

Universidade do Porto
Faculdade de Engenharia
Departamento de Engenharia Electrotécnica e de Computadores



Controlo e Cooperação de Robôs Móveis Autónomos Omnidireccionais

André Gustavo Scolari Conceição

Dissertação submetida para a obtenção do grau de
Doutor em Engenharia Electrotécnica e de Computadores
pela Faculdade de Engenharia da Universidade do Porto

Trabalho realizado sob a orientação do Professor Doutor
António Paulo Gomes Mendes Moreira
e do Professor Doutor
Paulo José Cerqueira Gomes da Costa

Outubro 2007.

Apoio



Com o apoio do Programa $Al\beta$ an, Programa de bolsas de alto nível da União Europeia para América Latina, bolsa nº E04D028256BR.

Agradecimentos

Aos meus orientadores,

Prof. Dr. António Paulo Gomes Mendes Moreira

e

Prof. Dr. Paulo José Cerqueira Gomes da Costa,

pelo apoio e suporte no projecto e execução deste trabalho.

À todos os colegas do Laboratório de Controlo e Robótica da FEUP
pela parceria no decorrer do doutoramento, e aos amigos pelo
companherismo.

Principalmente aos meus pais Menotes e Adelina, e à minha irmã
Ioná, pelo apoio incondicional.

[DAS UTOPIAS]

Se as coisas são inatingíveis... ora!

Não é motivo para não querê-las...

Que tristes os caminhos, se não fora

A presença distante das estrelas!

Mário Quintana

Pensamento extraído do livro “Espelho mágico (1951)”.

Resumo

Robôs móveis autónomos têm despertado um crescente interesse em diversos tipos de aplicações, tornando-se uma área de intensas pesquisas científicas relacionadas com o projecto de novas arquitecturas de locomoção e com o desenvolvimento de novos algoritmos para controlo e navegação.

Este trabalho aborda dois pontos fundamentais para o sucesso de robôs móveis autónomos nas suas tarefas: o controlo individual dos robôs e a coordenação e cooperação entre os robôs. Os robôs móveis utilizados neste trabalho são do tipo omnidireccionais (ou holonômicos) com rodas. Estes tipos de robôs possuem uma vantagem significativa sobre os robôs não-holonômicos, pois podem movimentar-se em todas direcções independentemente de sua orientação.

Com o objectivo de controlar os robôs no seguimento de trajectórias, foram desenvolvidas duas estratégias de controlo. A primeira estratégia é baseada na realimentação parcial do estado do robô, acção esta mais relacionada com a reacção, busca-se com esta estratégia um controlador flexível com baixo tempo computacional. A segunda estratégia aborda o conceito de antecipação ou predição, ideal para este tipo de aplicação onde se conhece as referências futuras, que são as trajectórias a seguir. Resultados de simulação e resultados reais com os robôs no seguimento de trajectórias são comparados e analisados.

Para a coordenação e cooperação dos robôs, desenvolveu-se uma arquitectura para a execução de tarefas complexas em ambientes dinâmicos. A arquitectura é baseada na definição de diferentes tácticas, operações coordenadas, papéis, tarefas e acções. Os papéis permitem a configuração de um comportamento individual pela execução de tarefas específicas usando um determinado conjunto de acções. Finalmente, a arquitectura é testada em um ambiente dinâmico onde a cooperação e a coordenação individual dos robôs são essenciais.

Résumé

Les robots mobiles autonomes ont excité un croissant intérêt en divers types d'applications, en se rendant dans une aire d'intenses recherches scientifiques rapportées au projet de nouvelles architectures de locomotion et au développement de nouveaux algorithmes pour le contrôle et la navigation.

Le travail aborde deux points fondamentaux pour le succès des robots mobiles autonomes dans leurs tâches: le contrôle individuel des robots et la coordination des robots. Les robots mobiles utilisés dans ce travail sont du type omnidirectionnel (ou holonomiques) avec des roues. Le type de robots a un avantage significatif sur les robots non-holonomiques, parce qu'ils peuvent se mouvoir dans toutes les directions indépendamment de leur orientation.

Avec le but de contrôler les robots dans la suite de trajectoires, on a développé deux stratégies de contrôle. La première stratégie est basée en réalimentation partielle de l'état du robot, cette action est plus rapportée avec la réaction, avec cette stratégie on cherche un contrôleur flexible avec de bas temps computationnel. La deuxième stratégie aborde le concept d'anticipation ou prédiction, idéal pour le type d'application où on connaît les références futures qui sont les trajectoires suivantes. Des résultats de simulation et des résultats réels avec des robots dans la suite de trajectoires sont comparés et analysés.

Pour la coordination et coopération des robots, on a développé une architecture pour l'exécution des tâches complexes dans des environnements dynamiques. L'architecture est basée dans la définition de différentes tactiques, opérations coordonnées, rôles, tâches et actions. Des rôles permettent la configuration d'une conduite individuelle par l'exécution de tâches spécifiques en utilisant un déterminé ensemble d'actions. Pour terminer, l'architecture est testée dans un environnement dynamique où la coopération et la coordination individuelle des robots sont essentielles.

Abstract

Autonomous mobile robots have aroused an increased interest in diverse types of applications, becoming an area of intense scientific research related to project of new locomotion architectures and the development of new algorithms for control and navigation.

This work presents two fundamental approaches to success of autonomous mobile robots in its tasks: the individual control of the robots and the coordination and cooperation between them. Omnidirectional(or holonomics) wheeled mobile robots are used in this work. These robots have a significant advantage over the non-holonomics ones, they can move at any time in any direction regardless its orientation.

In order to deal with the individual control of the robots, two control strategies of trajectory tracking for omnidirectional vehicles had been developed. Giving attention to real-time applications, the first control strategy presents a low computational cost approach based on robot partial state feedback. The second strategy is based on the concept of prediction, being ideal for applications where the future references are known, in this case, the trajectories to follow. Real and simulation results of the robot trajectory tracking are compared and analyzed.

Also, an architecture for coordinating multiple autonomous robots in the execution of cooperative tasks is presented. The architecture is based on the definition of a strategy that uses different tactics, coordinated operations and roles. The roles enable to configure individual behaviour by performing specific tasks using a given set of actions. Finally, the architecture is tested in a dynamic environment where the cooperation and the individual control of the robots are essential.

Índice

Resumo	viii
Résumé	ix
Abstract	x
Índice	xi
Lista de Figuras	xv
Lista de Tabelas	xviii
Nomenclatura	xx
1 Introdução	1
1.1 Contexto e motivação	1
1.2 Objectivos e contribuições da tese	5
1.3 Estrutura da tese	7
2 Descrição dos Robôs e da Aplicação	9
2.1 Introdução	9
2.2 Aplicação - <i>Middle Size League</i> (MSL) da RoboCup	9
2.3 Descrição dos Robôs Móveis	11
2.3.1 <i>Hardware</i>	11
2.3.2 <i>Software</i>	14
2.4 Conclusão	16
3 Modelização de um Robô Móvel Omnidireccional	17
3.1 Introdução	17
3.2 Modelo da Motorização do Robô Móvel	18
3.2.1 Modelo Determinístico e Mínimos Quadrados	18
3.2.2 Sinal de Excitação do Sistema	20

3.2.3	Aplicação dos Mínimos Quadrados	21
3.2.4	Validação do Modelo Dinâmico Estimado	25
3.2.5	Estimador com Variável Instrumental	28
3.2.6	Resultados com os Outros Motores	30
3.2.7	Sintonia dos Controladores PID para os Motores	32
3.3	Modelo do Robô Móvel	37
3.3.1	Dinâmica do Robô Móvel	38
3.3.2	Cinemática do Robô Móvel	41
3.3.3	Identificação dos Parâmetros do Modelo do Robô	42
3.3.3.1	Método 1 - Velocidade em Regime Permanente	42
3.3.3.2	Método 2 - Tracção Nula	46
3.3.3.3	Método 3 - Momento de Inércia	49
3.3.3.4	Comparando Métodos	52
3.3.4	Identificação dos Parâmetros dos Motores	53
3.3.5	Parâmetros do Modelo do Robô	55
3.3.6	Resultados Reais e de Simulação	56
3.4	Conclusão	61
4	Controladores para Robôs Móveis Omnidireccionais	62
4.1	Introdução	62
4.2	Controlador Reactivo	64
4.2.1	Definição do Controlador	65
4.2.2	Restrição nos Sinais de Controlo	68
4.2.3	Optimização dos Parâmetros do Controlador	74
4.2.4	Resultados Experimentais	80
4.2.5	Conclusões	84
4.3	Controlador Preditivo	85
4.3.1	Estrutura do Controlador	88
4.3.2	Modelo do Processo	90
4.3.3	Função Custo	91
4.3.4	Definição da Trajectória de Referência para o Controlador	92
4.3.5	Lei de Controlo	94
4.3.6	Optimização Numérica da Função Custo	96

4.3.6.1	Método do gradiente	96
4.3.6.2	Método dos Gradientes Conjugados	97
4.3.7	Resultados Experimentais	99
4.3.8	Resultados de Simulações	100
4.3.8.1	Ensaio simulado 1 - ($N = 10$)	101
4.3.8.2	Ensaio simulado 2 - ($N = 8$)	105
4.3.8.3	Ensaio simulado 3 - ($N = 12$)	108
4.3.9	Resultados Reais	111
4.3.9.1	Ensaio real 1 - ($N = 10$)	111
4.3.9.2	Ensaio real 2 - ($N = 12$)	115
4.3.10	Conclusões	119
5	Arquitectura de Coordenação e Cooperação para Sistemas Multi-Robô	121
5.1	Introdução	121
5.2	Conceptualização	123
5.3	Descrição da Arquitectura	128
5.3.1	Estratégia e Táticas, Operações Coordenadas	130
5.3.2	Papéis, Tarefas e Acções	131
5.4	Implementação da Arquitectura Aplicada ao Domínio do Futebol Robótico	132
5.4.1	Supervisor	134
5.4.2	Decisão	136
5.5	Conclusões	138
6	Conclusões e Trabalhos Futuros	139
6.1	Síntese e Conclusões do Trabalho	139
6.2	Trabalhos Futuros	141
6.3	Considerações Finais	142
A	Locomoção de robôs móveis com rodas	143
A.1	Omnidireccionais	143
A.2	Tracção simples	144
A.3	Tracção diferencial	144

ÍNDICE

A.4	Tracção síncrona	145
A.5	Ackermann	146
Referências		147

Lista de Figuras

1.1	Robôs móveis comerciais.	3
1.2	Robôs móveis em universidades.	3
1.3	Robôs móveis da Equipa 5dpo-2000.	5
2.1	Campo e robôs.	11
2.2	Robôs móveis e uma roda omnidireccional.	13
2.3	Estrutura de controlo e comunicação.	14
3.1	Processo da motorização.	18
3.2	Ruído Branco Gaussiano.	21
3.3	Curva de velocidade do motor 1.	22
3.4	Velocidade medida e estimada para a <i>FTb</i> - Estimação.	23
3.5	Resultados da validação para as diferentes FTs.	26
3.6	Velocidade medida e estimada para a <i>FTb</i> - Validação.	27
3.7	Auto correlação do erro $Erro_r$ para estimação com Mínimos Quadrados e VI.	29
3.8	Motores do robô.	31
3.9	Sistema em malha fechada.	32
3.10	Resposta ao degrau do sistema discreto e contínuo.	33
3.11	Sistema em malha fechada com <i>feedforward</i>	35
3.12	Resultados com o controlador PID.	36
3.13	Configuração do robô e símbolos.	39
3.14	Atritos.	40
3.15	Velocidade e correntes para velocidade v - 1º ensaio.	44
3.16	Forças e binário <i>versus</i> velocidades.	45

LISTA DE FIGURAS

3.17	Curvas de velocidades.	47
3.18	Velocidades e acelerações, usadas na estimação.	48
3.19	Esquemático da experiência.	49
3.20	Gráficos dos Ensaios.	51
3.21	Recta que determina os valores de K_v e R_a - motor da roda 1. . .	54
3.22	Velocidade $v = 1$, simulação 1.	57
3.23	Velocidade $vn = 1$, simulação 2.	58
3.24	Velocidade $w = 1$, simulação 3.	59
3.25	Velocidade $w = 1$, simulação 3.	60
3.26	Comparação entre a modelização não-linear e a linear.	60
4.1	Representação esquemática do controlador e do robô móvel. . . .	66
4.2	Representação do robô móvel.	68
4.3	Planos das velocidades máximas do robô nos eixos v, vn, w	71
4.4	Pontos P_{ref} , $\overline{P}_{ref}$	73
4.5	Pontos P_{ref} , $\overline{P}_{ref}$	73
4.6	Trajectória "Pulso", para cálculo da função custo.	75
4.7	Mínimos das funções custo, trajectória "Pulso".	76
4.8	Trajectória "8".	78
4.9	Mínimos das funções custo, trajectória "8".	79
4.10	Trajectória "Pulso", simulada e real.	80
4.11	Trajectória "8", simulada e real.	81
4.12	Trajectória para teste.	82
4.13	Resultados do Algoritmo de restrição.	83
4.14	Metodologia do controlador.	89
4.15	Estrutura do Controlador.	90
4.16	Esquemático do robô.	91
4.17	Trajectória de referência para o controlador, W	93
4.18	Fluxograma da lei de controlo.	95
4.19	Trajectória "Diagonal".	100
4.20	Trajectória "Pulso".	100
4.21	Trajectória Diagonal - ensaio simulado 1 ($N = 10$).	103
4.22	Trajectória Pulso - ensaio simulado 1 ($N = 10$).	104

