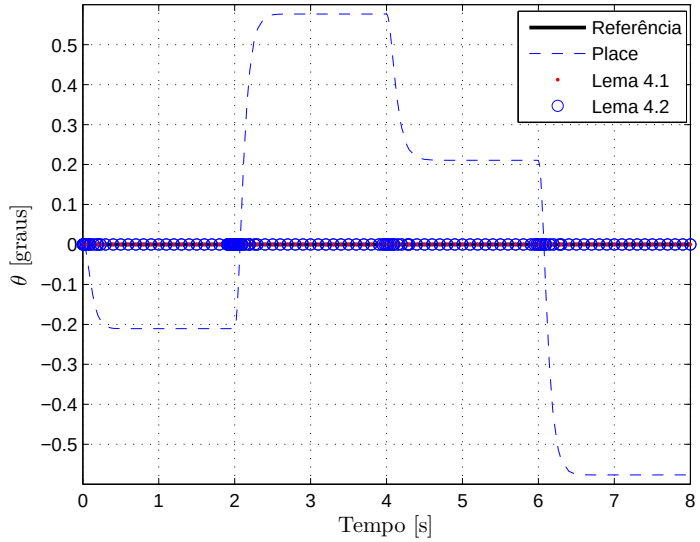


Figura 4.26: Deslocamento angular θ para trajetória quadrada sem rotação.

Tabela 4.4: Erros apresentados pelos controladores para uma trajetória quadrada sem rotação.

	Place	Lema 4.1	Lema 4.2
Erro Máximo	0,7080	0,7033	0,7032
Valor Relativo Percentual	-	0,6657	0,6784

As Figuras 4.24 e 4.25 mostram que, para essa trajetória, o desempenho dos controladores é semelhante, sendo que o `Place` apresenta pequenos desvios em relação aos demais. Já a Figura 4.26 mostra que o controlador utilizando o `Place` gera um erro máximo de aproximadamente 0,6° em relação a referência. Veja na Figura 4.28 que o controlador utilizando `Place` exige um maior

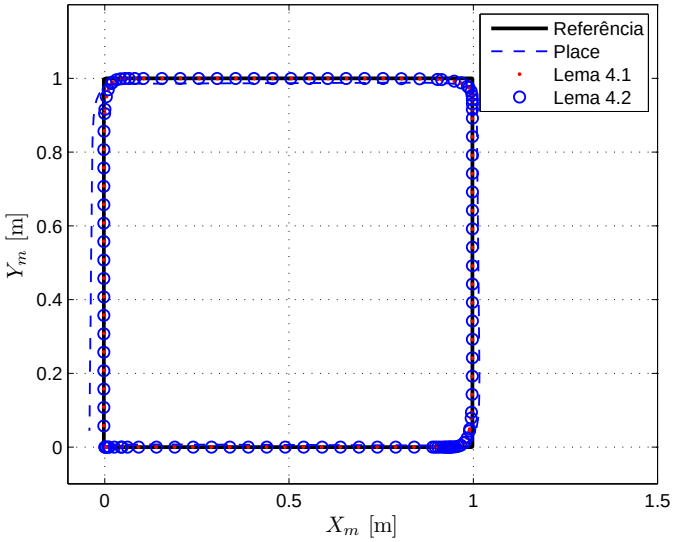


Figura 4.27: Trajetória quadrada sem rotação.

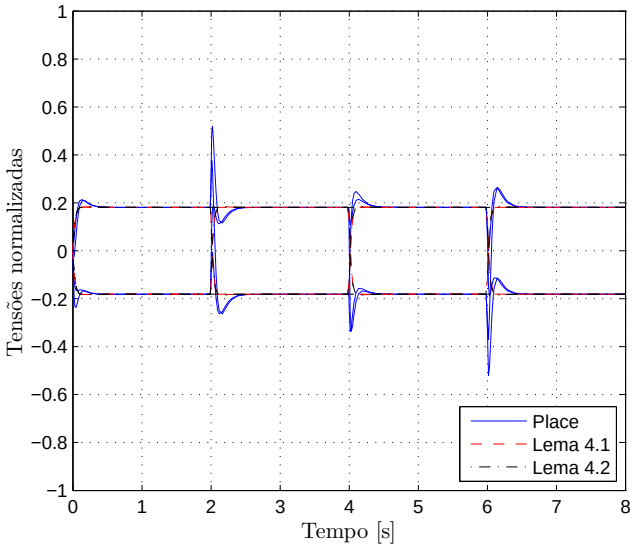


Figura 4.28: Tensões para trajetória quadrada sem rotação.

esforço dos atuadores conforme já havia sido mencionado no caso anterior. Ressalta-se ainda que mesmo para o caso nominal, as intensidades das ações de controle são maiores no `Place` em relação aos demais controladores. Além disso, o controlador utilizando `Place` apresenta um desvio no seguimento de trajetória mostrado na Figura 4.27. Nesse caso, o desempenho dos Lemas 4.1 e 4.2 apresenta uma pequena melhoria em relação ao `Place` , como pode ser visto na Tabela 4.4.

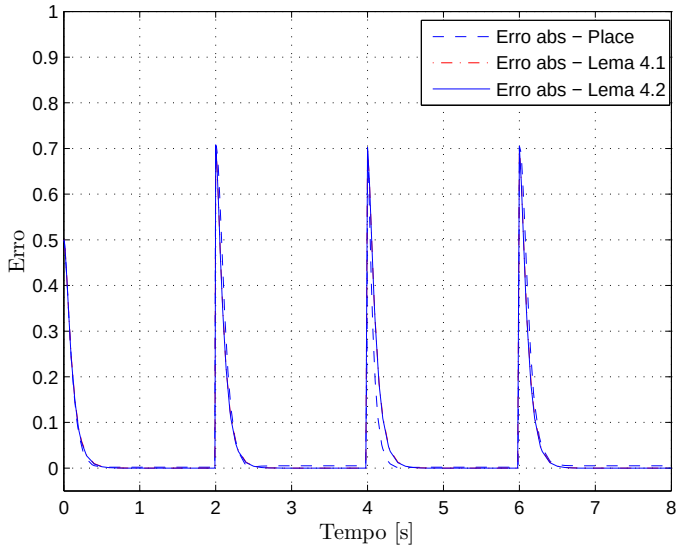


Figura 4.29: Erro absoluto para trajetória quadrada sem rotação.

Caso 4 - Trajetória Quadrada com rotação

A trajetória quadrada do caso anterior é utilizada neste estudo de caso com a inclusão de movimentos de rotação. É solicitado ao robô que mova-se no intervalo de 0 a 2 segundos com ângulo de rotação $\theta = 0°$, com variação uniforme no ângulo de rotação de $\theta = 0°$ a $\theta = 30°$ de 2 a 3 segundos, com $\theta = 30°$ entre 3 e 5 segundos, com variação uniforme no ângulo de rotação de $\theta = 30°$ a $\theta = 0°$ de 5 a 6 segundos e com $\theta = 0°$ de 6 a 8 segundos.

As Figuras 4.30, 4.31, 4.32, 4.33, 4.34 e 4.35 e a tabela 4.5 mostram os resultados das simulações.

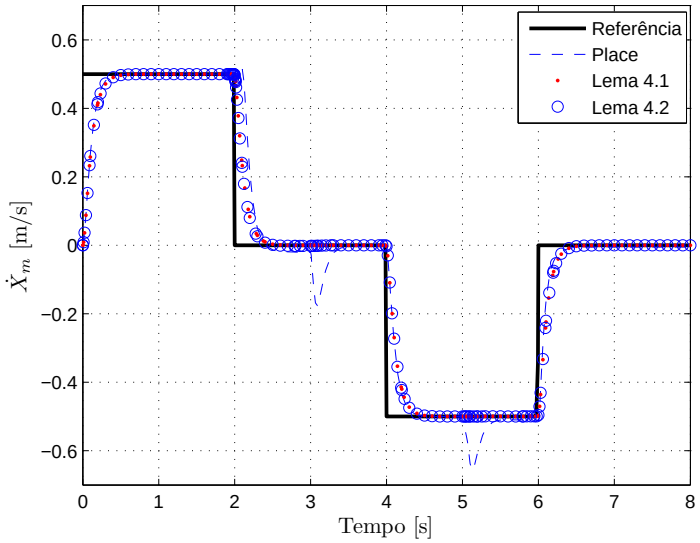


Figura 4.30: Velocidade $\dot{X}_m$ para trajetória quadrada com rotação.

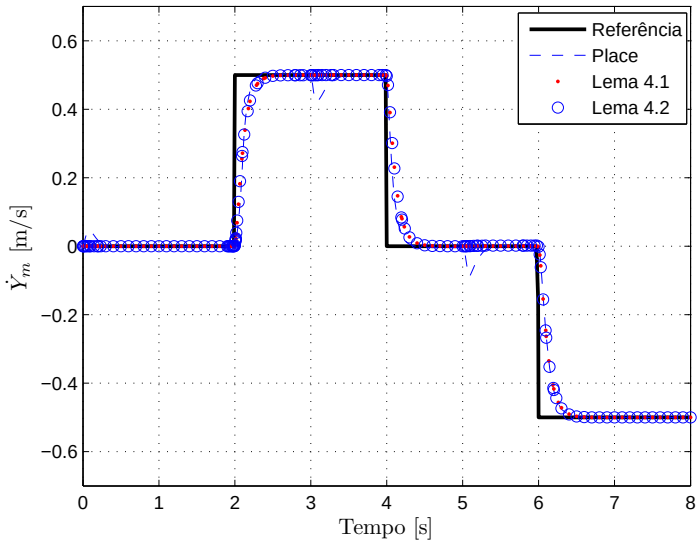


Figura 4.31: Velocidade $\dot{Y}_m$ para trajetória quadrada com rotação.

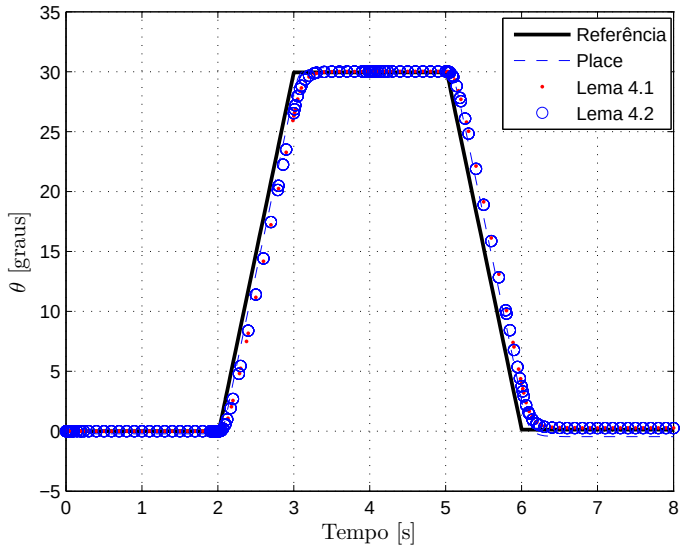


Figura 4.32: Deslocamento angular θ para trajetória quadrada com rotação.

Tabela 4.5: Erros apresentados pelos controladores para uma trajetória quadrada com rotação.

	Place	Lema 4.1	Lema 4.2
Erro Máximo	0,7183	0,7033	0,7032
Valor Relativo Percentual	-	2,1248	2,1377

Observando as Figuras 4.30 e 4.31, nota-se que ocorre um erro de seguimento nos instantes em que é solicitado ao robô que realize um movimento de rotação no caso do controlador utilizando o `Place`. Fica claro nesses gráficos que o desempenho dos Lemas 4.1 e 4.2 é melhor nessa situação. Percebe-se na Figura 4.34 que o controlador utilizando `Place` exige um esforço maior

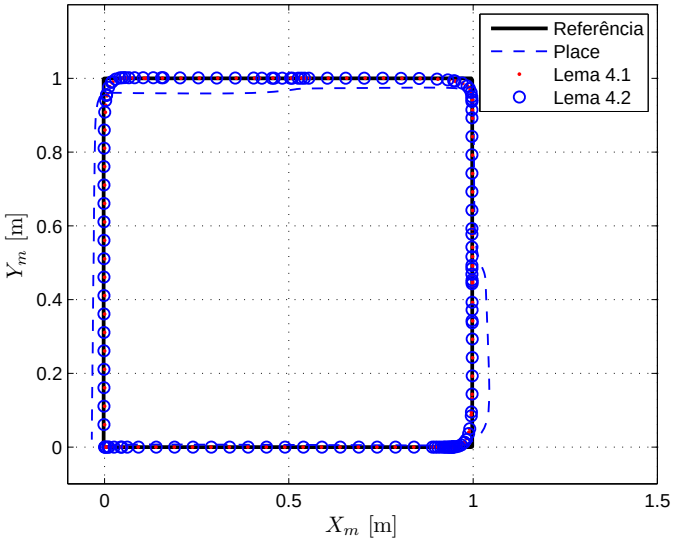


Figura 4.33: Trajetória quadrada com rotação.