LISTA DE FIGURAS

4.23	Trajectória Diagonal - ensaio simulado 2 ($N = 8$).	106
4.24	Trajectória Pulso - ensaio simulado 2 ($N = 8$).	107
4.25	Trajectória Diagonal - ensaio simulado 3 ($N = 12$).	109
4.26	Trajectória Pulso - ensaio simulado 3 ($N = 12$).	110
4.27	Trajectória diagonal - ensaio real 1 ($N = 10$).	113
4.28	Trajectória pulso - ensaio real 1 ($N = 10$).	114
4.29	Trajectória diagonal - ensaio real 2.	117
4.30	Trajectória pulso - ensaio real 2 ($N = 12$).	118
4.31	Comparação entre o controlador preditivo e o controlador reactivo.	120
5.1	Estrutura típica de um sistema multi-agente.	125
5.2	Arquitectura de comunicação entre agentes.	125
5.3	Diagrama comportamental-hierárquico da arquitectura.	130
5.4	Fluxograma do <i>Loop</i> de controlo da arquitectura.	134
5.5	<i>Software</i> "Supervisor".	135
5.6	<i>Software</i> "Decisão".	137
A.1	Rodas omnidireccionais.	143
A.2	Tracção simples.	144
A.3	Tracção diferencial.	145
A.4	Tracção síncrona.	145
A.5	Ackermann.	146

Lista de Tabelas

2.1	Características físicas do robô.	13
3.1	Resultados da estimação para diferentes FTs	24
3.2	Resultados da validação para as diferentes FTs.	25
3.3	Valores Estimados.	29
3.4	Valores Estimados - Mínimos Quadrados.	30
3.5	Valores Estimados - Variável Instrumental.	30
3.6	Controladores PI dos motores do robô.	36
3.7	Velocidades aplicadas e forças e binário resultante.	43
3.8	Equações e coeficientes - método 1.	43
3.9	Equações e coeficientes - método 2.	49
3.10	Momentos de Inércia.	50
3.11	Coeficientes de atritos estimados.	52
3.12	Momentos de inércia estimados.	52
3.13	Parâmetros.	55
4.1	Valores mínimos da função custo para diferentes v_r , trajectória "Pulso".	77
4.2	Valores mínimos da função custo para diferentes v_r , trajectória "8".	77
4.3	Parâmetros da trajectória.	100
4.4	Parâmetros da trajectória.	100
4.5	Parâmetros ensaio simulado 1.	101
4.6	Trajectória diagonal e pulso - ensaio simulado 1 ($N = 10$).	102
4.7	Parâmetros ensaio simulado 2.	105
4.8	Trajectória diagonal e pulso - ensaio simulado 2 ($N = 8$).	105

LISTA DE TABELAS

4.9	Parâmetros ensaio simulado 3.	108
4.10	Trajectória diagonal e pulso - ensaio simulado 3 ($N = 12$).	108
4.11	Parâmetros ensaio real 1.	111
4.12	Trajectória diagonal e pulso - ensaio real 1 ($N = 10$).	112
4.13	Truncamentos - ensaio real 1.	112
4.14	Parâmetros ensaio real 2.	115
4.15	Trajectória diagonal e pulso - ensaio real 2 ($N = 12$).	115
4.16	Truncamentos - ensaio real 2.	116

Nomenclatura

AGV *Automated Guided Vehicles*

DMC *Dynamic Matrix Control*

EHAC *Extended Horizon Adaptive Control*

EPSAC *Extended Prediction Self-Adaptive Control*

FIFA *International Federation of Association Football*

GC_FR Método dos Gradientes Conjugados Fletcher-Reeves

GC_PR Método dos Gradientes Conjugados Polak-Ribiere

G Método do gradiente

GPC *Generalized Predictive Control*

IAD Inteligência Artificial Distribuída

IA Inteligência Artificial

LQ Linear Quadrático

MPC *Model Predictive Control*

MSE *Mean Square Error*

MSL *Middle Size League*

NMPC *Nonlinear Model Predictive Control*

LISTA DE TABELAS

PC	<i>Personal Computer</i>
PFC	<i>Predictive Functional Control</i>
PGP	Planeamento Global Parcial
PID	Proporcional-Integral-Derivativo
PP	Posicionamento de Pólos
PWM	<i>Pulse Width Modulation</i>
SGC	Sistema Global de Coordenadas
TLC	<i>Trajectory Linearization Control</i>
UPC	<i>Unified Predictive Control</i>
WGN	<i>White Gaussian Noise</i>

Capítulo 1

Introdução

1.1 Contexto e motivação

A robótica móvel, pode parecer não ser nada mais do que a redução da robótica convencional a três graus de liberdade. Entretanto, os robôs convencionais geralmente operam no seu próprio ambiente, no qual os operadores não ousam interferir sem primeiro desligar o sistema. Consequentemente, seu ambiente pode ser controlado com cuidado, pois nenhum obstáculo desconhecido deve interferir e não há ajustes sem observação prévia. Nada disto acontece com um robô móvel, em que o ambiente de operação, em via de regra, é partilhado com pessoas e outros objectos. Tais ambientes estão sujeitos a mudanças constantes, sendo esperado, ainda assim, que o robô opere de forma rápida e segura. A fim de conseguir tal comportamento, tem-se que confiar na qualidade dos sensores e actuadores que interagem com o ambiente ao seu redor sendo este conhecido ou não. Isto implica adquirir e manipular dados provenientes de diversos sensores acoplados ao robô e também gerar sinais de controlo que façam com que os actuadores tenham o funcionamento desejado.

Um robô móvel é uma combinação de vários componentes físicos(*hardware*) e computacionais(*software*). Conforme [Dudek & Jenkin \(2000\)](#), os componentes de um robô móvel podem ser considerados uma colecção de sub-sistemas para:

- Locomoção $\leftrightarrow$ capacidade para se movimentar em um ambiente.
- Sensorização $\leftrightarrow$ capacidade para medir suas características e do ambiente.

- Racionalização $\leftrightarrow$ capacidade para mapear as medidas em acções.
- Comunicação $\leftrightarrow$ capacidade para comunicar com o exterior.

Várias estratégias diferentes de locomoção foram abordadas pelos projectistas de robôs móveis, governadas pela aplicação e o domínio onde o robô vai actuar. Baseado no domínio da aplicação, os robôs móveis podem ser divididos em quatro vastas categorias (Dudek & Jenkin (2000)): Terrestres, Aquáticos, Aéreos e Espaciais.

Uma das principais ramificações dos robôs móveis com rodas(terrestres) são os *Automated Guided Vehicles* - AGVs ou robôs móveis de serviço, que inicialmente foram criados com a intenção de serem um sistema de transporte mais flexível: um transportador no qual não precisasse de uma esteira contínua ou uma mesa com cilindros (Murphy (2000)). Idealmente, um AGV seria capaz de para apanhar artigos fabricados ou parte deles e entregar com precisão ao destino desejado. Entretanto, a navegação em ambientes fechados ou abertos ainda está longe de ser uma tarefa simples. Para realizar a tarefa de navegação, o AGV tem que saber a sua localização no ambiente, planear o caminho de sua localização actual para seu destino, e ainda evitar colidir com trabalhadores, outros AGVs e ferramentas do chão-de-fábrica. Actualmente expandiu-se a área de actuação destes tipos de robôs, que podem ser utilizados em várias aplicações, como:

1. risco de vida significativo para humanos(no espaço, uso militar, ambientes hostis);
2. economicamente desvantajoso o uso de trabalhadores humanos(indústria, agricultura);
3. uso humanitário (cadeiras de roda, hospitais, salvamentos);
4. uso educacional (ferramenta poderosa para ensino de engenharia).

Empresas especializadas em robótica comercializam robôs de serviço móveis, atendendo um mercado em expansão. Por exemplo, na área médica, pode-se citar as cadeiras de rodas robotizadas da Invacare Corporation(Invacare (2007)), Figura 1.1(a). Na área da indústria, a empresa Mobile Robotics Inc(MobileRobots



(a) Cadeira de rodas robotizadas.

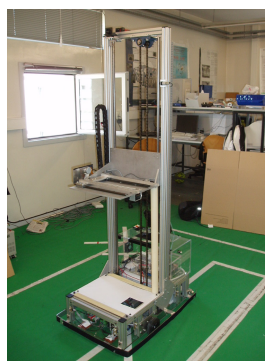


(b) AGV Powerbot.

Figura 1.1: Robôs móveis comerciais.



(a) Cadeira de rodas omnidireccional.



(b) AGV - FEUP.

Figura 1.2: Robôs móveis em universidades.

(2007)) comercializa robôs móveis de serviço (Figura 1.1(b)), possuindo sistemas de navegação e localização, comunicação *wireless*, sequenciamento de tarefas por software entre outras características. Em ambientes universitários existem grupos de pesquisa dedicados ao desenvolvimento de robôs móveis. Na Figura 1.2(a) apresenta uma cadeira de rodas com um sistema de locomoção omnidireccional. A cadeira foi desenvolvida no Grupo de Automação e Robótica da Universidade do Minho, e actualmente é comercializada pela Empresa SAR - Soluções de Automação e Robótica(SAR (2007)). Na Figura 1.2(b) apresenta-se um AGV desenvolvido no Laboratório de Controlo e Robótica da Faculdade de Engenharia da Universidade do Porto(FEUP). O AGV é um protótipo para transporte e armazenamento de pequenos volumes(Silva *et al.* (2007)), possui um controlo de movimento com tracção diferencial, movimentação autónoma, localização com

base em marcadores visuais, sensores ultra-som, sistema de elevação para transporte de pequenos volumes, comunicação de rede sem fios e identificação por código de barras dos volumes.

O estudo realizado nesta tese está enquadrado na categoria dos robôs terrestres mais especificamente os robôs móveis omnidireccionais com rodas. No Anexo A apresenta-se as principais configurações a respeito da locomoção de robôs móveis, sendo assim no restante da tese vamos focar apenas nos robôs moveis omnidireccionais com rodas.

Os robôs omnidireccionais ou holonômicos possuem uma vantagem significativa sobre os robôs não-holonômicos, pois não possuem restrições de movimentação, tornando esse tipo de robô uma opção atractiva em ambientes dinâmicos, ou em ambientes com pouco espaço de manobra como por exemplo células de processos ao nível do chão-de-fábrica. Além disso, o projecto de um controlador para robôs holonômicos é mais fácil, pois as equações relacionadas a movimentação são mais simples comparado as equações de robôs não-holonômicos que possuem restrições de movimentação em uma ou mais direcções.

Dois pontos são fundamentais para o sucesso dos robôs móveis em suas tarefas: o controlo individual do robô e a coordenação e cooperação dos robôs. Sendo assim, este é o problema que será abordado neste estudo. Numa primeira etapa do trabalho abordou-se o projecto de controladores para o seguimento de trajectórias dos robôs móveis omnidireccionais. Numa segunda etapa partiu-se para o projecto de uma arquitectura de coordenação e cooperação para os robôs.

Os robôs móveis omnidireccionais utilizados neste trabalho foram construídos pela equipa de futebol robótico 5dpo-2000, do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto. Os robôs participam das competições RoboCup na liga dos robôs médios (*F2000 - Middle Size League*). A equipa 5dpo-2000, desde 1999, tem participado em 9 competições internacionais da RoboCup e 7 eventos locais. Desde então, os principais tópicos de investigação são a modelização, controlo, fusão sensorial e coordenação de sistemas multi-robôs. O *know-how* adquirido ao longo desses anos tem sido transferido para projectos com a indústria em diversas áreas da engenharia, como por exemplo aplicações de processamento de imagem, sensores baseados em visão artificial para manipuladores robóticos, comandos remotos por

1.2 Objectivos e contribuições da tese

rádio, accionamento e controlo de motores, projecto e construção de AGVs, entre outros. Na Figura 1.3 apresenta-se a equipa de robôs móveis que foi construída no ano de 2004.



Figura 1.3: Robôs móveis da Equipa 5dpo-2000.

1.2 Objectivos e contribuições da tese

Os principais objectivos deste trabalho estão relacionados com o controlo e cooperação de robôs móveis omnidireccionais. Estas tarefas estão intimamente ligadas, pois não se consegue uma cooperação entre os robôs sem um bom controlo individual dos mesmos. Seguindo o mesmo raciocínio, não se consegue um bom controlador individual para os robôs sem o estudo e modelização das características cinemáticas e dinâmicas dos mesmos. Sendo assim, foi definida uma sequência de objectivos de forma a alcançar o desenvolvimento de controladores e por fim uma arquitectura de coordenação e cooperação para os robôs:

- estimação de um modelo linear para o sistema que engloba o motor, a caixa redutora do motor e o encoder, para as rodas do robô. O sistema (motor+caixa redutora+encoder) possui não-linearidades, como por exemplo folgas e atritos da caixa redutora. Obter uma estimação desses parâmetros é difícil e provavelmente pouco eficaz, sendo assim optou-se por estimar um

1.2 Objectivos e contribuições da tese

modelo linear que represente de forma satisfatória o sistema. O objectivo principal é usar o modelo para testes e sintonia dos controladores PIDs das rodas do robô;

- desenvolver um modelo não-linear cinemático e dinâmico do robô, tendo em conta restrições de limites de corrente e tensão nos motores. Além disso, inserir componentes dinâmicas como atritos viscosos e de coulomb no modelo dinâmico do robô. O objectivo principal é usar o modelo para o projecto e estrutura de controladores;
- projectar e implementar controladores para robôs omnidireccionais baseados em duas abordagens principais: reacção(realimentação) e predição. Os controladores devem respeitar restrições de tempo real das aplicações e as características dinâmicas do robô;
- projectar e implementar uma arquitectura de coordenação e cooperação para robôs autónomos na execução de tarefas cooperativas. O objectivo é obter uma arquitectura que permita flexibilidade e eficiência na operação de sistemas multi-robôs em ambientes dinâmicos.

As contribuições do trabalho estão relacionadas com novos métodos, algoritmos e estratégias desenvolvidas para alcançar os objectivos. A seguir são apresentadas as principais contribuições da tese:

- estudo e síntese dos trabalhos mais actuais e relevantes da área;
- aplicação de técnicas baseadas em estimadores mínimos quadrados e variável instrumental para estimar modelos lineares de sistemas dinâmicos com não linearidades;
- desenvolvimento de três métodos para a estimação de parâmetros das equações dinâmicas do robô. Os parâmetros estimados são os atritos viscosos, atritos de coulomb e o momento de inércia do robô;
- desenvolvimento de um modelo não-linear completo para os robôs móveis, incluindo características cinemáticas, dinâmicas, e elementos não-lineares;

- projecto e implementação de uma estratégia de controlo baseada em realimentação com baixa carga computacional (≤ 2 milissegundos em um PC@2Ghz), para robôs omnidireccionais. Para esta estratégia foi desenvolvido um algoritmo que analisa os sinais de controlo aplicados no robô, baseado nas restrições de velocidades dos motores e no modelo cinemático do robô;
- projecto e implementação de uma estratégia de controlo baseada em um controlador preditivo não-linear para robôs omnidireccionais. Uma nova abordagem neste tipo de aplicação é apresentada, onde métodos numéricos de optimização não-linear são usados para obter a minimização da função custo do controlador;
- projecto e implementação de uma arquitectura de cooperação para sistemas multi-robôs. A arquitectura é baseada na definição de uma estratégia que utiliza vários níveis hierárquicos para controlo dos robôs em ambientes dinâmicos. Além disso a arquitectura pode ser utilizada como uma plataforma base para desenvolvimento de pesquisa em Inteligência Artificial(IA), onde técnicas relacionadas a IA podem ser aplicadas e testadas nos agentes autónomos(robôs).

1.3 Estrutura da tese

A tese está dividida em seis Capítulos. Inicialmente no Capítulo 1 é abordado o tema da tese e as suas contribuições. No Capítulo 2 descreve-se o objecto de investigação que são os robôs móveis omnidireccionais e a aplicação e o domínio na qual os robôs estão inseridos.

No Capítulo 3 apresenta-se as técnicas de modelização desenvolvidas para os robôs. Numa primeira etapa descreve-se a estimação de um modelo linear para a motorização dos robôs, e na segunda etapa descreve-se um modelo cinemático e dinâmico não-linear para os robôs. Resultados de simulação e resultados reais são apresentados no decorrer do Capítulo.

No Capítulo 4, as estratégias de controlo para os robôs no seguimento de trajectórias são apresentadas, assim como resultados de simulação e resultados reais com um robô no seguimento de trajectórias.

O Capítulo 5 aborda sistemas multi-robôs em ambientes dinâmicos, apresenta-se uma arquitectura de coordenação e cooperação para a equipa de robôs. Uma implementação da arquitectura aplicada ao domínio do futebol robótico é apresentada.

Finalmente no Capítulo 6 são apresentadas as conclusões e trabalhos futuros.

Capítulo 2

Descrição dos Robôs e da Aplicação

2.1 Introdução

Neste Capítulo primeiramente é apresentada a aplicação dos robôs móveis, e em seguida são descritas as suas características físicas e computacionais.

2.2 Aplicação - *Middle Size League*(MSL) da RoboCup

A liga de robôs médios(MSL) da RoboCup([Robocup \(2007\)](#)) é uma aplicação que tem como domínio de testes o futebol robótico. O RoboCup é uma iniciativa internacional para investigação e educação. É uma tentativa para promover a Inteligência Artificial (IA) e pesquisa da robótica em geral, proporcionando um problema extremamente complexo onde uma vasta gama de tecnologias podem ser integradas e analisadas. Para este propósito, o futebol robótico foi escolhido como um domínio de teste. Para que uma equipa de robôs execute um jogo de futebol de facto, devem ser incorporadas várias tecnologias incluindo: projecto de agentes autónomos, colaboração multi-agentes, aquisição de estratégia, raciocínio em tempo-real, robótica, controlo, processamento de imagem e fusão sensorial. A federação da RoboCup definiu a seguinte meta para a iniciativa: “No ano de 2050, uma equipa de robôs autónomos humanóides, ser capaz de vencer a equipa campeã do mundo de futebol, num encontro disputado de acordo com as regras

2.2 Aplicação - *Middle Size League*(MSL) da RoboCup

da FIFA(*International Federation of Association Football*)”[Kitano et al. \(1997b\)](#).

Na liga de robôs médios, duas equipas de robôs com todos seus sensores *on-board* jogam futebol num campo. Características relevantes do campo são distinguidas por cores. A comunicação entre os robôs é através de comunicação *wireless*. Durante o jogo não é permitida intervenção humana, excepto para inserir ou retirar robôs do campo. As principais regras e definições da liga de robôs médios para o ano de 2007 são ([MSL \(2007\)](#)):

- As dimensões do campo são 18x12 metros;
- As dimensões das balizas são: 2 metros entre os postes e um metro entre a barra superior e o chão. As balizas são de cores diferentes: azul e amarelo;
- Características da bola: esférica, com uma circunferência maior que 68 cm e menor que 70 cm. A bola é da cor de laranja;
- Número de jogadores(robôs): as equipas podem jogar com no máximo seis jogadores, sendo um deles o guarda-redes. A diferença máxima do número de jogadores entre as equipas que começam o jogo é 1, se a equipa que tiver menos jogadores possuir 4 ou mais;
- Tamanho/peso dos robôs: cada robô deve possuir uma configuração incluindo seus actuadores que a sua projecção no chão esteja contida em um quadrado de no mínimo 30x30cm e no máximo 50x50cm. A altura do robô deve ser no mínimo 40 cm e no máximo 80 cm, e o peso máximo permitido é 40kg. Os robôs tem que ser da cor preta predominantemente, com marcadores das cores azul claro ou magenta;
- *Referee Box*: é uma aplicação que auxilia o árbitro, responsável por enviar os comandos do arbitro para as equipas que estão a jogar e também por fazer *logs* dos jogos. O operador da *Referee Box* é um árbitro assistente;
- Tempo de jogo: dois períodos iguais de 15 minutos. O intervalo é de 5 minutos.

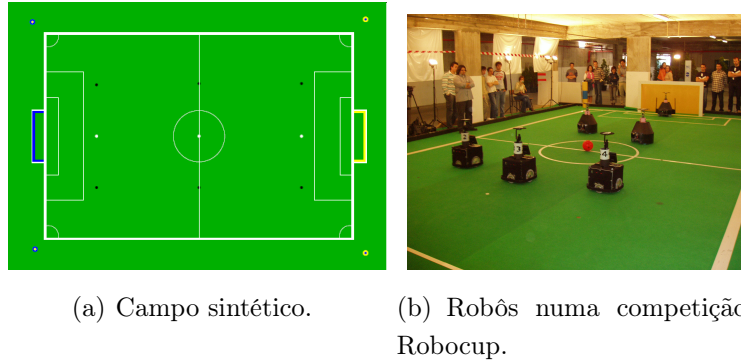


Figura 2.1: Campo e robôs.

Na Figura 2.1 apresentam-se o desenho do campo conforme as regras da federação Robocup e duas equipas posicionadas para o início do jogo em uma competição da Robocup.

2.3 Descrição dos Robôs Móveis

Nesta Secção serão descritas as principais características da equipa de robôs (Moreira *et al.* (2006)). Todo o projecto da mecânica, do *hardware* e do *software* foi desenvolvido por docentes e alunos da Faculdade de Engenharia, fazendo com que o conhecimento adquirido com as experiências nas gerações anteriores de robôs sejam preservado para as gerações seguintes.

2.3.1 *Hardware*

Todos os robôs da equipa possuem a mesma estrutura mecânica, excepto o guarda-redes. O guarda-redes (Figura 2.2(b)) possui um braço rotativo e um *kicker* (dispositivo de chuto electromecânico), usados para afastar a bola. Os robôs (Figura 2.2(a)) possuem um *kicker* mais potente, com o objectivo de chutar a bola com maior velocidade para dificultar a defesa das outras equipas. O *kicker* é um dispositivo electromecânico que possui um solenóide que é pulsado para puxar um embolo de ferro, fazendo com que vá de encontro com a bola. Para fornecer a energia necessária durante o pulso é usada uma bateria de condensadores, previamente carregada.

2.3 Descrição dos Robôs Móveis

Os robôs são equipados com 4 rodas omnidireccionais, conectadas a motores DC. Associado a cada roda está um encoder industrial. A roda frontal e a roda traseira são menores, conforme a Tabela (2.1), para permitir a presença do *kicker*. Para compensar o tamanho menor destas rodas, as caixas redutoras possuem uma redução menor, mantendo a velocidade máxima linear similar em todas rodas. Cada conjunto motor-roda-encoder está conectado a uma placa de controlo e a um drive de potência com limitação de correntes para os motores. A placa de controlo possui um microcontrolador que mede as velocidades das rodas e controla essas mesmas velocidades através de controladores PID. A frequência de amostragem deste nível mais baixo de controlo é 100 Hz. Os controladores PIDs estão conectados a um PC (*Personal Computer*) por uma porta série RS-232, a 115200 baud. O PC do robô dispõe de um processador Celeron 2 GHz. Não é utilizado um disco rígido, sendo assim o sistema operativo Linux e os programas são armazenados em um cartão de memória Compact Flash de 256Mbyte, que aparece ao sistema como um disco IDE.

Outro módulo essencial é o da visão do robô, trata-se de um sistema de visão omnidireccional para captar as principais características do ambiente. Estas informações são utilizadas para estimação da posição do robô, dos adversários e da bola. O sistema de visão também fornece o tempo de amostragem de controlo do robô ($t_s=40$ milissegundos).

Os robôs omnidireccionais ou holonômicos são capazes de se movimentar em todas direcções (Braunl (2003), Pin & Killough (1994)). No caso dos robôs móveis com rodas, essa característica é obtida a partir das rodas omnidireccionais. Na Figura 2.2(c) apresenta-se uma das rodas utilizadas nos robôs. As rodas foram projectadas pela equipe 5dpo-2000 e construídas pela empresa Tecnogial Lda (Tecnogial (2007)). Uma das características de uma configuração com quatro rodas é que com quatro pontos de contactos com o chão qualquer irregularidade pode provocar que uma ou mais rodas percam o contacto com o chão. Para prevenir isso, as rodas traseiras dos robôs possuem um sistema de suspensão. Na Tabela 2.1 apresenta-se dados das características físicas do robô. Na Figura 2.2 apresenta-se os robôs da equipa 5dpo-2000 e na Figura 2.2(b) em frente ao guarda-redes apresenta-se um robô da liga *small-size*.

2.3 Descrição dos Robôs Móveis

Dimensões	0.46x0.50 m
Altura	0.80 m
Peso	35 Kg
Rodas Laterais (raio)	0.098 m
Roda Dianteira (raio)	0.065 m
Roda Traseira (raio)	0.069 m

Tabela 2.1: Características físicas do robô.