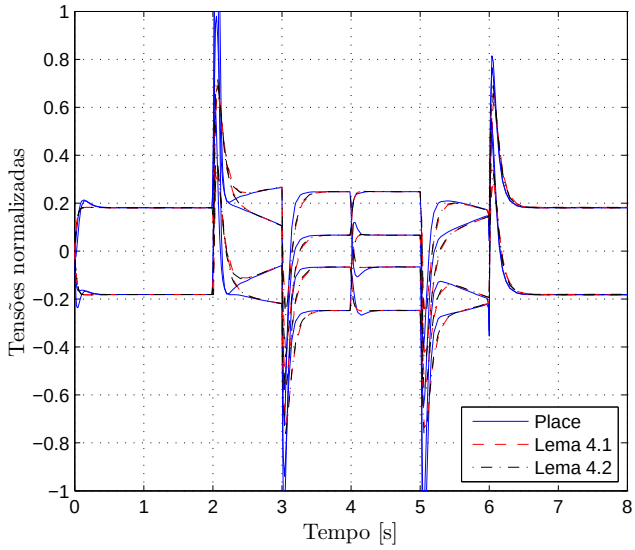


Figura 4.34: Tensões para trajetória quadrada com rotação.

dos atuadores chegando a provocar a saturação dos mesmos. Outro ponto a ser notado é o erro no seguimento de trajetória mostrado na Figura 4.33. A Tabela 4.5, mostra que os desempenhos dos Lemas 4.1 e 4.2 são semelhantes, apresentando uma pequena melhoria em relação ao `Place` .

Caso 5 - Trajetória em Forma de Oito

Nesta seção, uma trajetória em forma de oito é considerada para verificar o desempenho dos controladores no seguimento de referência quando as referências $\dot{X}_m$ e $\dot{Y}_m$ mudam frequentemente. Nessa trajetória, foi solicitado ao robô que movimente-se com ângulo de rotação $\theta = 0°$, mas com uma direção de deslocamento Ψ variando linearmente de $0°$ a $360°$. As componentes

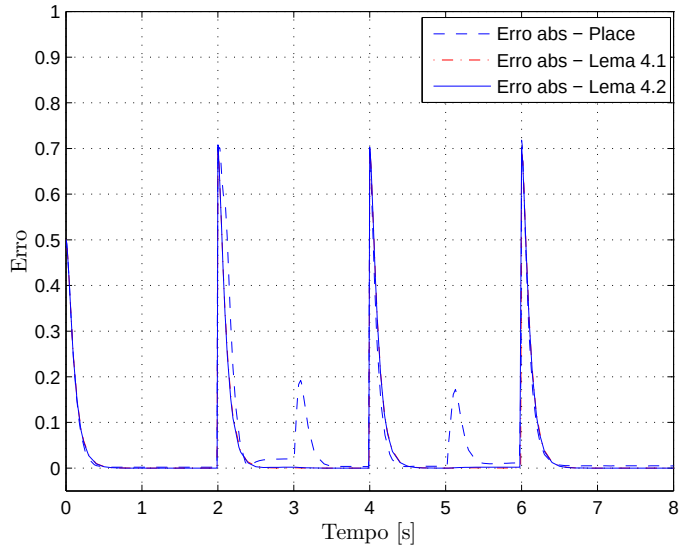


Figura 4.35: Erro absoluto para trajetória quadrada com rotação.

de velocidade nas direções x e y são dadas por (4.63).

$$v_x = 0{,}6\cos(2\Psi) \tag{4.63}$$

$$v_y = 0{,}3\cos(\Psi) \tag{4.64}$$

Os resultados são apresentados nas Figuras 4.36, 4.37, 4.38, 4.39, 4.40 e 4.41 e na Tabela 4.6.

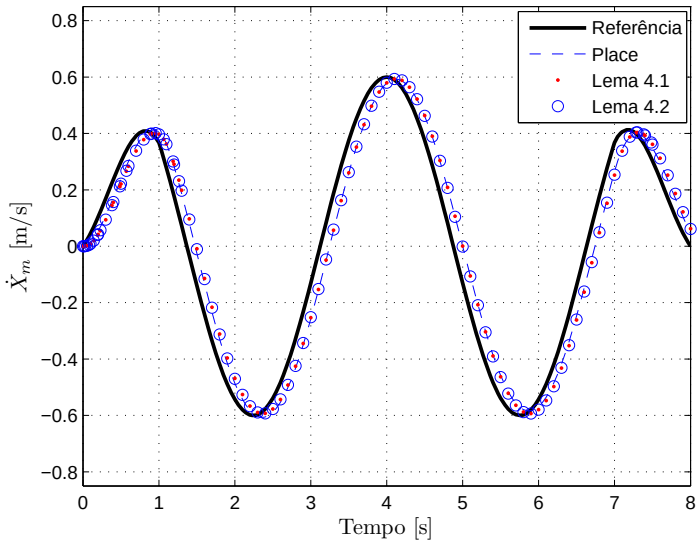


Figura 4.36: Velocidade $\dot{X}_m$ para trajetória em forma de oito sem rotação.

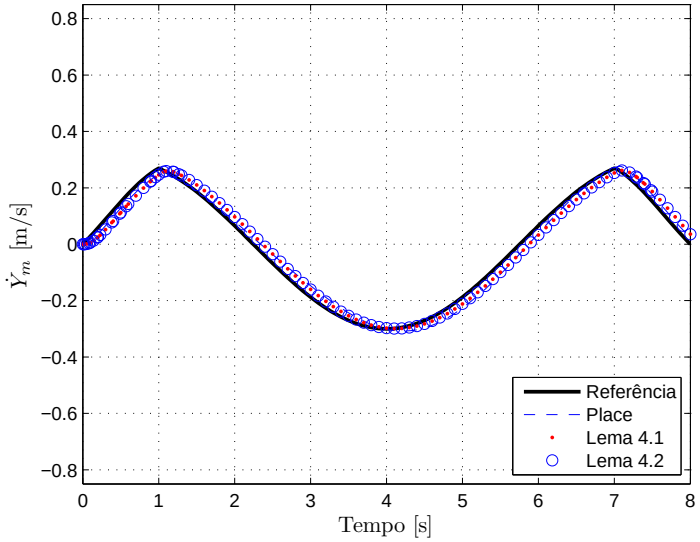


Figura 4.37: Velocidade $\dot{Y}_m$ para trajetória em forma de oito sem rotação.

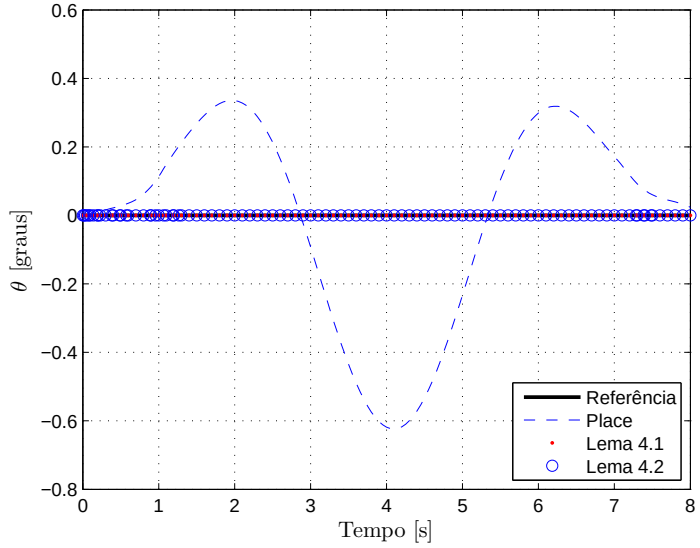


Figura 4.38: Deslocamento angular θ para trajetória em forma de oito sem rotação.

Tabela 4.6: Erros apresentados pelos controladores para uma trajetória em forma de oito sem rotação.

	Place	Lema 4.1	Lema 4.2
Erro Máximo	0,1388	0,1338	0,1319
Valor Relativo Percentual	-	3,7035	5,2151

O seguimento de velocidades mostrado nas Figuras 4.36 e 4.37, demonstra que todos controladores conseguem manter o robô próximo da referência e, conforme pode ser visto na Tabela 4.6, a diferença entre os erros apresentados por esses controladores é pequena. No entanto, quando a comparação é realizada em função do valor relativo percentual, percebe-se que existe

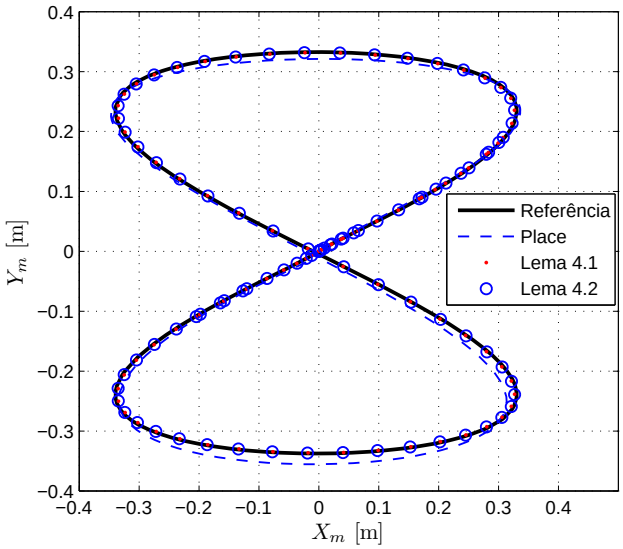


Figura 4.39: Trajetória em forma de oito sem rotação.

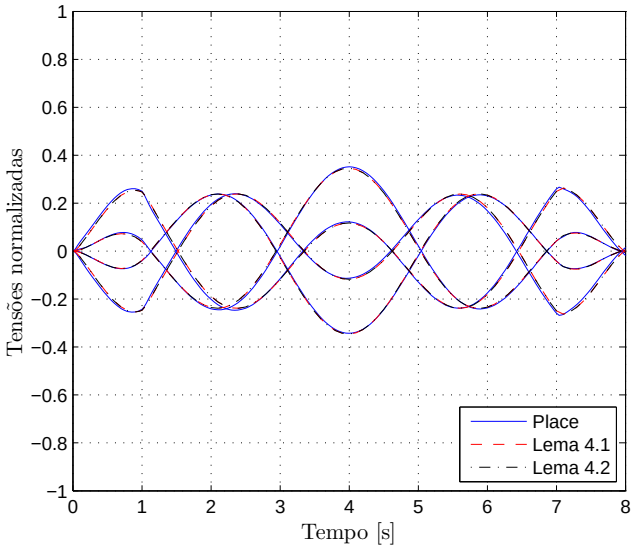


Figura 4.40: Tensões para trajetória em forma de oito sem rotação.

uma diferença considerável entre o resultado apresentado pelo `Place` e pelos Lemas 4.1 e 4.2. Novamente, ocorre uma pequena variação no deslocamento angular quando o `Place` é utilizado, conforme Figura 4.38.

Caso 6 - Trajetória em Forma de Oito com Rotação

A trajetória em forma de oito do caso anterior é novamente enviada ao robô nesse caso, porém são incluídos movimentos de rotação. É solicitado ao robô que mova-se no intervalo de 0 a 2 segundos com ângulo de rotação $\theta = 0°$, com variação uniforme no ângulo de rotação de $\theta = 0°$ a $\theta = 30°$ de 2 a 3 segundos, com $\theta = 30°$ entre 3 e 5 segundos, com variação uniforme

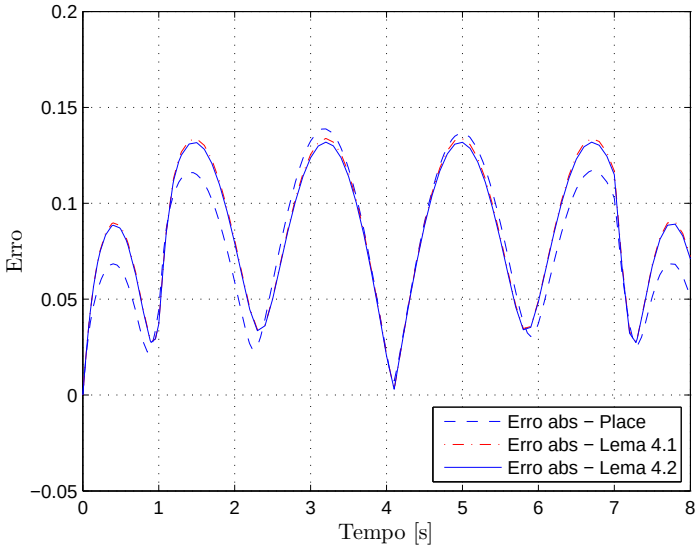


Figura 4.41: Erro absoluto para trajetória em forma de oito sem rotação.

no ângulo de rotação de $\theta = 30°$ a $\theta = 0°$ de 5 a 6 segundos e com $\theta = 0°$ de 6 a 8 segundos.