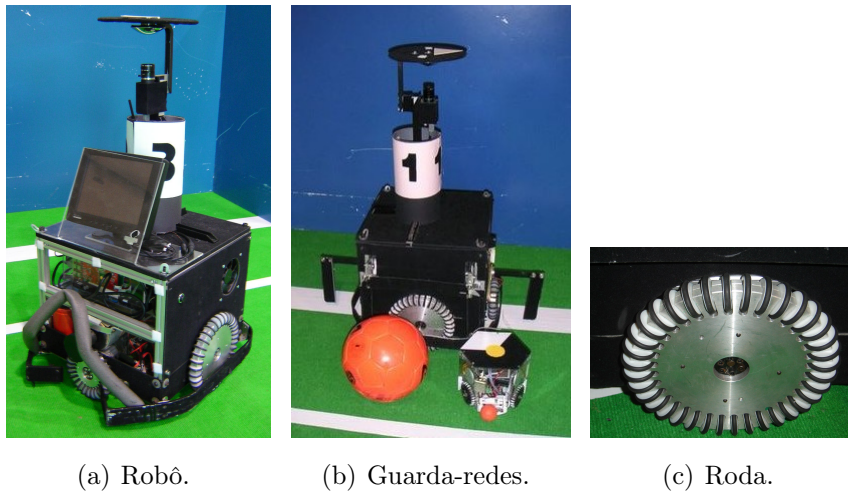


Figura 2.2: Robôs móveis e uma roda omnidireccional.

2.3.2 Software

A estrutura de controlo e comunicação dos robôs é apresentada na Figura 2.3. A estrutura é composta por três componentes principais: *Referee Box*, Supervisor e Robô. Existem quatro níveis distintos de software para gerir o funcionamento destes componentes, que são definidos por limites físicos mas também pelo tipo de informações transmitidas e processadas:

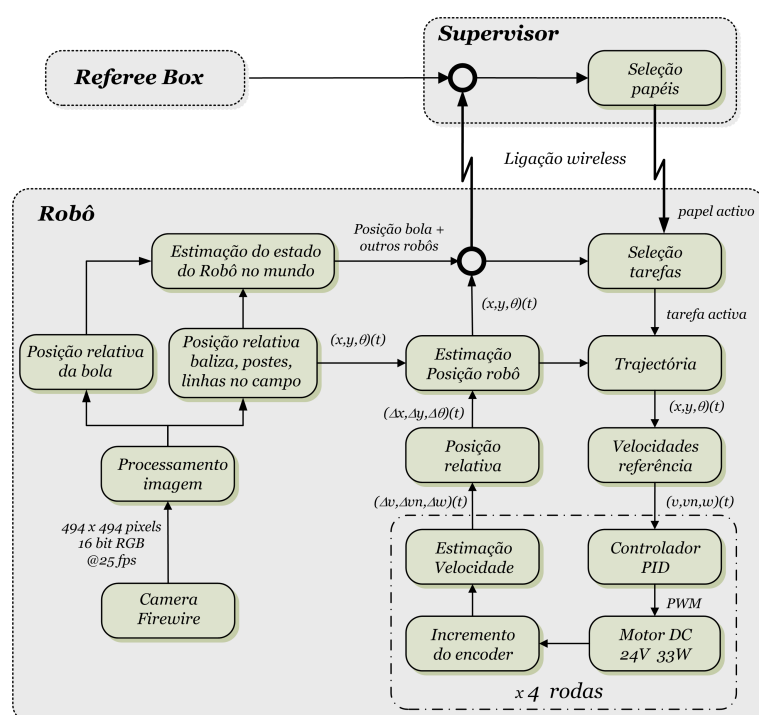


Figura 2.3: Estrutura de controlo e comunicação.

1. No nível 1 há o controlador para cada roda, essencial para o desempenho das funções de baixo nível. Esta aplicação está implementada num microcontrolador localizado na mesma placa da eletrônica de potência responsável pelo accionamento dos motores, sendo optimizado para a dinâmica do robô [Conceição et al. \(2005\)](#). O microcontrolador possui um controlador PID para cada roda do robô.
2. O nível 2 é composto por alguns módulos:
 - O módulo responsável pelo cálculo de velocidades de rotação e translação do robô, a partir de uma trajectória desejada. Assim como a conversão das velocidades do robô para as velocidades das rodas do robô, respeitando as limitações de velocidades dos motores([Conceição et al. \(2006a\)](#));
 - O módulo que utiliza as velocidades lidas pelos encoders para gerar os dados de odometria e actualizar a estimação da posição do robô;
 - O módulo de captura de imagem por um sistema de visão omnidireccional e a extração das características mais importantes da imagem: blocos de cor associados a bola, a baliza e outros robôs. Também são extraídas as transições verde/branco ou branco/verde para formar as linhas do campo. Estas informações são utilizadas para construir a estimação da posição relativa da bola, dos outros robôs e das balizas/postes. Reunindo essas informações com as linhas do campo, pode-se calcular uma estimativa da posição do robô, que é usada para actualizar o estado do robô no mundo, que será enviada para os outros robôs e para o supervisor.
3. No nível 3 as informações são tratadas de forma mais abstracta. Aqui assume-se que a posição do robô é conhecida com um determinado grau de confiança, e o objectivo principal é uma trajectória desejada com eventuais momentos de *kick*(chuto na bola). Existe um módulo responsável pelos papéis do robô(por exemplo: atacante, defensor ou central) que são atribuídos pelo supervisor. Em caso de problemas com o supervisor (pode acontecer devido a problemas com a rede *wireless*) o robô assume um papel

pré-definido. Definido o papel e o estado do jogo, uma máquina de estados selecciona a tarefa que será desempenhada pelo robô. Para cada tarefa, pode ser gerada uma trajectória usando diferentes conjuntos de acções. As acções são seleccionadas dependendo da percepção do estado do mundo: presença de obstáculos, grande confiança na posição estimada, se o robô está bloqueado por outros robôs, etc.

4. No nível 4 está o módulo do supervisor que recebe as informações através de *broadcast* dos outros robôs e os comandos da *referee box*. O supervisor cria uma estimativa do estado do mundo com base nas informações recolhidas dos diversos robôs, que inclui a posição dos robôs, seus papéis actuais e o estado do jogo determinado pela *referee box*. Outra informação extraída pelo supervisor é o estado do funcionamento dos robôs, que indica se algum robô apresenta problemas.

2.4 Conclusão

Neste Capítulo apresentou-se a equipa de robôs móveis futebolistas, projectada e construída na Faculdade de Engenharia da Universidade do Porto, para participar das competições de futebol robótico na liga de robôs médios dos eventos RoboCup. Descreveu-se as principais características de *hardware* e *software* dos robôs, assim como a estrutura de controlo e comunicação da equipa. É importante salientar que toda a investigação para o projecto e desenvolvimento da equipa vem sendo realizada por alunos (licenciatura(actuais e antigos), mestrado e doutoramento) e docentes da Faculdade de Engenharia, e que os conhecimentos adquiridos com esta iniciativa é aplicado com sucesso em outras aplicações de engenharia.

Capítulo 3

Modelização de um Robô Móvel Omnidireccional

3.1 Introdução

Neste Capítulo apresenta-se a modelização de um robô móvel omnidireccional. A modelização está dividida em duas partes. A Secção 3.2 abordará a estimação de um modelo linear para o processo que engloba o motor, a redução do motor e o encoder, para cada roda do robô. O objectivo principal é calibrar os controladores PID discretos implementados no microcontrolador do robô. Foram usadas técnicas baseadas em estimadores mínimos quadrados e sua variante variável instrumental para estimar a respectivas funções de transferência. Resultados da estimação dos processos e da sintonia dos controladores PID são apresentados.

Na Secção 3.3 apresentam-se o modelo cinemático e dinâmico do robô. Foram desenvolvidos três métodos para a identificação dos parâmetros das equações do modelo dinâmico do robô. Os parâmetros estimados são os atritos viscosos, atritos de coulomb e o momento de inércia do robô. Resultados de simulação e resultados reais com o robô são apresentados.

3.2 Modelo da Motorização do Robô Móvel

Os modelos lineares (1 para cada roda) que englobam os motores com a caixa redutora e o encoder, Figura 3.1, foram modelizados como um sistema dinâmico discreto com uma entrada(tensão) e uma saída(velocidade). Sendo assim, para representar a dinâmica do sistema foi considerado um modelo linear discreto. Devido as perturbações como por exemplo folgas e atritos da caixa redutora, sintetizar equações dinâmicas para o processo torna-se muito difícil e provavelmente ineficaz. Sendo assim, para a determinação dos coeficientes deste modelo foram utilizadas técnicas de estimação mínimos quadrados e sua variante variável instrumental. A identificação de um modelo para este processo possibilitará projectar e simular controladores de uma forma mais eficiente para tarefas de navegação do robô, como por exemplo os controladores PID discretos implementados no microcontrolador do robô. O modelo foi retirado com o motor em carga, ou seja, com as rodas pousadas e a suportarem o peso do robô.

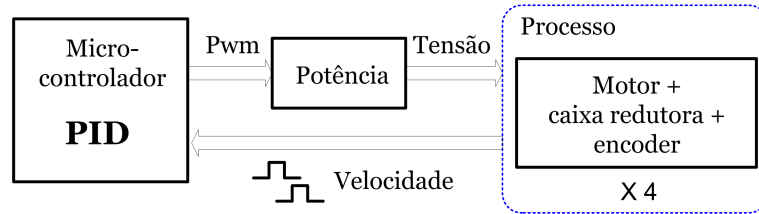


Figura 3.1: Processo da motorização.

3.2.1 Modelo Determinístico e Mínimos Quadrados

Em geral, sistemas dinâmicos lineares discretos invariantes no tempo podem ser representados por uma equação linear às diferenças de ordem n , relacionando a saída do sistema com as entradas e saídas dos instantes anteriores. Descrevendo o sistema por um modelo entrada-saída, conforme em [Carvalho \(1993\)](#) e [Vaccaro \(1995\)](#) tem-se:

$$y(k) = - \sum_{i=1}^{na} a_i y(k-i) + \sum_{i=0}^{nb} b_i u(k-i), \quad (3.1)$$

3.2 Modelo da Motorização do Robô Móvel

sendo $y(k)$ a sequência de saída, $u(k)$ a sequência de entrada e k a variável de tempo do sistema.

Aplicando a transformada Z à equação às diferenças 3.1 e supondo as condições iniciais nulas, obtém-se

$$(1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n})Y(z) = (b_o + b_1z^{-1} + \dots + b_nz^{-n})U(z), \quad (3.2)$$

multiplicando ambos os lados por z^n e reorganizando a equação, obtém-se a função de transferência de sistemas discretos no tempo representada por

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b_o z^n + b_1 z^{n-1} + \dots + b_n}{z^n + a_1 z^{n-1} + \dots + a_n}, \quad (3.3)$$

sendo o numerador e o denominador definidos respectivamente como

$$B(z) = b_o z^n + b_1 z^{n-1} + \dots + b_n, \quad (3.4)$$

$$A(z) = z^n + a_1 z^{n-1} + \dots + a_n. \quad (3.5)$$

Para a estimação do modelo do robô vamos considerar o modelo estocástico, conforme Goodwin & Payne (1997), da equação(3.6):

$$y(k) = H_1(z)u(k) + H_2(z)\xi(k), \quad (3.6)$$

onde $y(k)$ e $u(k)$ são as sequências de saída e entrada respectivamente, e $\xi(k)$ é a sequência de ruído branco com variância σ^2 . Considerando a situação onde H_1 e H_2 são parametrizados como $\frac{B(z)}{A(z)}$ e $\frac{1}{A(z)}$ respectivamente, conforme as equações (3.4) e (3.5), o modelo da equação (3.6) pode ser expresso como

$$A(q^{-1})y(k) = B(q^{-1})u(k) + \xi(k) \quad (3.7)$$

onde q^{-1} é o operador de atraso $q^{-1}y(k) = y(k-1)$, resultando:

$$\begin{aligned} y(k) &= -a_1 y(k-1) \dots - a_{na} y(k-na) + \dots \\ &\quad b_0 u(k) \dots + b_{nb} u(k-nb) + \xi(k) \\ &= x(k)^T \theta + \xi(k) \end{aligned} \quad (3.8)$$

onde

$$\theta^T = (a_1, \dots, a_{na}, b_0, \dots, b_{nb}) \quad (3.9)$$

$$\begin{aligned} x(k)^T &= (-y(k-1), \dots, -y(k-na) \dots \\ &\quad u(k), \dots, u(k-nb)). \end{aligned} \quad (3.10)$$

A equação 3.8 pode ser escrita na forma vectorial para N amostras como:

$$Y = X\theta + \Xi \quad (3.11)$$

onde

$$Y^T = [y(1), \dots, y(N)], \quad (3.12)$$

$$X^T = [x(1), \dots, x(N)], \quad (3.13)$$

$$\Xi^T = [\xi(1), \dots, \xi(N)]. \quad (3.14)$$

A equação (3.11) sugere a aplicação do estimador mínimos quadrados (Godwin & Payne (1997)), sendo o estimador resultante para θ ,

$$\hat{\theta} = [X^T X]^{-1} X^T Y. \quad (3.15)$$

3.2.2 Sinal de Excitação do Sistema

O sinal de excitação para uma função de transferência do tipo da equação 3.3 deve ter uma persistência de ordem $\geq n_a + n_b$, para que se possa revelar todas as características do modelo. Para uma estimação consistente do modelo, o sinal de excitação deve possuir algumas características probabilísticas:

- média nula, para a estimação não ser enviesada;
- ser um vector com variáveis aleatórias não correlacionadas e com variância idêntica.

O sinal de excitação utilizado foi um ruído branco gaussiano (WGN - White Gaussian Noise), gerado com a função matemática WGN do software Matlab (The MathWorks (1994-2006a)), o sinal gerado é apresentado na Figura 3.2. Verifica-se que o sinal possui um período de 50 amostras com valor constante e em seguida o inicia-se o ruído branco, este período inicial constante faz com que o processo fique no ponto de funcionamento antes da aplicação do sinal de excitação.

Para o algoritmo de estimação dos mínimos quadrados foi utilizado apenas as amostras a partir do instante 50. As tensões de alimentação dos motores DC

3.2 Modelo da Motorização do Robô Móvel

variam entre 0 (zero) e 24 volts, mapeados em uma palavra de controle de 8 bits (0-255). A quantidade de amostras (N) utilizada foi de 254 pontos, considerando que a taxa de amostragem do sistema (ts) é de $40ms$, teve-se um tempo de excitação com ruído de 10.16 segundos, para uma primeira experiência. Um outro sinal de excitação foi gerado da mesma forma que o citado acima, que será utilizado para a validação do modelo em uma outra experiência, descrita na secção 3.2.4.

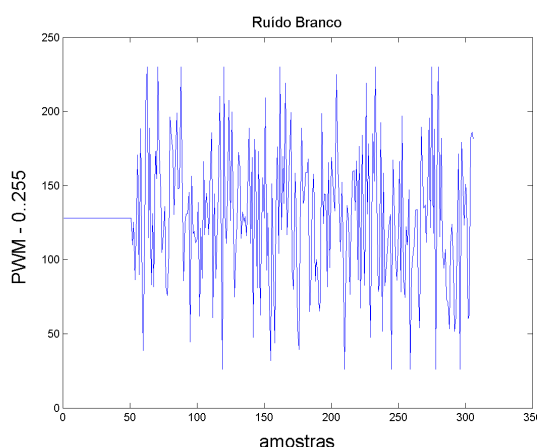


Figura 3.2: Ruído Branco Gaussiano.

3.2.3 Aplicação dos Mínimos Quadrados

Nas próximas secções será detalhada a estimação do processo para o motor 1 do robô. Em uma primeira etapa aplicou-se o sinal de excitação da Figura 3.2 no processo, obtendo-se a curva de velocidade do motor 1 em metros por segundo, conforme a Figura 3.3.

Verifica-se na Figura 3.3 o momento em que o ruído do sinal de excitação começa a actuar na velocidade do motor, a partir da amostra 50. Notam-se também perturbações e algum ruído de medida na etapa anterior à amostra 50, tal deve-se ao facto do sistema estar em malha aberta e qualquer irregularidade do piso ou rodas influenciam a velocidade.

Antes de aplicar os mínimos quadrados é necessário retirar o valor médio da entrada e da saída de modo a que uma entrada nula origine uma saída nula também.

3.2 Modelo da Motorização do Robô Móvel

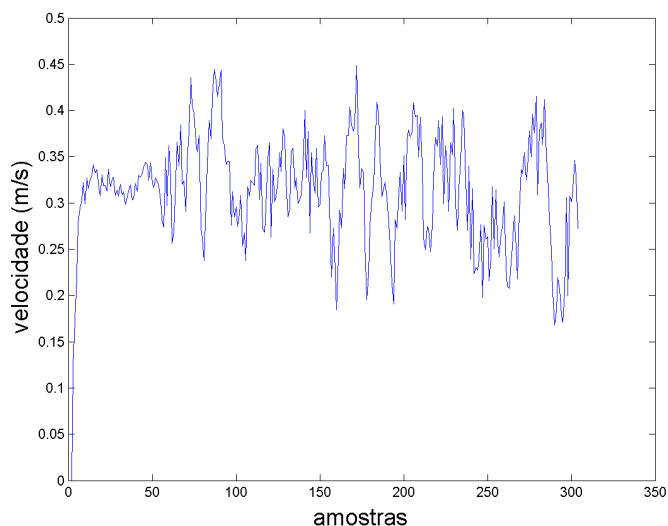


Figura 3.3: Curva de velocidade do motor 1.

De posse de todas as informações necessárias para efectuar a estimação dos mínimos quadrados, partiu-se para uma segunda etapa, com o objectivo estimar os coeficientes de funções de transferência que representem as características do processo. Foi realizada a estimação para 7 tipos de funções de transferência (FT), de grau 1 até 6. Na Tabela 3.1 apresenta-se o resultado da estimação para as diferentes funções de transferência, assim como os valores dos ganhos, pólos e zeros.

Na Figura 3.4 apresenta-se o sinal medido da velocidade do motor do robô e a velocidade estimada com a função de transferência do tipo FTb , conforme a Tabela 3.1.

3.2 Modelo da Motorização do Robô Móvel

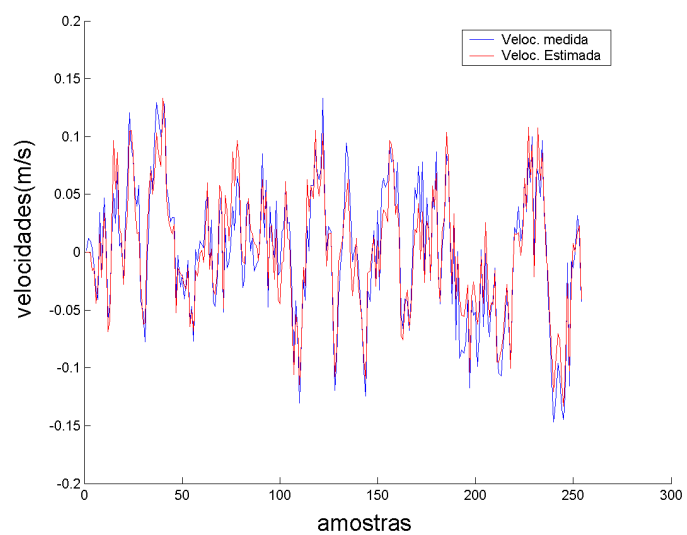


Figura 3.4: Velocidade medida e estimada para a *FTb* - Estimação.

3.2 Modelo da Motorização do Robô Móvel

FTs	FTa	FTb	FTc
grau	1	2	2
	$b1$ $z + a1$	$b2$ $z(z + a1)$	$b1z + b2$ $z(z + a1)$
ganho,zero, polo	$-5.73e - 6$ $(z - 0.7033)$	$8.16e - 4$ $z(z - 0.6827)$	$7.58e - 6(z + 107.7)$ $z(z - 0.6823)$
FTs	FTd		FTe
grau	3		4
	$b1z^2 + b2z + b3$ $z(z^2 + a1z + a2)$		$b1z^3 + b2z^2 + b3z + b4$ $z(z^3 + a1z^2 + a2z + a3)$
ganho,zero, polo	$6.25e - 6(z + 129.6)(z + 0.2702)$ $z(z - 0.7541)(z + 0.4024)$		$5.28e - 6(z + 153.5)(z^2 + 0.206z + 0.053)$ $z(z - 0.6946)(z + 0.457)(z - 0.183)$
FTs	FTf		
grau	5		
	$b1z^4 + b2z^3 + b3z^2 + b4z + b5$ $z(z^4 + a1z^3 + a2z^2 + a3z + a4)$		
ganho,zero, polo	$5.6e - 6(z + 143.8)(z - 0.1982)(z^2 + 0.4159z + 0.1107)$ $z(z - 0.6526)(z - 0.4272)(z^2 + 0.6709z + 0.1383)$		
FTs	FTg		
grau	6		
	$b1z^5 + b2z^4 + b3z^3 + b4z^2 + b5z + b6$ $z(z^5 + a1z^4 + a2z^3 + a3z^2 + a4z + a5)$		
ganho,zero, polo	$9.50e - 6(z + 84.98)(z^2 - 0.6704z + 0.2717)(z^2 + 0.8872z + 0.3613)$ $z(z - 0.6577)(z^2 - 0.728z + 0.2254)(z^2 + 0.9745z + 0.3502)$		

Tabela 3.1: Resultados da estimação para diferentes FTs

3.2.4 Validação do Modelo Dinâmico Estimado

Para a validação e escolha do modelo deve-se aplicar no processo e na função de transferência estimada um sinal de excitação diferente do que o sinal utilizado na estimação com o algoritmo dos mínimos quadrados. Após aplicar este novo sinal, calculou-se o MSE (Mean Square Error, [The MathWorks \(1994-2006b\)](#)) do erro entre as duas curvas de velocidade resultantes.

O MSE é definido como,

$$MSE() = \frac{\sum_{i=1}^N [(\hat{\Phi}(i) - \Phi(i))^2]}{N}, \quad (3.16)$$

que calcula a média do quadrado da diferença entre o conjunto de velocidades estimadas ($\hat{\Phi}$)

$$\hat{\Phi} = [\hat{y}(1), \dots, \hat{y}(N)]^T,$$

e o conjunto de velocidades medidas (Φ) dos motores,

$$\Phi = [y(1), \dots, y(N)]^T.$$

sendo N o número de amostras.

Sendo assim, na Tabela 3.2 e na Figura 3.5 apresentam-se os erros de validação para as diferentes funções de transferência. Na Figura 3.6 apresenta-se as curvas de velocidade medida e simulada para a função de transferência do tipo FTb .

Tipo FT	Grau	MSE(erro)
FTa	1	0.0042402
FTb	2	0.0010013
FTc	2	0.00099887
FTd	3	0.00099609
FTe	4	0.00099436
FTf	5	0.00099183
FTg	6	0.00099442

Tabela 3.2: Resultados da validação para as diferentes FTs.

A partir dos erros apresentados na validação, verifica-se que a função de transferência do tipo FTb , de grau 2, pode representar uma boa aproximação do processo. Esse resultado tem coerência, sendo que o sistema apresentado na figura

3.2 Modelo da Motorização do Robô Móvel

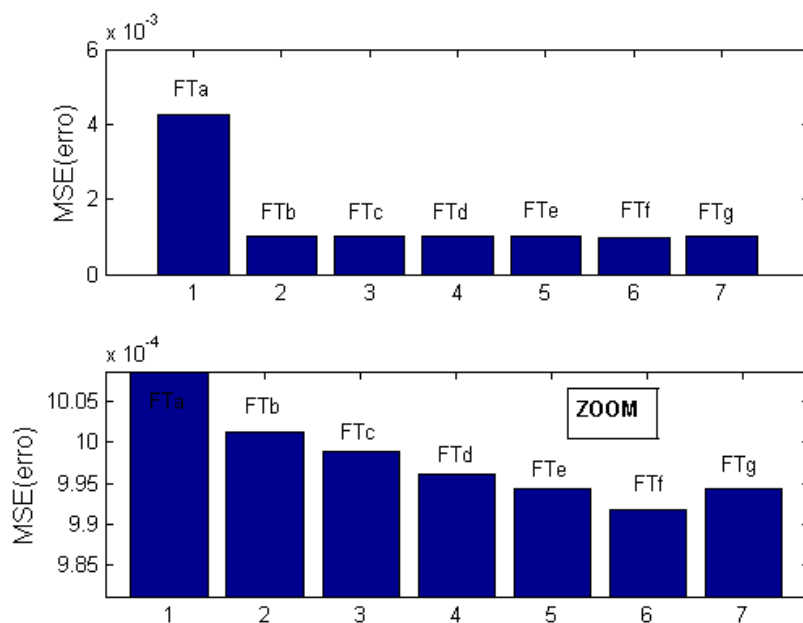


Figura 3.5: Resultados da validação para as diferentes FTs.

1 possui um atraso devido ao *loop* de comunicação, representado pelo pólo na origem. O processo com um motor DC pode ser aproximado por um sistema de primeira ordem, pois pode-se desprezar a indutância do motor devido a constante eléctrica ser muito pequena comparada com a constante mecânica do mesmo. A FT mais indicada seria a do tipo *FTf*, pois apresentou o menor erro, e a partir da *FTg* o erro começou a aumentar. Mas levando-se em conta que a *FTb* é um sistema mais simples, facilitando o cálculo de controladores em geral, e que a diferença do erro entre a *FTf* e a *FTb* é muito pequena, optou-se pela *FTb*. A *FTa* não apresentou um resultado satisfatório.

3.2 Modelo da Motorização do Robô Móvel

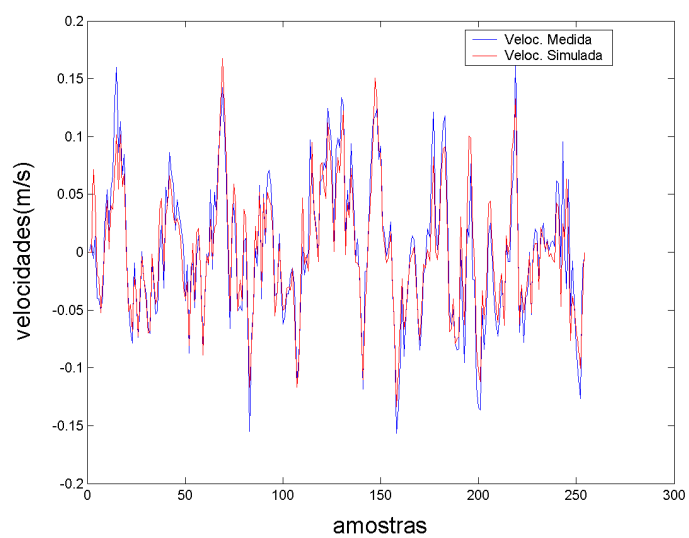


Figura 3.6: Velocidade medida e estimada para a *FTb* - Validação.