Os resultados são apresentados nas Figuras 4.42, 4.43, 4.44, 4.45, 4.46 e 4.47 e na tabela 4.7.

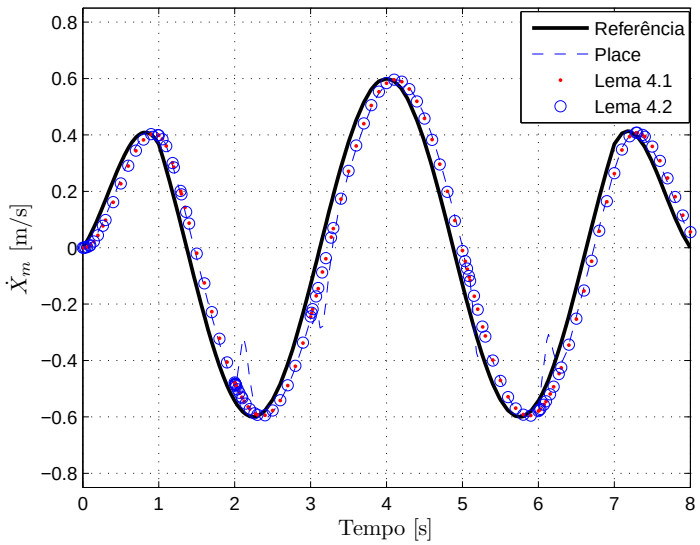


Figura 4.42: Velocidade $\dot{X}_m$ para trajetória em forma de oito com rotação.

Com a inclusão de movimentos de rotação, as respostas do robô ao seguimento de velocidades utilizando o `Place` apresenta erros nos instantes em que o robô inicia os movimentos de rotação, conforme Figuras 4.42 e 4.43. Além disso, a Figura 4.46 mostra picos de tensão nesses instantes chegando a causar saturação dos atuadores. Veja também, na Figura 4.45, que o seguimento de trajetória é prejudicado e o controlador utilizando `Place` não consegue seguir a referência.

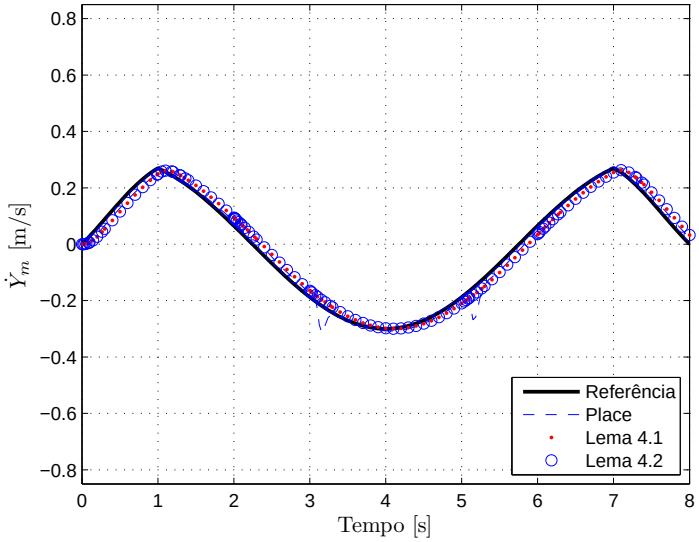


Figura 4.43: Velocidade $\dot{Y}_m$ para trajetória em forma de oito com rotação.

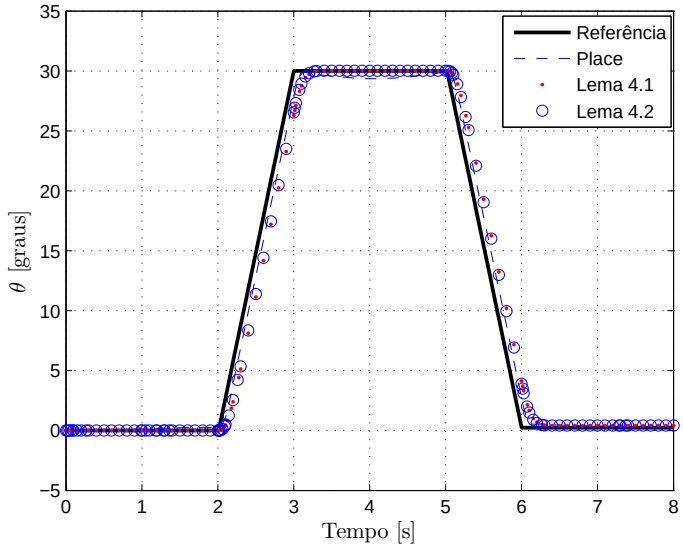


Figura 4.44: Deslocamento angular θ para trajetória em forma de oito com rotação.

Tabela 4.7: Erros apresentados pelos controladores para uma trajetória em forma de oito com rotação.

	Place	Lema 4.1	Lema 4.2
Erro Máximo	0,3324	0,1235	0,1217
Valor Relativo Percentual	-	169,2601	173,1317

Neste estudo de caso, fica claro que, quando são utilizados os Lemas 4.1 e 4.2 para síntese dos controladores, o desempenho desses controladores é realmente superior do que quando utilizado o `Place` , conforme pode ser visto na Figura 4.47 e na Tabela 4.7.

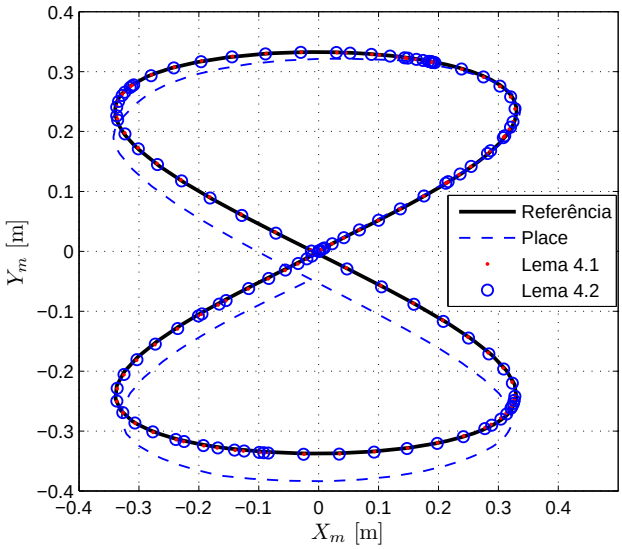


Figura 4.45: Trajetória em forma de oito com rotação.

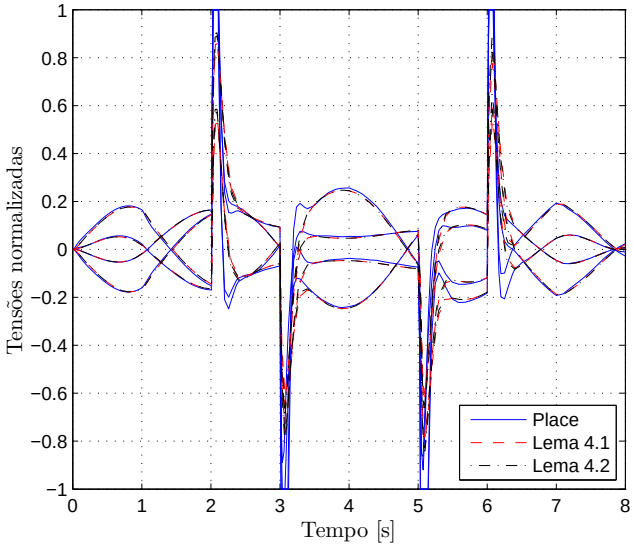


Figura 4.46: Tensões para trajetória em forma de oito com rotação.

Caso 7 - Trajetória Circular

É considerada neste estudo de caso uma trajetória circular com raio de 0,5 metros. A velocidade linear resultante desejada é dada por $v_d = 0{,}5236\ m/s$, no entanto, uma rampa foi gerada durante 1 segundo no início e no fim do intervalo de tempo. É assumido que o robô locomova-se com ângulo de rotação $\theta = 0°$ durante todo o trajeto, mas com uma direção de deslocamento $\Psi = 0°$ nos intervalos de 0 a 1 segundos e de 7 a 8 segundos e com Ψ variando linearmente entre 0° e 360° no intervalo de 1 a 7 segundos. As componentes de velocidade nas direções x e y são dadas pela equação (4.65).

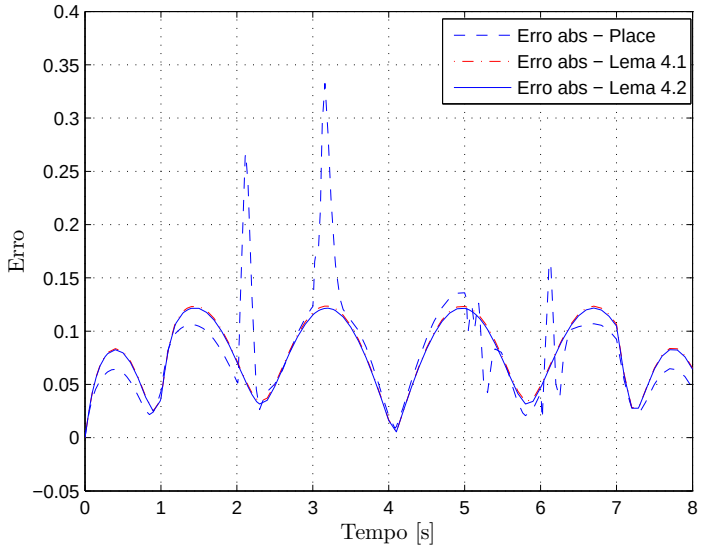


Figura 4.47: Erro absoluto para trajetória em forma de oito com rotação.

$$v_x = v_d \cos(\Psi) \tag{4.65}$$

$$v_y = v_d \sin(\Psi) \tag{4.66}$$

Os resultados das simulações são apresentados nas Figuras 4.48, 4.49, 4.50, 4.51, 4.52 e 4.53 e na tabela 4.8.

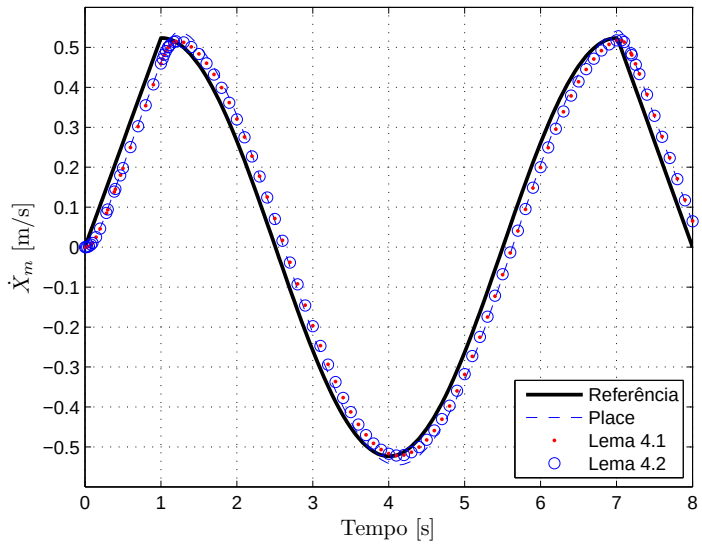


Figura 4.48: Velocidade $\dot{X}_m$ para trajetória circular sem rotação.

Este estudo de caso mostra, conforme Tabela 4.8 que a síntese de controladores utilizando os Lemas 4.1 e 4.2 apresenta resultados consideravelmente melhores do que quando utilizado o

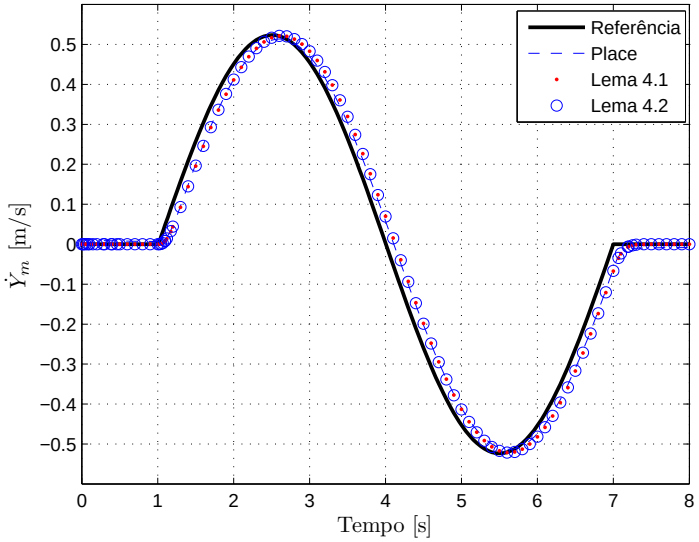


Figura 4.49: Velocidade $\dot{Y}_m$ para trajetória circular sem rotação.