3.2.5 Estimador com Variável Instrumental

Para o modelo escolhido o estimador de mínimos quadrados pode não ser suficientemente consistente se o ruído de medida for significativo, também pode ser difícil testar a sua consistência para um funcionamento generalizado. Outros estimadores considerados consistentes, conforme [Goodwin & Payne \(1997\)](#), podem ser implementados para demonstrar mais propriedades do sistema e também ter um parâmetro para comparação com o estimador mínimos quadrados. Nesta Seção vão ser apresentados os resultados do estimador com variável instrumental, que comparando-se com os mínimos quadrados substitui-se $\hat{\theta}$ por $\bar{\theta}$, sendo

$$\bar{\theta} = [Z^T X]^{-1} Z^T Y$$

a matriz Z pode ser construída usando um modelo auxiliar

$$z_k^T = (-\hat{y}_{k-1}, -\hat{y}_{k-2}, \dots, -\hat{y}_{k-n}, u_k, u_{k-1}, \dots, u_{k-n})$$

onde

$$\hat{A}(z)\hat{y} = \hat{B}(z)u_k$$

e $\hat{A}$, $\hat{B}$ são polinômios em z que podem ser obtidos através de mínimos quadrados em uma primeira iteração e em seguida obter o estimador $\bar{\theta}$. Na iteração seguinte utiliza-se o resultado da estimativa anterior para recalcular Z_k e assim sucessivamente. Este algoritmo tem a característica de convergir com poucas iterações.

Na Tabela 3.3 são apresentados resultados do estimador variável instrumental(VI) e a comparação com mínimos quadrados para a função de transferência do tipo FTb . Com 3 iterações já houve uma convergência, estabilizando os valores do pólo e do ganho.

Verifica-se que o MSE do erro diminuiu levemente com cada de iteração, resultando em uma pequena mudança nos valores da FT. Através da auto correlação do erro calculado conforme (3.17) e (3.18), verifica-se a brancura dos resíduos. Na Figura 3.7 apresenta-se a auto correlação para a estimativa com os Mínimos Quadrados e a estimativa com Variáveis Instrumentais, para 100 amostras.

$$\hat{y}(k) = -\hat{a}1 * \bar{y}(k-1) + \hat{b}1 * u(k-2), \quad (3.17)$$

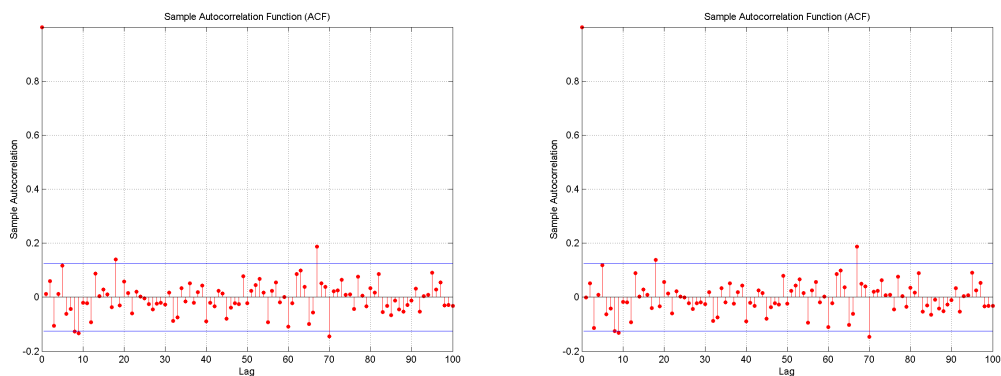
$$Erro_r(k) = \hat{y}(k) - \bar{y}(k). \quad (3.18)$$

3.2 Modelo da Motorização do Robô Móvel

FTs	Min. Quad.	VI iteração 1	VI iteração 2	VI iteração 3
$b1$	0.00081626	0.00081540	0.00081538	0.00081538
$z(z + a1)$	$z(z - 0.6827)$	$z(z - 0.7047)$	$z(z - 0.7051)$	$z(z - 0.7051)$
$MSE(erro)$	0.0010013	0.00099248	0.00098949	0.00098799

Tabela 3.3: Valores Estimados.

Sendo $\hat{y}$ a saída estimada, $\bar{y}$ a saída medida e u o sinal de entrada. A média do erro $Erro_r$ da equação (3.18) para Mínimos quadrados foi $5.2879e - 4$ e para variáveis instrumentais $5.2554e - 4$, verifica-se que houve uma pequena melhora. Verifica-se também que a auto correlação do erro $Erro_r$ para os mínimos quadrados já é muito parecido com um ruído branco, por este motivo não houve uma melhora muito significativa com Variáveis Instrumentais.



(a) Auto correlação do erro - Mínimos Quadrados (b) Auto correlação do erro - Variável Instrumental

Figura 3.7: Auto correlação do erro $Erro_r$ para estimação com Mínimos Quadrados e VI.

3.2.6 Resultados com os Outros Motores

Nas secções anteriores foi detalhado o procedimento para estimar o sistema do motor 1, agora serão apresentados resultados referentes aos outros 3 motores do robô. Na Tabela 3.4 estão os resultados estimados para os 4 motores do robô, utilizando o estimador Mínimos Quadrados e na Tabela 3.5 utilizando o estimador Variável Instrumental.

As rodas laterais(motores 2 e 4) possuem diâmetros maiores que das rodas dianteira(motor 1) e traseira(motor 3), conforme as Tabelas 3.4 e 3.5 os ganhos dos motores 2 e 4 evidenciam esta diferença. Também deve-se levar em conta que existe uma distribuição irregular do peso na base do robô. Essas características faz com que as funções de transferências tenham diferenças entre si. No motor 4 verifica-se um pólo mais lento, isso deve-se ao maior peso sobre esta roda, devido à posição das baterias na base do robô.

Na Figura 3.8 apresentam-se gráficos com sinais reais das velocidades dos motores e as velocidades estimadas com as funções de transferência do tipo FTb .

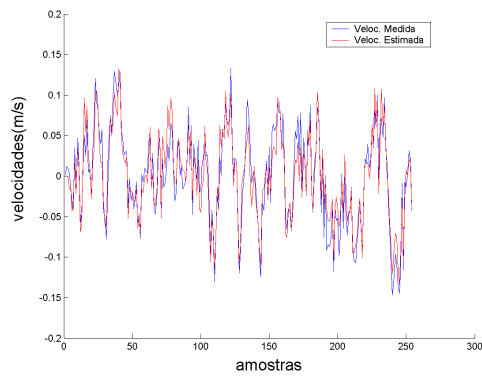
Mínimos Quadrados	Motor 1 Dianteira	Motor 2 Lateral esquerda	Motor 3 Traseira	Motor 4 Lateral direita
$b1$ $z(z + a1)$	0.0008163 $z(z-0.6827)$	0.00044521 $z(z-0.6425)$	0.00075587 $z(z-0.677)$	0.00037482 $z(z-0.7448)$

Tabela 3.4: Valores Estimados - Mínimos Quadrados.

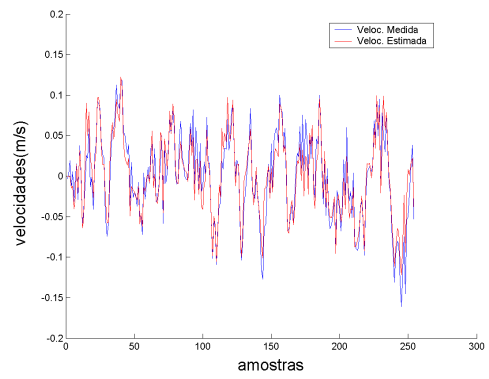
Variável Instrumental	Motor 1 Dianteira	Motor 2 Lateral esquerda	Motor 3 Traseira	Motor 4 Lateral direita
$b1$ $z(z + a1)$	0.00081538 $z(z-0.7051)$	0.00044513 $z(z-0.6734)$	0.00075507 $z(z-0.6953)$	0.0003748 $z(z-0.7565)$

Tabela 3.5: Valores Estimados - Variável Instrumental.

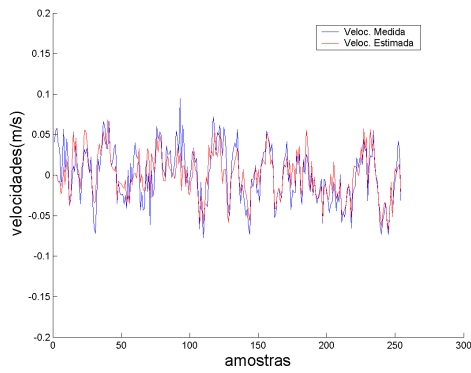
3.2 Modelo da Motorização do Robô Móvel



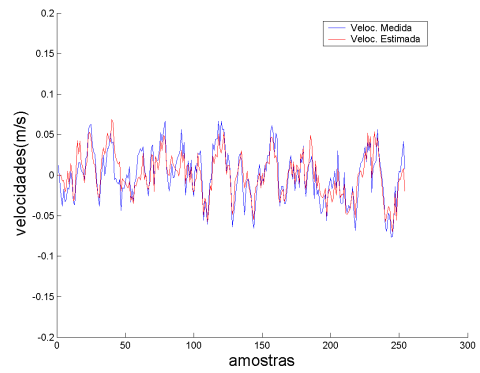
(a) Motor 1 - roda dianteira



(b) Motor 3 - roda traseira



(c) Motor 2 - roda lateral esquerda



(d) Motor 4 - roda lateral direita

Figura 3.8: Motores do robô.

3.2.7 Sintonia dos Controladores PID para os Motores

A partir da definição de um modelo para o processo, partiu-se para a sintonia inicial dos controladores PID implementados no microcontrolador do robô. Um controlador do tipo proporcional, integral e derivativo, geralmente denominado PID, pode ser descrito conforme [Astrom & Hagglund \(1995\)](#):

$$u(t) = K_c \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (3.19)$$

Considerando um sistema em malha fechada conforme a Figura 3.9. A variável $u(t)$ representa o sinal de controlo e a variável $e(t)$ representa o sinal de erro ($e(t) = y_{ref}(t) - y(t)$). O sinal de controlo é calculado pela soma de três termos: o termo P (no qual é proporcional ao erro), o termo I (no qual é proporcional ao integral do erro) e o termo D (no qual é proporcional à derivada do erro). Os parâmetros de controlo são o ganho proporcional K_c , o tempo integral T_i e o tempo derivativo T_d . Para a sintonia do controlador PID foi utilizado o procedimento de posicionar os pólos em malha fechada do sistema baseado no protótipo de Bessel, conforme em [Vaccaro \(1995\)](#).

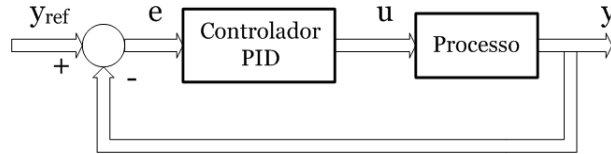


Figura 3.9: Sistema em malha fechada.

A seguir apresenta-se os cálculos dos parâmetros do PID para o motor 1 do robô, a função de transferência escolhida para o processo foi obtida através do estimador Variável Instrumental conforme a Secção 3.2.5,

$$G(z) = \frac{0.00081538}{(z - 0.7051)} = \frac{b1}{z - a1} \quad (3.20)$$

desprezou-se o pólo na origem da função de transferência, pois o mesmo deve-se ao atraso existente na comunicação entre o microcontrolador e o PC do robô. Este atraso apenas influencia a recolha de dados e como o controlador vai ficar implementado no microcontrolador não o devemos considerar.

3.2 Modelo da Motorização do Robô Móvel

Pode-se calcular um equivalente contínuo do processo da equação (3.20),

$$G(s) = \frac{K}{\tau s + 1},$$

usando a seguinte relação:

$$K = \frac{b1}{1 + a1},$$

$$\tau = \frac{1}{|ps|},$$

$$pz = e^{Tps}.$$

Sendo T o período de amostragem, ps o pólo no plano S , e pz o pólo no plano Z . Na equação (3.21) apresenta-se a função contínua resultante, e na Figura 3.10 uma resposta ao degrau dos dois processos, discreto e contínuo.

$$G(s) = \frac{0.002765}{0.1145s + 1} \quad (3.21)$$

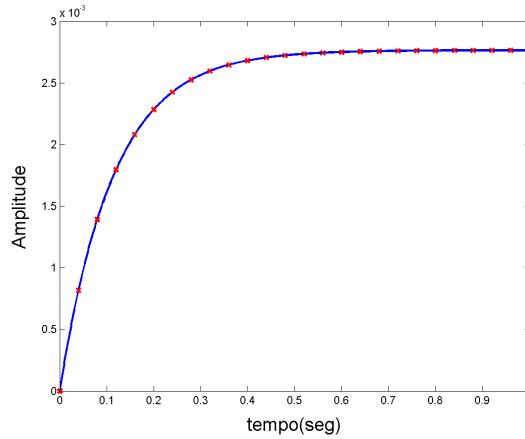


Figura 3.10: Resposta ao degrau do sistema discreto e contínuo.