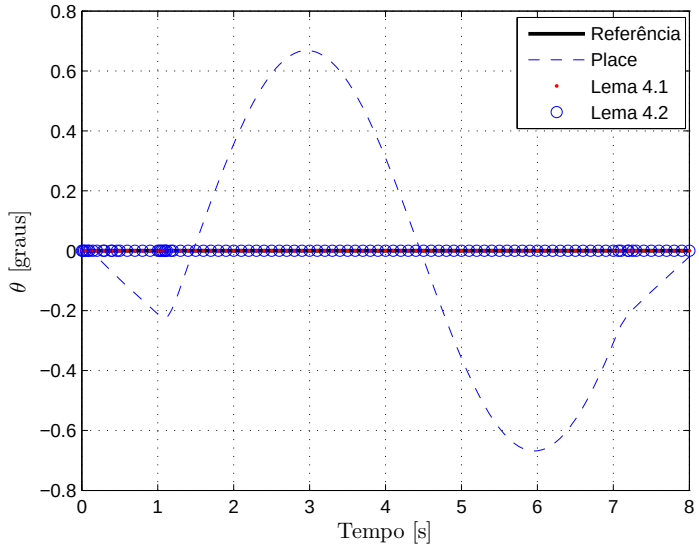


Figura 4.50: Deslocamento angular θ para trajetória circular sem rotação.

Tabela 4.8: Erros apresentados pelos controladores para uma trajetória circular sem rotação.

	Place	Lema 4.1	Lema 4.2
Erro Máximo	0,0813	0,0678	0,0668
Valor Relativo Percentual	-	20,0156	21,7950

`Place` . Veja na Figura 4.51 que o controlador dado pelo `Place` apresenta um erro se seguimento, enquanto que os demais controladores conseguem manter o robô sobre a trajetória de referência.

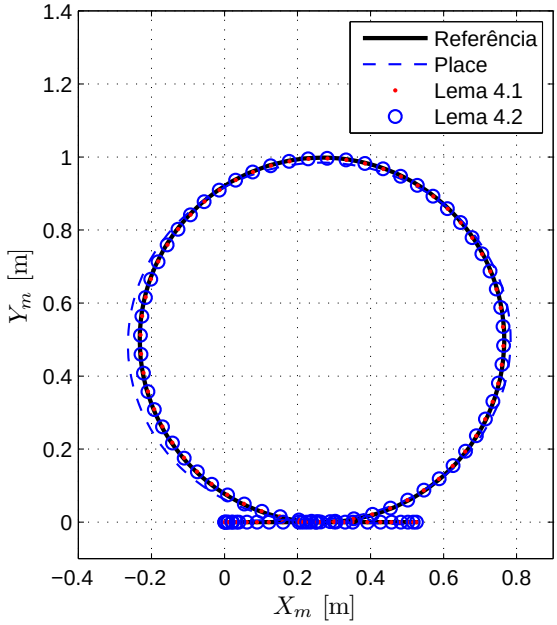


Figura 4.51: Trajetória circular sem rotação.

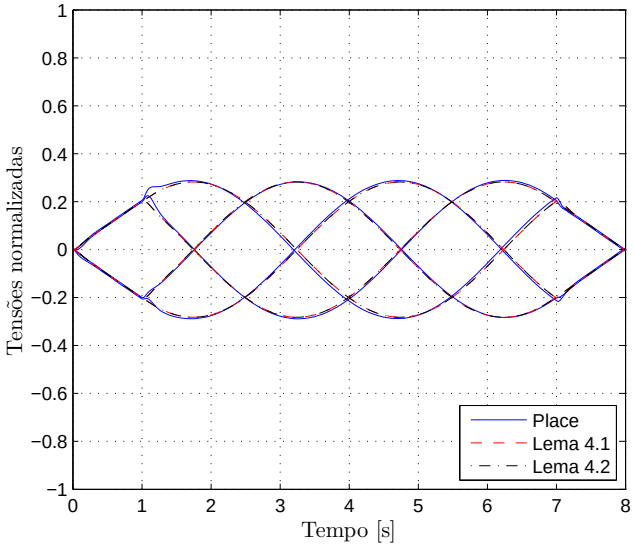


Figura 4.52: Tensões para trajetória circular sem rotação.

Caso 8 - Trajetória Circular com Rotação

A trajetória circular do caso anterior é agora utilizada com a inclusão de movimentos de rotação. Nesse caso, o robô deve locomover-se durante os 3 primeiros segundos com ângulo de rotação $\theta = 0°$, com variação uniforme no ângulo de rotação de $\theta = 0°$ a $\theta = 45°$ de 3 a 6 segundos e com $\theta = 45°$ nos 2 segundos restantes.

Veja nas Figuras 4.54, 4.55, 4.56, 4.57, 4.58 e 4.59 e na tabela 4.9 os resultados das simulações.

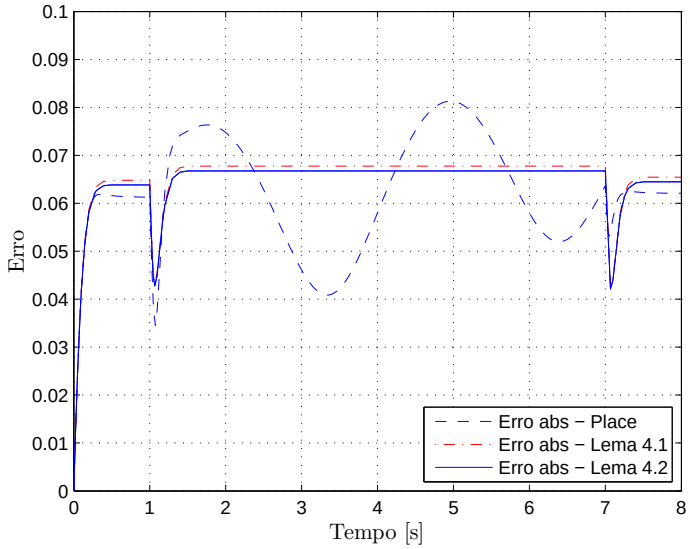


Figura 4.53: Erro absoluto para trajetória circular sem rotação.

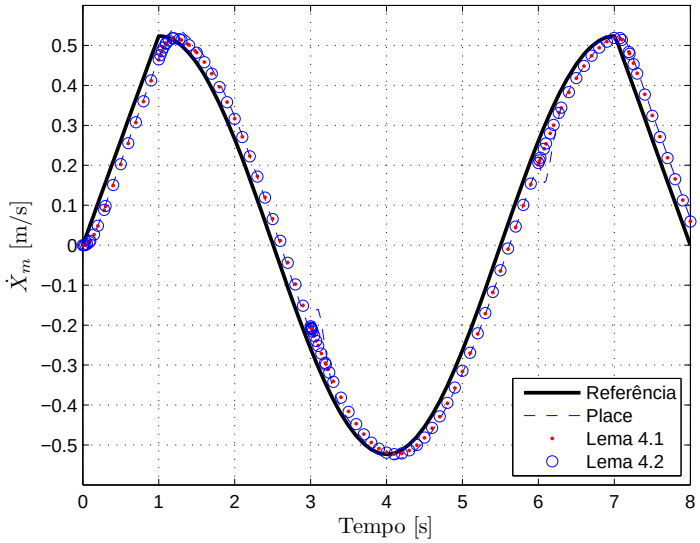


Figura 4.54: Velocidade $\dot{X}_m$ para trajetória circular com rotação.

Tabela 4.9: Erros apresentados pelos controladores para uma trajetória circular com rotação.

	Place	Lema 4.1	Lema 4.2
Erro Máximo	0,1750	0,0635	0,0619
Valor Relativo Percentual	-	175,5911	182,8437

Esse estudo de caso apresenta os resultados mais consideráveis dentre os casos apresentados. Veja na Tabela 4.9 os valores relativos percentuais. Fica evidente a superioridade dos métodos apresentados nos Lemas 4.1 e 4.2 em relação ao `Place` . Observe que os problemas relativos a esse método são bastante evidentes. Ocorre um elevado esforço nos atuadores conforme Figura

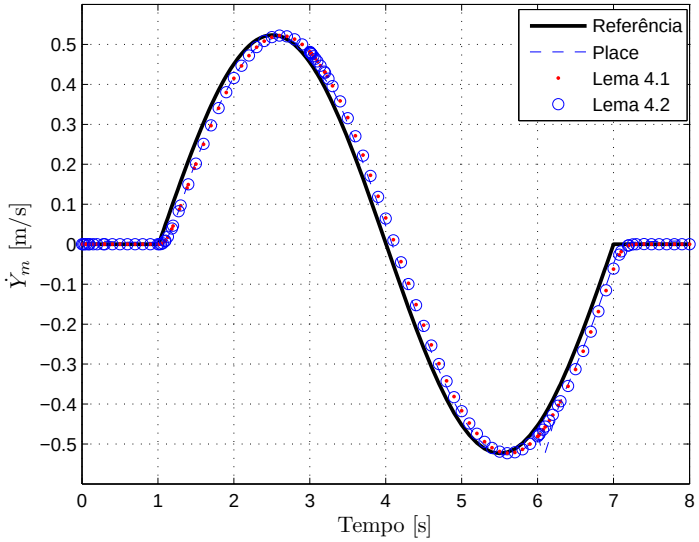


Figura 4.55: Velocidade $\dot{Y}_m$ para trajetória circular com rotação.

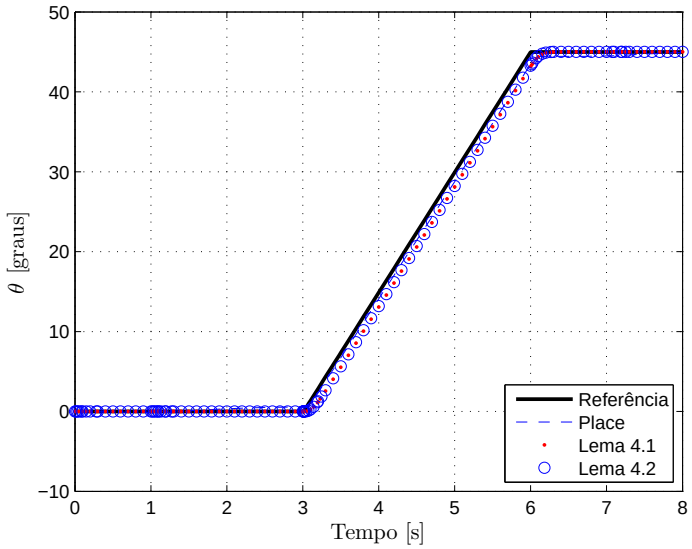


Figura 4.56: Deslocamento angular θ para trajetória circular com rotação.

4.58 e um erro no seguimento de trajetória na Figura 4.57.

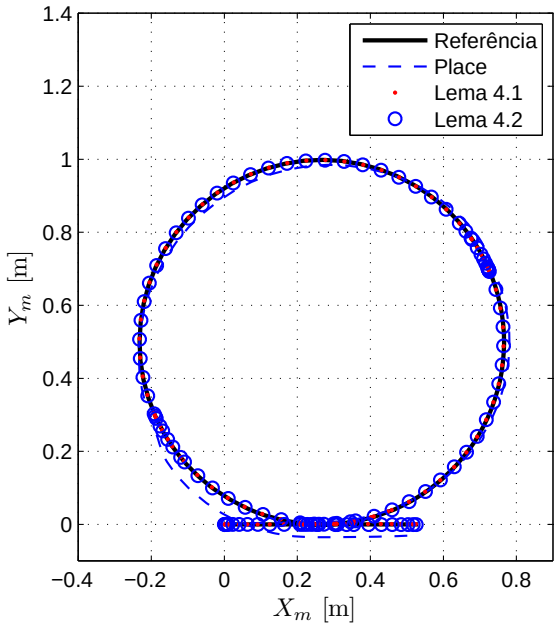


Figura 4.57: Trajetória circular com rotação.