A função de transferência do controlador, conforme a equação (3.22) representa a equação de um controlador PI. Escolheu-se um controlador PI, eliminando-se a parcela derivativa com $Td = 0$, devido às características do processo (primeira ordem). A acção integral Ti faz com que o sistema siga uma entrada de valor constante com erro nulo em regime permanente. A acção proporcional K_c tem um impacto imediato na rapidez de resposta do sistema.

3.2 Modelo da Motorização do Robô Móvel

$$G_c(s) = \frac{K_c(T_i s + 1)}{T_i s} \quad (3.22)$$

A partir da função de transferência do sistema em malha fechada, que resultou num sistema de segunda ordem, posicionou-se os pólos do controlador PI de forma a obter um tempo de estabelecimento em torno de 0.6seg para uma resposta ao degrau, que representa uma boa resposta levando-se em conta características de peso e tamanho do robô. O sistema em malha aberta G_{MA} está descrito na equação(3.23),

$$G_{MA}(s) = G_c(s).G(s) = \frac{K_c(T_i s + 1)}{T_i s} \cdot \frac{K}{\tau s + 1} \quad (3.23)$$

Calculando o sistema em malha fechada com realimentação unitária de $G_{MA}(s)$ obtem-se,

$$G_{MF}(s) = \frac{\frac{KK_c}{T_i\tau} + \frac{KK_cs}{\tau}}{s^2 + \frac{(1+KK_c)s}{\tau} + \frac{KK_c}{T_i\tau}} \quad (3.24)$$

Considerando a Tabela dos polinômios de Bessel (em Vaccaro (1995)), para um sistema de segunda ordem com tempo de estabelecimento de 1seg os pólos são $p_{1s} = -4.0530 \pm j2.3400$. Para o tempo de estabelecimento desejado (0.6seg), basta dividir p_{1s} por 0.6, resultando em pólos $p_{0.6s} = -6.7550 \pm j3.9000$. Para obter os valores de K_c e T_i , basta igualar o polinômio do denominador da $G_{MF}(s)$ com o polinômio da equação (3.25),

$$(s - p_1)(s - p_2) = s^2 - (p_1 + p_2)s + p_1p_2 \quad (3.25)$$

então, para obter K_c ,

$$\begin{aligned} -(p_1 + p_2) &= \frac{(1 + K * K_c)}{\tau} \\ K_c &= \frac{-\tau * (p_1 + p_2) - 1}{K} \end{aligned} \quad (3.26)$$

e para obter T_i ,

$$\begin{aligned} (p_1p_2) &= \frac{KK_c}{T_i\tau} \\ T_i &= \frac{KK_c}{\tau p_1p_2} \end{aligned} \quad (3.27)$$

3.2 Modelo da Motorização do Robô Móvel

A FT resultante é apresentada na equação(3.28)

$$G_{MF}(s) = \frac{4.775s + 0.3747}{s^2 + 13.51s + 60.84} \quad (3.28)$$

Na equação (3.29) apresenta-se a função do controlador PI para o motor 1 e na equação (3.30) o equivalente discreto invariante na resposta ao degrau(ZOH-zero-order hold)Franklin *et al.* (1997), para uma taxa de amostragem de 10ms.

$$G_c(s) = \frac{197.68(0.07848s + 1)}{0.07848s} \quad (3.29)$$

$$G_c(z) = \frac{197.68(z - 0.8726)}{(z - 1)} \quad (3.30)$$

Testou-se os parâmetros no controlador PI no robô, obtendo-se a resposta da Figura 3.12(a), verifica-se que com 0.6seg atingiu-se a referência, com uma resposta transitória satisfatória. Para melhorar esta resposta, introduziu-se no sistema em malha fechada um ganho *feedforward*(g_f), da entrada de referência até a entrada do processo, conforme a Figura 3.11. Este ganho não afecta a es-

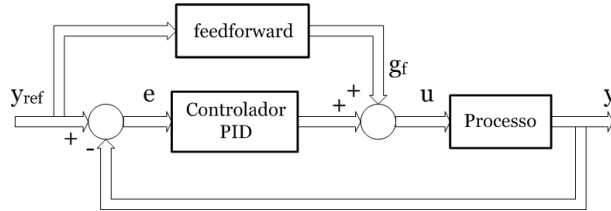
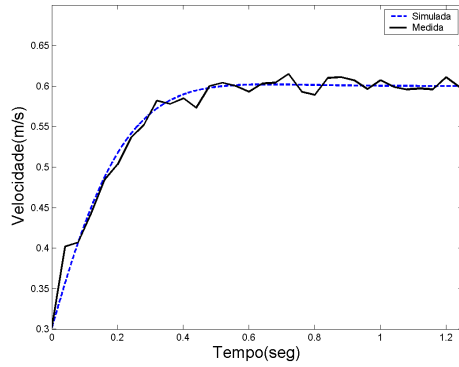


Figura 3.11: Sistema em malha fechada com *feedforward*.

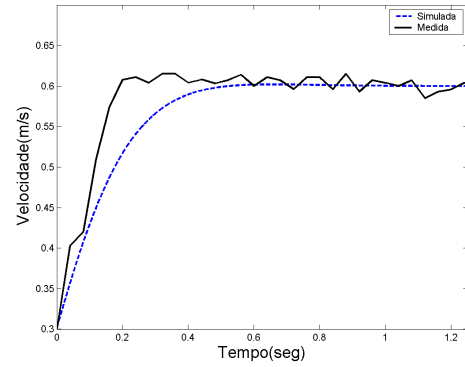
tabilidade do sistema, pois não altera os pólos em malha fechada. Entretanto, o ganho *feedforward* melhora a resposta transitória do sistema. Na Figura 3.12(b) verifica-se a melhoria na resposta transitória com $g_f = 200$, este valor foi obtido através de simulações. A entrada de referência foi um degrau de amplitude 0.6(m/s). Diferenças entre a simulação e as velocidades medidas são aceitáveis neste caso, tendo em conta a simplicidade do modelo escolhido(primeira ordem) e as aproximações cometidas.

Na Tabela 3.6 apresentam-se as funções dos controladores PI para os quatro motores das rodas do robô.

3.2 Modelo da Motorização do Robô Móvel



(a) Sistema sem *feedforward*.



(b) Sistema com *feedforward*.

Figura 3.12: Resultados com o controlador PID.

PI	Motor 1	Motor 2	Motor 3	Motor 4
Gc(z)	$\frac{197.68(z-0.8726)}{(z-1)}$	$\frac{269.0271(z-0.8321)}{(z-1)}$	$\frac{196.5306(z-0.8625)}{(z-1)}$	$\frac{608.4586(z-0.9069)}{(z-1)}$

Tabela 3.6: Controladores PI dos motores do robô.

3.3 Modelo do Robô Móvel

O modelo dinâmico e cinemático de robôs móveis é essencial para o projecto de controladores, principalmente quando os robôs executam trajectórias com velocidades elevadas ou trajectórias com mudanças bruscas de direcção e sentido, exigindo força máxima dos actuadores. Robôs omnidireccionais possuem a habilitade de simultaneamente e independentemente executar movimentos de translação e rotação, conforme detalhado em [Pin & Killough \(1994\)](#) e [Jun *et al.* \(1996\)](#). Entretanto, não linearidades, como por exemplo a saturação dos motores, podem comprometer o desempenho do robô, mais precisamente quando o robô está em aceleração.

Actualmente as estratégias de controlo para robôs omnidireccionais baseiam-se na dinâmica do robô para obter um desempenho melhor, como por exemplo em [Keigo *et al.* \(1998\)](#), [Albagul & Wahyudi \(2004\)](#), [Tai-Yu & Ching-Chih \(2004\)](#), [Sungyong & Minor \(2004\)](#), [Hamerlain *et al.* \(2005\)](#). Mas para se projectar um bom controlador é necessário derivar as equações dinâmicas e cinemáticas do robô e estimar seus parâmetros. [Canudas de Wit & Tsiotras \(1999\)](#) desenvolveu um modelo de força dos atritos para o contacto entre os pneus e o chão. O modelo baseou-se em um modelo dinâmico de atrito para pontos de contacto, chamado modelo de LuGre. Em [Williams *et al.* \(2002\)](#) foi desenvolvido um modelo dinâmico para robôs móveis omnidireccionais, incluindo as derrapagens das rodas, os coeficientes de atrito foram experimentalmente medidos, assim como as forças que causam as derrapagens. Em [Jae Heon *et al.* \(2003\)](#) apresentam-se o modelo dinâmico completo de um robô omnidireccional incluindo a dinâmica das rodas. [Golda *et al.* \(2004\)](#) e [Iagnemma *et al.* \(2004\)](#) desenvolveram métodos para obter características de terrenos irregulares, que são usados em algoritmos de controlo de tracção dos robôs. [Venture *et al.* \(2004\)](#) usou o modelo dinâmico inverso para identificar os parâmetros dinâmicos da suspensão e chassis de um veículo, através do método dos mínimos quadrados.

Objectivando um modelo completo para o robô, nesta Secção apresenta-se um modelo dinâmico e cinemático, onde estão modelados os elementos não-lineares e de restrições do robô. Desenvolveu-se três métodos para estimar os parâmetros das equações da dinâmica do robô. Os parâmetros são os coeficientes dos atritos

viscosos e de coulomb relacionados com as velocidades lineares e de rotação do robô, e o momento de inércia do robô. Por fim, serão apresentados resultados de simulação, bem como a comparação com dados reais do robô. Será feita uma comparação entre os resultados do modelo não-linear e uma versão linear.

3.3.1 Dinâmica do Robô Móvel

A dinâmica é o estudo do movimento na qual as forças aplicadas no robô são modelizadas, e a cinemática é o estudo do movimento sem considerar as forças que afectam o movimento, considerando apenas a geometria que governa o sistema [Dudek & Jenkin \(2000\)](#). O modelo do robô móvel omnidireccional foi desenvolvido com base na dinâmica e cinemática de toda a estrutura do robô, incluindo os controladores PID e motores DC das rodas do robô. Na Figura [3.13](#) apresenta-se a configuração do robô no sistema global de coordenadas em duas dimensões $SGC = \{O_G X_G Y_G\}$. Os seguintes símbolos em unidades SI (Sistema Internacional) são usados no modelo do robô:

- $b [m]$ → distância entre o ponto P e as rodas
- $M [kg]$ → massa do robô
- $J [kg.m^2]$ → momento de inércia do robô
- $r_1, r_2, r_3, r_4 [m]$ → raio das rodas do robô
- l_1, l_2, l_3, l_4 → redução dos motores do robô
- $v, v_n [m/s]$ → velocidades lineares no robô
- $w [rad/s]$ → velocidade angular no robô
- $\theta [rad]$ → posição angular do robô
- $F_v, F_{v_n} [N]$ → forças de tracção no robô nas direcções das velocidades v e v_n
- $\Gamma [N.m]$ → binário de rotação no robô na direcção da velocidade w
- $F_{Bv}, F_{Bv_n} [N]$ → atrito viscoso nas direcções das velocidades v e v_n
- $\Gamma_{Bw} [N.m]$ → atrito viscoso na direcção da velocidade w
- $F_{Cv}, F_{Cv_n} [N]$ → atrito de coulomb nas direcções das velocidades v e v_n
- $\Gamma_{Cw} [N.m]$ → atrito de coulomb na direcção da velocidade w
- $B_v, B_{v_n} [N/(m/s)]$ → coeficientes de atrito viscoso nas direcções das velocidades v e v_n
- $B_w [N.m/(rad/s)]$ → coeficiente de atrito viscoso na direcção da velocidade w
- $C_v, C_{v_n} [N]$ → coeficientes de atrito de coulomb nas direcções das velocidades v

e vn

- $C_w [N.m] \rightarrow$ coeficiente de atrito de coulomb na direcção da velocidade w
- $v_1, v_2, v_3, v_4 [m/s] \rightarrow$ velocidades lineares nas rodas
- $f_1, f_2, f_3, f_4 [N] \rightarrow$ força de tracção nas rodas
- $T_1, T_2, T_3, T_4 [N.m] \rightarrow$ binário nas rodas

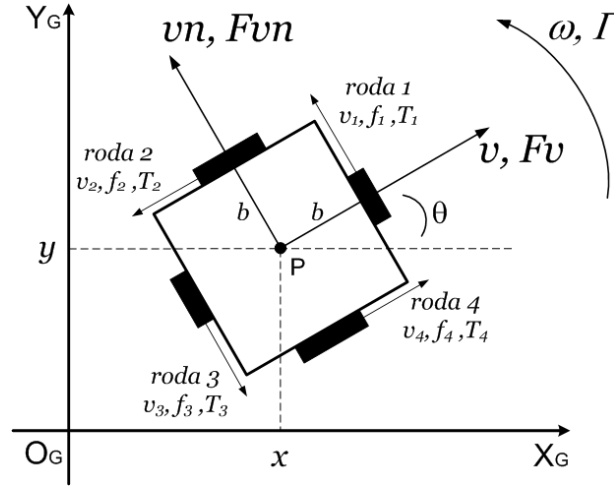


Figura 3.13: Configuração do robô e símbolos.