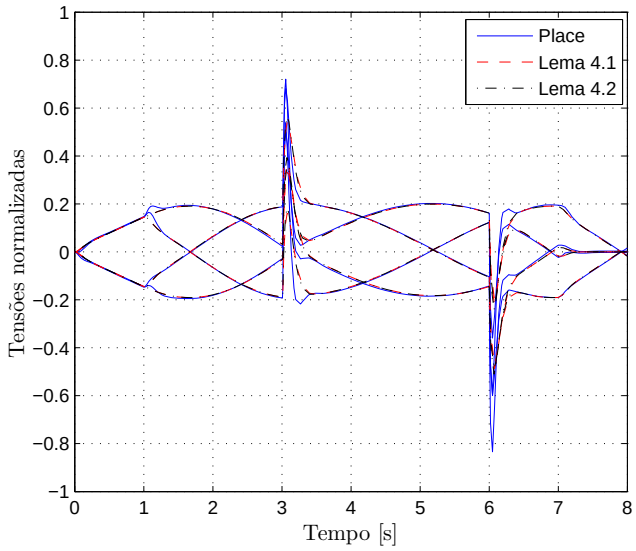


Figura 4.58: Tensões para trajetória circular com rotação.

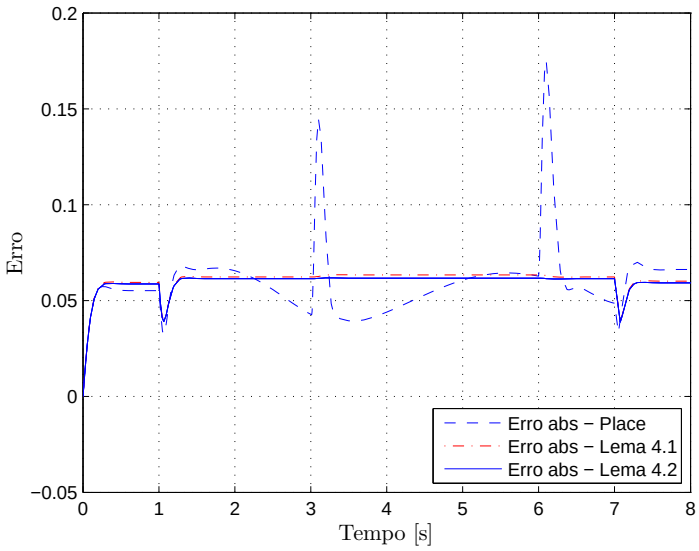


Figura 4.59: Erro absoluto para trajetória circular com rotação.

4.6 Conclusões

Neste capítulo foi realizada a linearização de um sistema não linear incerto nos parâmetros utilizando linearização por realimentação de estados. Ao sistema linearizado pela técnica de linearização por realimentação robusta, foram inseridas incertezas paramétricas representadas na forma de um politopo de incertezas $\mathcal{A}$. A análise de técnicas para síntese de controladores robustos foram apresentadas e comparadas com uma técnica convencional. A análise de desempenho foi realizada com base nos erros máximos cometidos por cada controlador. Percebe-se que os erros variam conforme a trajetória a ser seguida. Nas trajetórias em que as velocidades do robô nas direções x e y variam pouco, como nas trajetórias reta e quadrada sem rotação, os erros cometidos pelo `Place` e pelos controladores robustos são semelhantes. Neste caso, o erro relativo percentual ficou abaixo de 1 %, mas já é possível perceber uma melhoria do controlador dado pelo Lema 4.2 em relação ao Lema 4.1 e ao `Place` . Quando movimentos de rotação são acrescentados às trajetórias reta e quadrada, pode-se ver um aumento nos erros relativos percentuais, principalmente no caso da trajetória reta com rotação, em que os controladores robustos apresentam uma melhoria de aproximadamente 4 % em relação ao `Place` . Já nas trajetórias que exigem mais variações nas velocidades do robô, como nas trajetórias em forma de oito e circular, os controladores robustos apresentam uma melhoria significativa em relação ao `Place` , principalmente quando movimentos de rotação e translação são realizados simultaneamente. Nestes casos, a melhoria dos controladores robustos em relação ao `Place` chega a mais de 180 %. Isso mostra que dependendo das ações a serem realizadas pelo robô, o comportamento do controlador utilizando `Place` altera-se significativamente e, em alguns casos causa a saturação dos atuadores. Nota-se ainda, que o controlador utilizando o `Place` apresenta erros de seguimento para o deslocamento angular, enquanto que os demais não apresentam esse tipo de erro.

Conclui-se então que, quando a dinâmica do sistema é pouco excitada, os controladores robustos apresentam desempenho semelhante ao caso convencional, no entanto, quando trajetórias com grandes variações nas referências e com movimentos de rotação e translação simultâneos são consideradas, os controladores robustos apresentam desempenhos consideravelmente melhores.

Capítulo 5

Conclusões e Perspectivas

A tarefa de navegação autônoma de robôs deve ser cumprida de maneira eficiente, caso contrário, um objetivo pré-definido pode não ser atingido. Diversos fatores influenciam em uma navegação eficiente. Este trabalho tratou questões relacionadas ao modelo matemático, ao projeto do posicionamento das rodas e ao controle de trajetórias de robôs móveis omnidirecionais. Aspectos relacionados a escolha da disposição física das rodas mostraram, através de simulações, uma proposta para otimizar a distribuição de aceleração do robô. Estes aspectos, se levados em consideração durante o projeto de um robô omnidirecional de quatro rodas, pode auxiliar na escolha dos motores a serem utilizados, evitando a aquisição de motores sobre dimensionados e consequentemente reduzir o consumo energético.

Quanto a aplicação de controle, mostrou-se a viabilidade de um controlador não linear robusto que combina a linearização por realimentação de estados robusta a um controlador linear robusto. A associação desses dois métodos permite a obtenção de um controle robusto pois a transformação realizada pela linearização por realimentação robusta respeita o comportamento do sistema não linear, permitindo que as propriedades do controlador linear sejam conservadas. Além disso, a síntese do controlador linear é simplificada pois as variações do sistema linearizado conservam seu significado físico quando a linearização por realimentação robusta é utilizada. Os problemas gerados por incertezas paramétricas no modelo do robô foram tratados utilizando-se controladores robustos na forma de LMIs. Em particular, utilizou-se para a síntese dos controladores lineares robustos dois métodos para alocação de polos em uma região circular do plano complexo, garantindo que mesmo na presença de incertezas paramétricas os polos estejam dentro da região especificada. O desempenho dos controladores robustos foi comparado com o de um controlador clássico utilizando alocação exata de polos. Mostrou-se que mesmo na presença de incertezas os controladores robustos garantem a alocação dos polos de malha fechada na região circular especificada, enquanto que o controlador clássico não garante essa alocação. Os resultados das simulações para seguimento de referência realizadas para o robô F180 com quatro rodas evidenciam as vantagens do controle robusto em relação ao método clássico.

5.1 Contribuições da Dissertação

Nesta dissertação, um modelo matemático para um robô omnidirecional de quatro rodas foi elaborado de forma detalhada. Este modelo foi apresentado de forma genérica, podendo ser utilizado para robôs com parâmetros e distribuição de rodas diferentes dos utilizados neste trabalho. Foi também proposta uma alternativa para a melhoria do desempenho do robô através da escolha adequada da distribuição das rodas, apresentada no Teorema 3.1. O modelo não linear desenvolvido foi então linearizado através da aplicação da técnica de linearização por realimentação de estados robusta. Foram propostos dois controladores robustos na forma de desigualdade matriciais lineares (LMIs) e aplicados sobre o modelo linearizado considerado incerto.

5.2 Trabalhos Futuros

Para trabalhos futuros, são propostas as seguintes direções de pesquisa:

- Realizar testes experimentais de aplicação dos controladores propostos a um robô real. Estes testes têm como objetivo complementar o estudo desenvolvido neste trabalho, que já compreendeu a parte teórica e simulações.

- Aplicar técnicas de identificação de sistemas ao robô real para definir a faixa de variação de seus parâmetros.

- Estudar a associação do método de linearização por realimentação robusta a outros tipos de controladores lineares, especialmente os que geram ganhos dependentes de parâmetros.

- Estudar a aplicação de controladores não lineares diretamente ao modelo não linear do robô. O objetivo seria avaliar o desempenho desses controladores em relação aos já desenvolvidos.

- Aplicar outros índices para avaliar o desempenho dos controladores propostos.

Apêndice A

Conceitos de Geometria Diferencial

Neste apêndice são apresentados alguns conceitos de geometria diferencial, os quais foram obtidos em (Slotine e Li, 1991), que são empregados no desenvolvimento dessa dissertação.

Dadas uma função escalar $h(x)$ e campo vetorial $f(x)$, define-se:

Definição A.1 (Gradiente) *Seja $h : \mathbb{R}^n \to \mathbb{R}$ uma função escalar suave de $x \in \mathbb{R}^n$. O gradiente de $h(x)$, denotado como ∇h, é definido como:*

$$\nabla h = \frac{\partial h}{\partial x}, \tag{A.1}$$

sendo, então, o gradiente de $h(x)$ um vetor linha, cujos elementos são $\nabla h_j = \dfrac{\partial h}{\partial x_j}$.

Definição A.2 (Jacobiano) *Seja $f : \mathbb{R}^n \to \mathbb{R}^p$ um campo vetorial suave de $x \in \mathbb{R}^n$. O jacobiano de $f(x)$, denotado ∇f, é definido como:*

$$\nabla f = \frac{\partial f}{\partial x}, \tag{A.2}$$

sendo, então, o jacobiano de $f(x)$ uma matriz $p \times n$, formada pelos elementos $\nabla f_{i,j} = \dfrac{\partial h_i}{\partial x_j}$.

Definição A.3 (Derivada de Lie) *Sejam $h : \mathbb{R}^n \to \mathbb{R}$ uma função escalar suave de $x \in \mathbb{R}^n$ e $f : \mathbb{R}^n \to \mathbb{R}^n$ um campo vetorial suave em $\mathbb{R}^n$. A derivada de Lie de $h(x)$ com respeito a $f(x)$, denotada $\mathcal{L}_f h$, é uma função escalar definida como:*

$$\mathcal{L}_f h = \nabla h f. \tag{A.3}$$

Então, a derivada de Lie $\mathcal{L}_f h$ é simplesmente a derivada direcional de h ao longo da direção do vetor f.

A derivada de Lie pode ser aplicada repetidamente, e definida recursivamente como:

$$\begin{aligned} \mathcal{L}_f^0 h &= h, \\ \mathcal{L}_f^i h &= \mathcal{L}_f(\mathcal{L}_f^{i-1} h) = \nabla(\mathcal{L}_f^{i-1} h)\, f \qquad \textit{para} \qquad i \geq 1. \end{aligned} \tag{A.4}$$

Se $g : \mathbb{R}^n \to \mathbb{R}^n$ *é um outro campo vetorial em* $\mathbb{R}^n$*, então a função escalar* $\mathcal{L}_g\mathcal{L}_f h$ *é dada por:*

$$\mathcal{L}_g\mathcal{L}_f h = \nabla(\mathcal{L}_f h)g. \tag{A.5}$$

Definição A.4 (Colchete de Lie) *Sejam* $f : \mathbb{R}^n \to \mathbb{R}^n$ *e* $g : \mathbb{R}^n \to \mathbb{R}^n$ *dois campos vetoriais suaves em* $\mathbb{R}^n$*. O colchete de Lie de* $f(x)$ *e* $g(x)$*, denotado* $[f,g]$*, é um terceiro campo vetorial definido como:*

$$[f,g] = \nabla g\ f - \nabla f\ g. \tag{A.6}$$

O colchete de Lie $[f,g]$ *é comumente representado por* $ad_f g$*, e pode ser aplicado repetidamente, e logo ser escrito recursivamente como:*

$$\begin{aligned} ad_f^0 g &= g, \\ ad_f^i g &= [f,ad_f^{i-1}] = ad_f(ad_f^{i-1}g) \qquad i \geq 1. \end{aligned} \tag{A.7}$$

O colchete de Lie apresenta as seguintes propriedades:

1. *Bilinearidade:*

$$\begin{aligned} [\alpha_1 f_1 + \alpha_2 f_2,g] &= \alpha_1[f_1,g] + \alpha_2[f_2,g], \\ [f,\alpha_1 g_1 + \alpha_2 g_2] &= \alpha_1[f,g_1] + \alpha_2[f,g_2]. \end{aligned} \tag{A.8}$$

2. *Anti-simetria:*

$$[f,g] = -[g,f]. \tag{A.9}$$

3. *Identidade de Jacobi:*

$$\mathcal{L}_{ad_f g}h = \mathcal{L}_f\mathcal{L}_g h - \mathcal{L}g\mathcal{L}_f h. \tag{A.10}$$

Sendo $f_1 : \mathbb{R}^n \to \mathbb{R}^n$*,* $f_2 : \mathbb{R}^n \to \mathbb{R}^n$*,* $g_1 : \mathbb{R}^n \to \mathbb{R}^n$*,* $g_2 : \mathbb{R}^n \to \mathbb{R}^n$ *campos vetoriais suaves e,* $\alpha_1 \in \mathbb{R}$*,* $\alpha_2 \in \mathbb{R}$*, constantes.*

Apêndice B

Ferramentas

B.1 Desigualdades Matriciais Lineares - LMIs

Com o desenvolvimento de métodos de pontos interiores para problemas de programação semi definida, SDP (do inglês semidefinite programming problem), desigualdades matriciais lineares, LMIs (do inglês linear matrix inequalities), têm sido uma ferramenta útil para resolução de problemas de controle. A ideia básica do método de LMIs é expressar o problema dado como um SDP. A formulação de LMIs é relevante por várias razões. Uma dessas razões é que escrevendo um dado problema nessa forma, as soluções numéricas podem ser encontradas de forma eficiente (Boyd et al., 1994; Ghaoui e lulian Niculescu, 2000).

Uma desigualdade matricial linear tem a seguinte forma:

$$F(p) = F_0 + \sum_{i}^{m} p_i F_i > \mathbf{0}, \tag{B.1}$$

em que $p_i \in \mathbb{R}^m$, para $i = 1, \ldots, m$, são variáveis escalares a serem determinadas e $F_i \in \mathbb{R}^{n \times n}$, para $i = 0, 1, \ldots, m$ são matrizes simétricas precisamente conhecidas. A desigualdade significa que $F(p)$ é uma matriz definida positiva, ou seja,

$$z^T F(p) z > 0, \quad \forall \quad z \neq \mathbf{0}, z \in \mathbb{R}^n. \tag{B.2}$$

Isso significa que $F(p)$ é uma função afim dos elementos de p, que representa um vetor $p = [p_1, \ldots, p_m]$.

A equação (B.1) é uma LMI estrita. No caso em que $F(p)$ é semi definida positiva, essa seria uma LMI não estrita. A LMI estrita é factível quando existe um vetor p que torna a desigualdade verdadeira, ou seja, $F(p)$ uma matriz definida positiva. Uma LMI não estrita factível pode ser reduzida para o caso de uma LMI estrita factível equivalente. Isso pode ser feito acrescentando um valor constante a matriz $F(p)$, tornando essa matriz definida positiva, ou seja, $F(p) + \epsilon \mathbf{I} > \mathbf{0}$.

Uma LMI pode ser reescrita em termos de um conjunto de desigualdades escalares. De forma mais específica, considere a LMI (B.1), ela é equivalente a n desigualdades polinomiais. Para exemplificar, considere as desigualdades apresentadas em (B.3).

$$\begin{aligned} P &< \mathbf{0}; \\ A^T PA - P &> \mathbf{0}. \end{aligned} \tag{B.3}$$

Essas desigualdades são LMIs, sendo

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \tag{B.4}$$

As desigualdades podem ser escritas em termos das incógnitas do problema, p_1, p_2 e p_3, assim

$$\begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} > \mathbf{0} \Leftrightarrow \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} p_1 + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} p_2 + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} p_3 > \mathbf{0}. \tag{B.5}$$

$$A^T PA - P = \begin{bmatrix} a^2p_1 + 2acp_2 + c^2p_3 & abp_1 + (ad+bc)p_2 + cdp_3 \\ abp_1 + (ad+bc)p_2 + cdp_3 & b^2p_1 + 2bdp_2 + d^2p_3 \end{bmatrix} \tag{B.6}$$

$$= \begin{bmatrix} a^2 & ab \\ ab & b^2 \end{bmatrix} p_1 + \begin{bmatrix} 2ac & bc+ad \\ ad+bc & 2bd \end{bmatrix} p_2 + \begin{bmatrix} c^2 & cd \\ cd & 2b^2 \end{bmatrix} p_3 < \mathbf{0}. \tag{B.7}$$

Observe que no exemplo dado, a matriz F_0 é nula. De acordo com (Wylie e Barret, 1995; Vanantwerp e Braatz, 2000), as restrições podem ser colocadas em termos dos menores principais líderes de P e de $A^T PA - P$, resultando em

$$\begin{aligned} p_1 &> 0, \\ p_3 &> 0, \\ p_1p_3 - p_2^2 &> 0. \end{aligned} \tag{B.8}$$

$$\begin{aligned} a^2p_1 + 2acp_2 + c^2p_3 &< 0, \\ b^2p_1 + 2bdp_2 + d^2p_3 &< 0, \\ (a^2p_1 + 2acp_2 + c^2p_3)(b^2p_1 + 2bdp_2 + d^2p_3) - (abp_1 + (ad+bc)p_2 + cdp_3)^2 &< 0. \end{aligned} \tag{B.9}$$

É importante salientar que, ao se representar uma LMI por um conjunto de n desigualdades polinomiais, há a possibilidade de alguns desses polinômios serem não lineares, como acontece no exemplo dado acima. Uma importante propriedade das LMIs é a convexidade, ou seja, o conjunto de soluções x que atende a desigualdade é convexo. Em um problema de otimização convexa, o mínimo local encontrado é o mínimo global. Isso torna a solução do problema simples do ponto de vista de otimização. Um conjunto $\mathcal{C}$ é convexo se $\lambda x + (1-\lambda)y \in \mathcal{C}$ para todo $x, y \in \mathcal{C}$ e $\lambda \in [0,1]$.

B.2 Complemento de Schur

Considere a matriz quadrada simétrica X

$$X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}. \tag{B.10}$$

O complemento de Schur pode ser usado na caracterização da positividade de X, com as seguintes propriedades:

- $X > \mathbf{0}$ se e somente se $A > \mathbf{0}$ e $C - B^T A^{-1} B > \mathbf{0}$;
- $X > \mathbf{0}$ se e somente se $C > \mathbf{0}$ e $A - BC^{-1}B^T > \mathbf{0}$;
- se $A > \mathbf{0}$, $X \geq \mathbf{0}$ se e somente se $C - B^T A^{-1} B \geq \mathbf{0}$;
- se $C > \mathbf{0}$, $X \geq \mathbf{0}$ se e somente se $A - BC^{-1}B^T \geq \mathbf{0}$.

A matriz $C - B^T A^{-1} B$ é chamada de complemento de Schur de X em relação a A se $det(A) \neq 0$. Se $det(C) \neq 0$, $A - BC^{-1}B^T$ é o complemento de Schur de X em relação a C. Manipulações envolvendo o complemento de Schur permitem transformar desigualdades convexas não lineares, que regularmente aparecem em problemas de controle, em LMIs (Vanantwerp e Braatz, 2000; Oliveira e Peres, 2010).

Prova: A partir da transformação de congruência, caso exista A^{-1}, tem-se

$$X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ B^T A^{-1} & \mathbf{I} \end{bmatrix}}_{\mathcal{T}^T} \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & C - B^T A^{-1} B \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{I} & A^{-1}B \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathcal{T}}. \tag{B.11}$$

Como $\mathcal{T}$ é uma matriz não singular

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} > \mathbf{0} \Leftrightarrow \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & C - B^T A^{-1} B \end{bmatrix} > \mathbf{0}. \tag{B.12}$$

Analogamente, se existe C^{-1}, então existe,

$$X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} = \begin{bmatrix} \mathbf{I} & BC^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} A - BC^{-1}B^T & \mathbf{0} \\ \mathbf{0} & C \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ C^{-1}B^T & \mathbf{I} \end{bmatrix}, \tag{B.13}$$

e, portanto,

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} > \mathbf{0} \Leftrightarrow \begin{bmatrix} A - BC^{-1}B^T & \mathbf{0} \\ \mathbf{0} & C \end{bmatrix} > \mathbf{0}. \tag{B.14}$$

■

Bibliografia

Almeida, A. D. V. (2005). *B-Spline: CAEP - Algoritmos culturais para geração de trajetórias B-Spline de robôs móveis*. Dissertação de Mestrado, Pontifícia Universidade Católica do Paraná, Curitiba, Outubro de 2005.

Andrade, J. F. A., Mendeleck, A., e Zampieri, D. E. (2001). Geração de trajetórias para robôs móveis autônomos usando redes neurais artificiais. In *Anais do V Congresso Brasileiro de Redes Neurais. Rio de Janeiro, RJ, abril de 2001.*, volume 1, páginas 571–576.

Angeles, J. (2003). *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*. New York: Springer-Verlag New York, Inc. 2^a ed.

Arantes, J. M. O., Abdalla Júnior, M. A., e Nepomuceno, E. G. (2009). Desenvolvimento de um sistema de navegação baseado em autômatos celulares e no algoritmo de propagação wavefront. In *IX Simpósio Brasileiro de Automação Inteligente, 2009, Brasília. Anais do IX Simpósio Brasileiro de Automação Inteligente, 2009*, volume 1, páginas 1–6.

Arras, K. O. e Vestli, S. J. (1998). Hybrid, high-precision localisation for the mail distributing mobile robot system MOPS. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation. Leuven, Belgium*, volume 4, páginas 3129–3134.

Asimov, I. (1950). *I, Robot*. Gnome Press.

Asoh, H., Hayamizu, S., Hara, I., Motomura, Y., Akaho, S., e Matsui., T. (1997). Socially embedded learning of office-conversant robot JIJO-2. In *Proceedings of International Joint Conferences on Artificial Intelligence. IJCAI, Inc.*

Balicki, M., Uneri, A., Iordachita, I., Handa, J., Gehlbach, P., e Taylor, R. (2010). Micro-force sensing in robot assisted membrane peeling for vitreoretinal surgery. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6363 LNCS(PART 3):303–310.

Barmish, B. R. (1985). Necessary and sufficient conditions for quadratic stabilizability of an uncertain system. *Journal of Optimization Theory and Applications*, 46:399–408.

Barnes, N. e Zelinsky, A. (2008). A national perspective on the needs, themes, and major groups - robotics research in australia. *IEEE Robotics and Automation Magazine*, 15(2):89–95.

Bloch, A., Reyhanoglu, M., e McClamroch, N. (1992). Control and stabilization of nonholonomic dynamic systems. *IEEE Transactions on Automatic Control*, 37(11):1746–1757.

Bogue, R. (2010). Microrobots and nanorobots: A review of recent developments. *Industrial Robot*, 37(4):341–346.

Borenstein, J., Everett, H. R., Feng, L., e Wehe, D. (1997). Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems*, 14:231–249.

Bourles, H. (2000). Addendum to "w-stability and local input-output stability results". *IEEE Transactions on Automatic Control*, 45(6):1220–1221.

Bourles, H. e Colledani, F. (1995). W-stability and local input-output stability results. *IEEE Transactions on Automatic Control*, 40(6):1102–1108.

Boyd, S., El Ghaoui, L., Feron, E., e Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA.

Braunl, T. (2003). *Embedded Robotics*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Burgard, W., Cremers, A., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., e Thrun, S. (1999). Experiences with an interactive museum tourguide robot. *Artificial Intelligence*, 114(1-2):3–55.

Canudas de Wit, C. (1998). Trends in mobile robot and vehicle control. In Siciliano, B. e Valavanis, K., editors, *Control Problems in Robotics and Automation*, volume 230 of *Lecture Notes in Control and Information Sciences*, páginas 151–175. Springer Berlin/Heidelberg.

Cao, A., Wei, J., Zhao, B., e Li, J. (2007a). Recognizing robot-soccer with half-chord method. In *Proceedings of the IEEE International Conference on Control and Automation. ICCA 2007.*, páginas 2097–2100.

Cao, Q., Huang, Y., e Leng, C. (2007b). An amendatory dynamic model with slip for four-wheeled omnidirectional mobile robot. In *Proceedings of the SPIE - The 2007 International Conference on Mechatronics and Information Technology (ICMIT2007) on Mechatronics, MEMS, and Smart Materials*, volume 6794, página 67945G.

Carter, B., Good, M., Dorohoff, M., Lew, J., Ii, R. L. W., e Gallina, P. (2001). Mechanical design and modeling of an omni-directional robocup player. In *Proceedings RoboCup 2001 International Symposium, Seattle, WA*, página 10.

Celiberto Jr., L. A., Ribeiro, C. H. C., Costa, A., e Bianchi, R. A. C. (2008). Heuristic reinforcement learning applied to robocup simulation agents. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5001 LNAI:220–227.

Chen, C.-T. (1999). *Linear System Theory and Design*. Oxford University Press, Nova York, NY, USA.

Chilali, M., Gahinet, P., e Apkarian, P. (1999). Robust pole placement in LMI regions. *IEEE Transactions on Automatic Control*, 44(12):2257–2270.

Choi, J.-S. e Kim, B. K. (2001). Near minimum-time direct voltage control algorithms for wheeled mobile robots with current and voltage constraints. *Robotica*, 19:29–39.

Dickerson, S. e Lapin, B. (1991). Control of an omni-directional robotic vehicle with mecanum wheels. In *Proceedings of the National Telesystems Conference Vol. 1, NTC '91.*, páginas 323–328.

Douret, J., Benosman, R., Bouzar, S., e Devars, J. (2003). Localization of robots in F180 league using projective geometry. In Kaminka, G., Lima, P., e Rojas, R., editors, *RoboCup 2002: Robot Soccer World Cup VI*, volume 2752 of *Lecture Notes in Computer Science*, páginas 312–318. Springer Berlin/Heidelberg.

Egerstedt, M. e Martin, C. F. (2001). Optimal trajectory planning and smoothing splines. *Automatica*, 37(7):1057–1064.

El'youssef, E. S. (2007). *Análise e Aplicação de Controle de Sistemas Não Lineares com Observadores de Estados: Controle por Realimentação Linearizante Robusta e Controle Energy Shaping*. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, Brasil.

Endres, H., Feiten, W., e Lawitzky, G. (1998). Field test of a navigation system: autonomous cleaning in supermarkets. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, páginas 1779–1781.

Engelberger, J. F. (1993). Health-care robotics goes commercial: The "helpmate" experience. *Robotica*, 11(6):517–523.

Excell, J. (2008). Subsea saviours. *Engineer*, 293(7760):22–25.

Feron, E., Apkarian, P., e Gahinet, P. (1996). Analysis and synthesis of robust control systems via parameter-dependent lyapunov functions. *IEEE Transactions on Automatic Control*, 41(7):1041–1046.

Ferreira, A., Pereira, F. G., Vassallo, R. F., Bastos Filho, T. F., e Sarcinelli Filho, M. (2009). An approach to avoid obstacles in mobile robot navigation: the tangential escape. *SBA: Controle & Automação, Sociedade Brasileira de Automática*, 19(4):393–401.

Fialho, A. S., Búrigo, A. C., Jurkovski, B., de Azevedo, C. K., Adolfo, C., Dalbem, C. M., de Melo Leonardi, E., da Fontoura Beltrão, F., Chies, F. A., Silva, G. H., Schmidt, I. A., Lopes, I. F., de Mello Bones da Rocha, J. P., da Silveira, K. S., Barreto, L. C. M., Zawacki, L. F., Cavinato, M. V., e Barone, D. A. C. (2010). RoboPET team description paper. Proceedings of 14th International RoboCup Symposium, Singapore, June 2010.

Franco, A. C. S. (2006a). *Geração e Controle de Trajetória de Robôs Móveis Omnidirecionais*. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Mecatrônica, Universidade Federal da Bahia, Salvador, Brasil.

Franco, A. L. D. (2006b). *Controle Não Linear Robusto: Um Método Baseado em uma Linearização por Realimentação*. Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, Brasil.

Freitas, A. A. d. (2007). *Linearização Exata Adaptativa Aplicada ao Controle Preditivo de Processo de Nível Multivariável*. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade de Brasília, Brasília, Brasil.

Gahinet, P., Apkarian, P., e Chilali, M. (1996). Affine parameter-dependent Lyapunov functions and real parametric uncertainty. *IEEE Transactions on Automatic Control*, 41(3):436–442.

Gahinet, P., Nemirovski, A., Laub, A. J., e Chilali, M. (1995). *LMI Control Toolbox for use with Matlab, User's Guide, The Math Works Inc., Natick, MA.*

Garcia, E., Jimenez, M., Santos, P., e Armada, M. (2007). The evolution of robotics research. *IEEE Robotics Automation Magazine*, 14(1):90–103.

Garcia, G. e Bernussou, J. (1995). Pole assignment for uncertain systems in a specified disk by state feedback. *IEEE Transactions on Automatic Control*, 40(1):184 –190.

Garrido, B. M., Bittencout, F. R., Zarate, L. R., e Becker, M. (2002). Aplicação de redes neurais na geração de trajetórias para robôs móveis. In *In: Anais do II Congresso Nacional de Engenharia Mecânica. João Pessoa - PB*, volume 1, páginas 1–8.

Ghaoui, L. E. e lulian Niculescu, S., editors (2000). *Advances in Linear Matrix Inequality Methods in Control*. Society for Industrial and Applied Mathematics, Philadephia, PA.

Gomes, G. K. (2006). *Controle preditivo em tempo-real para seguimento de trajetória de veículos autônomos*. Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis.

Gu, D. e Hu, H. (2002). Neural predictive control for a car-like mobile robot. *International Journal of Robotics and Autonomous Systems*, 39(2):73–86.

Gu, Y. e Veloso, M. (2009). Effective multi-model motion tracking using action models. *Source International Journal of Robotics Research archive*, 28(1):3–19.

Guillard, H. e Bourlès, H. (2000). Robust feedback linearization. In *Proceedings of the* 14th *International Symposium on Mathematical Theory of Networks and Systems (MTNS'2000), Perpignan, França, Junho 2000.*

Gurzoni, J., Martins, M., Tonidandel, F., e Bianchi, R. (2011). On the construction of a robocup small size league team. *Journal of the Brazilian Computer Society*, 17:69–82.

Haddad, W. e Bernstein, D. (1992). Controller design with regional pole constraints. *IEEE Transactions on Automatic Control*, 37(1):54–69.

Hill, C. S., Amodeo, A., Joseph, J. V., e Patel, H. R. H. (2008). Nano-and microrobotics: How far is the reality? *Expert Review of Anticancer Therapy*, 8(12):1891–1897.

Hirata, Y., Song, H., Wang, Z., e Kosuge, K. (2007). Structural design for omni-directional mobile base of passive-type mobile robot. páginas 192–197.

Horswill, I. (1993). Polly: A vision-based artificial agent. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93*, páginas 824–829. Press.

Huang, L., Y. S. Lim, David Lee, e Christopher E. L. Teoh (2004). Design and analysis of a four-wheel omnidirectional mobile robot. In *Proceedings of the Second International Conference on Autonomous Robots and Agents, ICARA 2004, Massey University, Palmerston North, New Zealand*, páginas 425–428.

Hunt, L., Su, R., e Meyer, G. (1983). Global transformations of nonlinear systems. *IEEE Transactions on Automatic Control*, 28(1):24–31.

Huntsberger, T. L., Trebi-Ollennu, A., Aghazarian, H., Schenker, P. S., Pirjanian, P., e Nayar, H. D. (2004). Distributed control of multi-robot systems engaged in tightly coupled tasks. *Autonomous Robots*, 17(1):79–92.

Iocchi, L., Matsubara, H., Weitzenfeld, A., e Zhou, C., editors (2009). *RoboCup 2008: Robot Soccer World Cup XII [papers from the 12th annual RoboCup International Symposium, Suzhou, China, July 15-18, 2008]*, volume 5399 of *Lecture Notes in Computer Science*. Springer.

Isidori, A. (1989). *Nonlinear Control Systems - An Introduction*. Springer-Verlag.

Jr., J. A. G., Nascimento, E., Malheiro1, D., Zanatto1, F., Francischini1, G., Pereira, L. R. A., Cortez, M., Tebet, B., Martinez, S., Bianchi, R. A. C., e Tonidandel, F. (2010). RoboFEI 2010 team description paper. Proceedings of 14th International RoboCup Symposium, Singapore, June 2010.

Jung, M.-J., Kim, H.-S., Kim, S., e Kim, J.-H. (2000). Omnidirectional mobile base OK-II. volume 4, páginas 3449–3454.

Kalmár-Nagy, T., D'Andrea, R., e Ganguly, P. (2004). Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. *Robotics and Autonomous Systems*, 46(1):47–64.

Kececi, F. (2009). Fireproofing the firefighting robot. *Photonics Spectra*, 43(6):43–45.

Khalil, H. K. (2002). *Nonlinear Systems*. Prentice Hall, 3^a ed.

Kim, M. S., Shim, J. H., e Lee, J. J. (2000). Design of robust adaptive controller for a mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, páginas 1816–1820, Kagawa University, Takamatsu, Japan.

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., e Osawa, E. (1997). Robocup: The robot world cup initiative. In *AGENTS '97: Proceedings of the first international conference on Autonomous agents*, páginas 340–347, New York, NY, USA. ACM.

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., e Matsubara, H. (1998a). Robocup: A challenge problem for ai and robotics. In *RoboCup-97: Robot Soccer World Cup I*, páginas 1–19, London, UK. Springer-Verlag.

Kitano, H., Tambe, M., Stone, P., Veloso, M. M., Coradeschi, S., Osawa, E., Matsubara, H., Noda, I., e Asada, M. (1998b). The robocup synthetic agent challenge 97. In *RoboCup-97: Robot Soccer World Cup I*, páginas 62–73, London, UK. Springer-Verlag.

Kolmanovsky, I. e McClamroch, N. H. (1995). Developments in nonholonomic control problems. *IEEE Control Systems Magazine*, 15(6):20–36.

Kottke, H. d. C. (2005). *Uma Visão Global da Dinâmica de Newton-Euler Aplicada a robôs Manipuladores*. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Mecânica, Universidade de Taubaté, Taubaté, Brasil.

Kraft, T. e Hofmann, A. (2003). Vienna cubes: Robocup F180 small size league team. *OGAI Journal (Oesterreichische Gesellschaft fuer Artificial Intelligence)*, 22(3):14–18.

Kwok, C. e Fox, D. (2004). Map-based multiple model tracking of a moving object. *Proceedings of Eight RoboCup International Symposium, Lisbon, Portugal.*

Lages, W. F. (1998). *Controle e Estimação de Posição e Orientação de Robôs Móveis*. Dissertação de Mestrado, Instituto Tecnológico de Aeronáutica, São José dos Campos.

Lages, W. F. e Alves, J. A. V. (2006). Controle de robôs móveis em tempo real utilizando controle preditivo baseado em modelo linearizado. In *Anais do XVI Congresso Brasileiro de Automática*, Salvador – BA – Brasil.

Langner, C. G. (2004). *Síntese de Controladores $\mathcal{H}_2$ e $\mathcal{H}_\infty$ para Sistemas Sujeitos a Incertezas e/ou Restrições no Domínio do Tempo*. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná, Curitiba, Brasil.

Laue, T., Fritsch, S., Huhn, K., Humann, A., Mester, M., Peter, J., Reich, B., e Trocha, M. (2010). B-Smart team description for robocup 2010. Proceedings of 14th International RoboCup Symposium, Singapore, June 2010.

Leite, V. J. S., Montagner, V. F., Oliveira, P. J. d., Oliveira, R. C. L. F., Ramos, D. C. W., e Peres, P. L. D. (2004). Estabilidade robusta de sistemas lineares através de desigualdades matriciais lineares. *SBA: Controle & Automação, Sociedade Brasileira de Automática*, 15:24 – 40.

Leite, V. J. S. e Peres, P. L. D. (2005). Pole location control design of an active suspension system with uncertain parameters. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 43:561–579.

Liu, J.-S., Liang, T.-C., e Lin, Y.-A. (2004). Realization of a ball passing strategy for a robot soccer game: a case study of integrated planning and control. *Robotica*, 22(03):329–338.

Liu, Y., Zhu, J. J., Williams, II, R. L., e Wu, J. (2008). Omni-directional mobile robot controller based on trajectory linearization. *Robotics and Autonomous Systems*, 56(5):461–479.

Loebis, D., Sutton, R., Chudley, J., e Naeem, W. (2004). Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system. *Control Engineering Practice*, 12(12):1531–1539.

Lofberg, J. (2004). Yalmip : a toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, páginas 284–289.

Martins, F. N., Celeste, W. C., Carelli, R., Sarcinelli-Filho, M., e Bastos-Filho, T. F. (2008). An adaptive dynamic controller for autonomous mobile robot trajectory tracking. *Control Engineering Practice*, 16(11):1354–1363.

Masey, R. J. M., Gray, J. O., Dodd, T. J., e Caldwell, D. G. (2010). Guidelines for the design of low-cost robots for the food industry. *Industrial Robot*, 37(6):509–517.

Matía, F., Jiménez, A., AL-Hadithi, B. M., Rodríguez-Losada, D., e Gálan, R. (2006). The fuzzy Kalman filter: State estimation using possibilistic techniques. *Fuzzy Sets and Systems*, 156(16):2145–2170.

Mills, J. K. e Ing, J. G. L. (1996). Dynamic modeling and control of a multi-robot system for assembly of flexible payloads with applications to automotive body assembly. *Journal of Robotic Systems*, 13(12):817–836.

Miyagi, P. E. e Villani, E. (2004). Mecatrônica como solução de automação. *Revista Ciências Exatas*, 9/10(1-2):53–59.

Montagner, V., Leite, V., Oliveira, R., e Peres, P. (2006). State feedback control of switched linear systems: An LMI approach. *Journal of Computational and Applied Mathematics*, 194(2):192–206.

Mori, T. e Kokame, H. (2000). A parameter-dependent Lyapunov function for a polytope of matrices. *IEEE Transactions on Automatic Control*, 45(8):1516–1519.

Murata, S. e Hirose, T. (1993). Onboard locating system using real-time image processing for a self-navigating vehicle. *IEEE Transactions on Industrial Electronics*, 40(1):145–154.

Nelson, W. (1989). Continuous steering-function control of robot carts. *IEEE Transactions on Industrial Electronics*, 36(3):330–337.

Neto, A. T. (1999). Parameter dependent Lyapunov functions for a class of uncertain linear systems: an LMI approach. In *Proceedings of the 38th IEEE Conference on Decision and Control*.

Nijmeijer, H. e Van der Schaft, A. J. (1990). *Nonlinear Dynamical Control Systems*. Springer-Verlag.

Nouyan, S., Gro$\tilde{A}\ddot{Y}$, R., Dorigo, M., Bonani, M., e Mondada, F. (2006). Group Transport Along a Robot Chain in a Self-Organised Robot Colony. In *Proc. of the 9th Int. Conf. on Intelligent Autonomous Systems*, páginas 433–442, Amsterdam, The Netherlands. IOS Press.

Oliveira, H. P., Sousa, A. J., Moreira, A. P., e Costa, P. J. (2008a). Dynamical models for omni-directional robots with 3 and 4 wheels. In *Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics*, volume 1, páginas 189–196, Funchal, Portugal.

Oliveira, H. P., Sousa, A. J., Moreira, A. P., e Costa, P. J. (2008b). Precise modeling of a four wheeled omni-directional robot. In *Proceedings of the 8th Conference on Autonomous Robot Systems and Competitions*, páginas 57–62, Aveiro, Portugal.

Oliveira, M. C., Bernussou, J., e Geromel, J. C. (1999). A new discrete-time robust stability condition. *Systems and Control Letters*, 37(4):261–265(5).

Oliveira, R. C. L. F. e Peres, P. L. D. (2010). Tutoriais do XVIII congresso brasileiro de automática, capítulo análise e controle de sistemas lineares por meio de desigualdades matriciais lineares. *Cultura Acadêmica Editora*, páginas 203–227.

Oliveira, V. M. d. (2001). *Técnicas de controle de robôs móveis*. Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis.

Ottoni, G. d. L. e Lages, W. F. (2003). Navegação de robôs móveis em ambientes desconhecidos utilizando sonares de ultra-som. *SBA: Controle & Automação, Sociedade Brasileira de Automática*, 14:402–411.

Oubbati, M. (2006). *Neural Dynamics for Mobile Robot Adaptive Control*. Tese de Doutorado, Institut f*ü*r Parallele und Verteilte Systeme der Universit*ä*t Stuttgart, Tag der m*ü*ndlichen Pr*ü*fung, Junho 2006.

Ould-Khessal, N. (2005). Design and implementation of a robot soccer team based on omni-directional wheels. páginas 544–549.

Palhares, R. M. e Peres, P. L. D. (1999). Robust $\mathcal{H}_\infty$-filtering design with pole placement constraint via linear matrix inequalities. *Journal of Optimization Theory and Applications*, 102:239–261.

Papadopoulos, E. e Misailidis, M. (2007). On differential drive robot odometry with application to path planning. In *Proceedings of the European Control Conference*, páginas 5492–5499, Kos, Greece.

Paromtchik, I. E. e Asama, H. (2000). A motion generation approach for an omnidirectional vehicle. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation. San Francisco, CA, USA*, volume 2, páginas 1213–1218.

Paromtchik, I. E. e Rembold, U. (1994). A practical approach to motion generation and control for an omnidirectional mobile robot. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation. San Diego, CA*, volume 4, páginas 2790–2795.

Pranggonoh, Y., Ng, B. S., Yang, T., Kwong, A. L., Yue, P. K., e Zhou, C. (2010). Field Rangers team description paper. Proceedings of 14th International RoboCup Symposium, Singapore, June 2010.

Purwin, O. e Andrea, R. D. (2006). Trajectory generation and control for four wheeled omnidirectional vehicles. *Robotics and Autonomous Systems*, 54(1):13–22.

Purwin, O., D'Andrea, R., e Lee, J.-W. (2008). Theory and implementation of path planning by negotiation for decentralized agents. *Robotics and Autonomous Systems*, 56(5):422436.

Ramirez, A. R. G., Pieri, E. R., e Guenther, R. (2003). Controle em cascata de um manipulador robótico com um elo e uma transissão flexível. *SBA: Controle & Automação, Sociedade Brasileira de Automática*, 14(4):393–401.

Reis, G. A. d. (2005). *Controle $\mathcal{H}_\infty$ não linear de robôs móveis com rodas*. Dissertação de Mestrado, Universidade de São Paulo, São Carlos.

Rohr, E. R. (2008). *Análise de uma Classe de Sistemas Não-Lineares Incertos Mediante Linearização por Realimentação de Estados*. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Brasil.

Rojas, R. e Förster, A. G. (2006). Holonomic control of a robot with an omnidirectional drive. *Künstliche Intelligenz*, 20(2):12–17.

Sadhu, S., Mondal, S., Srinivasan, M., e Ghoshal, T. K. (2006). Sigma point Kalman filter for bearing only tracking. *Signal processing*, 86(12):3769–3777.

Sakai, H. e Amasaka, K. (2007). The robot reliability design and improvement method and the advanced Toyota production system. *Industrial Robot*, 34(4):310–316.

Schroeder, G. N., Espíndola, D. B., da C. Botelho, S. S., de L. Bicho, A., e de Oliveira, V. M. (2005). Simulador gráfico para controle de robôs móveis omnidirecionais. *INFOCOMP Journal of Computer Science*, 4(4):38–47.

Schulz, D., Burgrad, W., e Fox, D. (2003). People tracking with mobile robots using sample-based joint probabilistic data association filters. *International Journal of Robotics*, 22(2):99–116.

Selvatici, A. H. P. e Costa, A. H. R. (2007). Aprendizado da coordenação de comportamentos primitivos para robôs móveis. *SBA: Controle & Automação, Sociedade Brasileira de Automática*, 18(2):173–186.

Serrano, M. I. (2007). *Controle de Força de um Servoatuador Hidráulico Através da Técnica de Linearização por Realimentação.* Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil.

Sharbafi, M. A., Hoshyari, M., Esmaeelpourfard, S., Babersad, O. B., Haajseyedjavadi, M., e Esmaeeli, D. (2010). MRL team description 2010. Proceedings of 14th International RoboCup Symposium, Singapore, June 2010.

Siegwart, R. e Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots, 2^a ed. (Intelligent Robotics and Autonomous Agents).* MIT Press.

Simmons, R., Goodwin, R., Haigh, K., Koenig, S., , e O'Sullivan, J. (1997). A layered architecture for office delivery robots. In *First International Conference on Autonomous Agents, Marina del Rey, CA*, páginas 235–242.

Slotine, J. E. e Li, W. (1991). *Applied Nonlinear Control.* Prentice Hall.

Soares, S. J. P., Abdalla Júnior, M. A., Nepomuceno, E. G., e Leite, V. J. S. (2009). Regras de locomoção de robôs omnidirecionais baseadas em cadeias de markov e campos potenciais. In *IX Simpósio Brasileiro de Automação Inteligente, 2009, Brasília. Anais do IX Simpósio Brasileiro de Automação Inteligente, 2009*, volume 1, páginas 1–6.

Strang, G. (1988). *Linear Algebra and its Applications.* Harcourth Brace Jovanovich.

Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653.

Sujan, V. A., Meggiolaro, M. A., e Belo, F. A. W. (2006). A new technique in mobile robot simultaneous localization and mapping. *SBA: Controle & Automação, Sociedade Brasileira de Automática*, 17:189–204.

Sun, D. e Mills, J. K. (2002). Adaptive synchronized control for coordination of multirobot assembly tasks. *IEEE Transactions on Robotics and Automation*, 18(4):498–510.

Swetha, V., Jeganathan, E., e Shah, S. (2010). Robotic technology in ophthalmic surgery. *Current Opinion in Ophthalmology*, 21(1):75–80. cited By (since 1996) 0.

Tadokoro, S., Hayashi, M., Manabe, Y., Nakami, Y., e Takamori, T. (1995). On motion planning of mobile robots which coexist and cooperate with human. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots'*, volume 2, páginas 518–523.

Takaba, K., Iiguni, Y., e Tokumaru, H. (1996). An improved tracking Kalman filter using a multilayered neural network. *Mathematical and computer modelling*, 23(1-2):119–128.

Tambe, M. (1998). Implementing agent teams in dynamic multi-agent environments. *Applied Artificial Intelligence*, 12(2-3):189–210.

Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., e Schulz, D. (1999). Minerva: a second-generation museum tour-guide robot. In *Proc. IEEE International Conference on Robotics and Automation*, volume 3, páginas 1999–2005.

Vaidehi, V., Chitra, N., Chokkalingam, M., e Krishnan, C. (2001). Neural network aided Kalman filtering for multitarget tracking applications. *Computers and Electrical Engineering*, 27(2):217–228.

Vanantwerp, J. G. e Braatz, R. D. (2000). A tutorial on Linear and Bilinear Matrix Inequalities. *Journal of Process Control*, 10(4):363–385.

Visser, U., Ribeiro, F., Ohashi, T., e Dellaert, F., editors (2008). *RoboCup 2007: Robot Soccer World Cup XI, July 9-10, 2007, Atlanta, GA, USA*, volume 5001 of *Lecture Notes in Computer Science*. Springer.

Wada, M. (2005). An omnidirectional 4wd mobile platform for wheelchair applications. páginas 576–581.

Wada, M. (2009). Mechanism and control of a 4wd robotic platform for omnidirectional wheelchairs. páginas 4855–4862.

Watanabe, K. (1998). Control of ominidirectional mobile robot. In *2nd Int. Conf. on Knowledge-Based Intellegent Electronic Systems. Adelaide, Australia*, páginas 21–23.

Watanabe, K., Shiraishi, Y., Tzafestas, S. G., Tang, J., e Fukuda, T. (1998). Feedback control of an omnidirectional autonomous platform for mobile service robots. *Journal of Intelligent and Robotic Systems*, 22:315–330.

Williams, R.L., I., Carter, B., Gallina, P., e Rosati, G. (2002). Dynamic model with slip for wheeled omnidirectional robots. *IEEE Transactions on Robotics and Automation*, 18(3):285–293.

Wu, T., Duan, Z., e Wang, J. (2010). The design of industry mobile robot based on LLWIN function blocks language and embedded system. In *Proceedings of the 2nd International Conference on Computer Engineering and Technology, ICCET 2010*, volume 1, páginas V1–622–V1–625.

Wylie, C. R. e Barret, L. C. (1995). *Advanced Engineering Mathematics*. McGraw-Hill, 6ª edição.

Yamada, T., Watanabe, K., Kiguchi, K., e Izumi, K. (2001). Dynamic model and control for a holonomic omnidirectional mobile robot. *Autonomous Robots*, 11:173–189.

Yamamoto, Y. e Yun, X. (1994). Coordinating locomotion and manipulation of a modile manipulator. *IEEE Transactions on Automatic Control*, 39(6):1326–1332.

Yang, X., He, K., Guo, M., e Zhang, B. (1998). An intelligent predictive control approach to path tracking problem of autonomous mobile robot. volume 4, páginas 3301–3306.

Yoshimura, K., Barnes, N., Rönnquist, R., e Sonenberg, L. (2003). Towards real-time strategic teamwork: A robocup case study. In Kaminka, G., Lima, P., e Rojas, R., editors, *RoboCup 2002: Robot Soccer World Cup VI*, volume 2752 of *Lecture Notes in Computer Science*, páginas 342–350. Springer Berlin/Heidelberg.

Zhou, K., Doyle, J. C., e Glover, K. (1996). *Robust and optimal control*. Prentice Hall Upper Saddle River.

Zhoua, Q. e Aggarwalb, J. (2006). Object tracking in an outdoor environment using fusion of features and cameras. *Image and Vision Computing*, 24(11):1244–1255.

Zickler, S., Biswas, J., Bruce, J., Licitra, M., e Veloso, M. (2010). CMDragons 2010 extended team description. Proceedings of 14th International RoboCup Symposium, Singapore, June 2010.

Zickler, S. e Veloso, M. (2008). Playing Creative Soccer: Randomized Behavioral Kinodynamic Planning of Robot Tactics. In *RoboCup 2008: Robot Soccer World Cup XII*, páginas 414–425. Springer.