Segundo as coordenadas definidas na Figura 3.13 e pelas leis de Newton derivam-se as equações de translação e rotação do robô móvel,

$$F_v(t) - F_{Bv}(t) - F_{Cv}(t) = M \frac{dv(t)}{dt}, \quad (3.31)$$

$$F_{vn}(t) - F_{Bvn}(t) - F_{Cvn}(t) = M \frac{dvn(t)}{dt}, \quad (3.32)$$

$$\Gamma(t) - \Gamma_{Bw}(t) - \Gamma_{Cw}(t) = J \frac{dw(t)}{dt}. \quad (3.33)$$

Os atritos viscosos ($F_{Bv}, F_{Bvn}, \Gamma_{Bw}$) são forças com uma relação linear entre a força aplicada e a velocidade (Kuo (1995)),

$$F_{Bv}(t) = B_v v(t), \quad (3.34)$$

$$F_{Bvn}(t) = B_{vn} vn(t), \quad (3.35)$$

$$\Gamma_{Bw}(t) = B_w w(t). \quad (3.36)$$

Os atritos de coulomb ($F_{Cv}, F_{Cvn}, \Gamma_{Cw}$) são forças que tem uma amplitude constante com respeito à mudança da velocidade, mas o sinal da força de atrito muda

com a inversão do sentido da velocidade(Kuo (1995)),

$$F_{Cv}(t) = C_v \frac{v(t)}{|v(t)|}, \quad (3.37)$$

$$F_{Cvn}(t) = C_{vn} \frac{vn(t)}{|vn(t)|}, \quad (3.38)$$

$$\Gamma_{Cw}(t) = C_w \frac{w(t)}{|w(t)|}. \quad (3.39)$$

Na Figura 3.14 apresentam-se as relações entre atrito e velocidade.

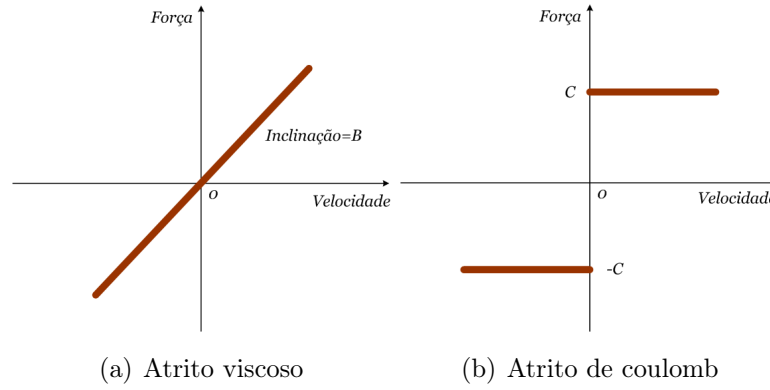


Figura 3.14: Atritos.

As relações entre as forças de tracção no robô e as forças de tracção nas rodas, descrevem-se como:

$$F_v(t) = f_4(t) - f_2(t) \quad (3.40)$$

$$F_{vn}(t) = f_1(t) - f_3(t) \quad (3.41)$$

$$\Gamma(t) = (f_1(t) + f_2(t) + f_3(t) + f_4(t))b \quad (3.42)$$

A força de tracção das rodas(f) e o binário das rodas(T), para cada motor DC, são representados por:

$$f(t) = \frac{T(t)}{r} \quad (3.43)$$

$$T(t) = l.K_t.i_a(t) \quad (3.44)$$

Sendo $i_a(t)$ a corrente do motor, K_t a constante do binário do motor e l a redução da caixa do motor. A corrente do motor é limitada para economizar bateria e para evitar que os transístores e/ou o motor se danifiquem,

$$0 \leq |i_a(t)| \leq i_{max}$$

onde i_{max} é um parâmetro.

A dinâmica para cada motor DC pode ser descrita usando as seguintes equações,

$$u(t) = L_a \frac{di_a(t)}{dt} + R_a i_a(t) + K_v w_m(t) \quad (3.45)$$

$$T(t) = K_t i_a(t) \quad (3.46)$$

Onde L_a é a indutância do motor, R_a é a resistência do motor, $E_a(t)$ é a força contra electromotriz (F_{emf}), $u(t)$ é a tensão aplicada ao motor, $w_m(t)$ é a velocidade angular do motor, k_v é a constante da força contra-electromotriz. A tensão $u(t)$ aplicada ao motor está limitada em:

$$-24 \leq u(t) \leq 24 \quad \text{para} \quad t \geq 0.$$

O binário de saída do motor é linearmente relacionado com a corrente do motor pela constante do binário do motor K_t , conforme a equação 3.46. A constante do binário do motor K_t e a constante da força contra-electromotriz K_v em unidades do sistema SI(Sistema Internacional) são idênticas, conforme em Kuo (1995): $K_t[N.m/A] = K_v[Volts/(rad/seg)]$.

3.3.2 Cinemática do Robô Móvel

A partir da geometria do robô é possível derivar as equações cinemáticas de movimento do robô, que relacionam as velocidades lineares das rodas do robô(v_1, v_2, v_3, v_4), as velocidades lineares (v e vn) e a velocidade angular(w) do robô,

$$\begin{aligned} \frac{dx(t)}{dt} &= v(t)\cos(\theta(t)) - vn(t)\sin(\theta(t)) \\ \frac{dy(t)}{dt} &= v(t)\sin(\theta(t)) + vn(t)\cos(\theta(t)) \\ \frac{d\theta(t)}{dt} &= w(t) \end{aligned} \quad (3.47)$$

$$\begin{aligned} v_1(t) &= vn(t) + bw(t) \\ v_2(t) &= -v(t) + bw(t) \\ v_3(t) &= -vn(t) + bw(t) \\ v_4(t) &= v(t) + bw(t) \end{aligned} \quad (3.48)$$

Sendo $x(t)$ e $y(t)$ a posição do ponto P , e $\theta(t)$ o ângulo de orientação do robô, conforme a Figura 3.13.

3.3.3 Identificação dos Parâmetros do Modelo do Robô

A estimação dos parâmetros relacionados com as equações da dinâmica do robô móvel foi realizada através de ensaios experimentais com o robô e o uso do método dos mínimos quadrados aplicado aos dados dos ensaios. Os parâmetros a serem identificados são os coeficientes dos atritos viscosos (B_v, B_{vn}, B_w) , os coeficientes dos atritos de coulomb (C_v, C_{vn}, C_w) e o momento de inércia J . A massa M do robô foi obtida com uma balança.

Desenvolveu-se três métodos para a estimação dos parâmetros, detalhados nas próximas sub-seções. Os coeficientes foram identificados com os métodos 1 e 2. O momento de inércia do robô foi identificado de duas formas: combinando os métodos 1 e 2, e com o método 3. Sendo assim, usou-se dois métodos para identificar os coeficientes dos atritos(1 e 2) e dois métodos para identificar o momento de inércia(1+2 e 3).

3.3.3.1 Método 1 - Velocidade em Regime Permanente

Neste método é possível identificar os coeficientes dos atritos viscosos (B_v, B_{vn}, B_w) e de coulomb (C_v, C_{vn}, C_w) do robô. A ideia é aplicar no robô velocidades v, vn e w com valores constantes, medindo-se a força de tracção F_v, F_{vn} e o binário $\Gamma(t)$ aplicado ao robô. Uma vez que a velocidade do robô entre em regime permanente de funcionamento, isto é, seja constante, pode-se nas equações 3.31, 3.32 e 3.33 desconsiderar a parcela associada à massa e ao momento de inércia do robô, resultando as seguintes equações diferenciais, para velocidades positivas:

$$F_v(t) = B_v v(t) + C_v \quad (3.49)$$

$$F_{vn}(t) = B_{vn} vn(t) + C_{vn} \quad (3.50)$$

$$\Gamma(t) = B_w w(t) + C_w \quad (3.51)$$

Sendo assim, realizou-se três ensaios independentes:

- 1º ensaio $\rightarrow$ aplicou-se 4 velocidades diferentes de v e calcularam-se as forças de tracção F_v para cada velocidade.
- 2º ensaio $\rightarrow$ aplicou-se 4 velocidades diferentes vn e calcularam-se a forças de tracção F_{vn} para cada velocidade.

3.3 Modelo do Robô Móvel

- 3º ensaio → aplicou-se 4 velocidades diferentes w e calcularam-se o binário Γ para cada velocidade.

As forças e binários resultantes, na Tabela 3.7, foram calculadas com base nas correntes medidas dos motores do robô, utilizando as equações 3.40 a 3.44. Apresentam-se na Figura 3.15 as velocidades e correntes obtidas para o ensaio com velocidades $v = vn = 1(m/s)$ e $w = 1(rad/s)$.

v (m/s)	F_v (N)	vn (m/s)	F_{vn} (N)	w (rad/s)	Γ (N.m)
0.6	30.585	0.6	30.285	0.6	5.556
0.8	31.459	0.8	30.814	0.8	5.730
1	31.874	1	31.174	1	5.842
1.2	32.765	1.2	32.106	1.2	6.009

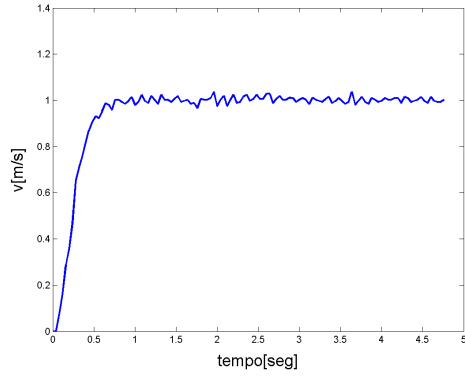
Tabela 3.7: Velocidades aplicadas e forças e binário resultante.

Com os valores das velocidades e das forças podem-se calcular os coeficientes. Verifica-se que as equações 3.49, 3.50 e 3.51 são idênticas a equação de um recta ($y = mx + b$). Utilizou-se os mínimos quadrados para ajustar uma recta aos resultados obtidos. A figura 3.16 mostra gráficos das velocidades aplicadas e as forças resultantes, assim como a recta de ajuste calculada. As equações resultantes com os atritos estimados são apresentados na Tabela 3.8.

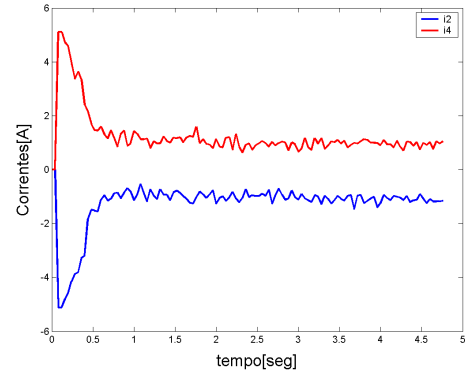
	Equações	B_v, B_{vn} B_w	C_v, C_{vn} C_w
v	$F_v(t) = 3.45v(t) + 28.55$	3.45	28.55
vn	$F_{vn}(t) = 2.90vn(t) + 28.46$	2.90	28.46
w	$\Gamma(t) = 0.73w(t) + 5.12$	0.73	5.12

Tabela 3.8: Equações e coeficientes - método 1.

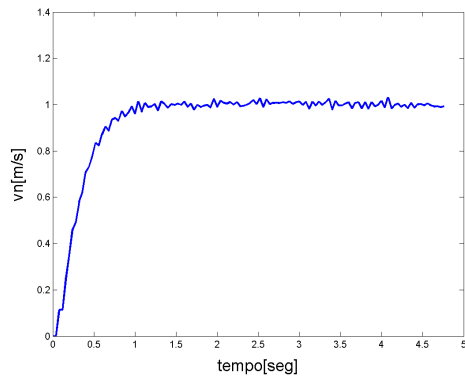
3.3 Modelo do Robô Móvel



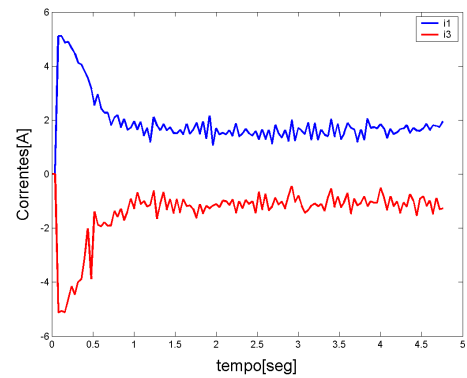
(a) $v = 1(m/s)$ - ensaio 1



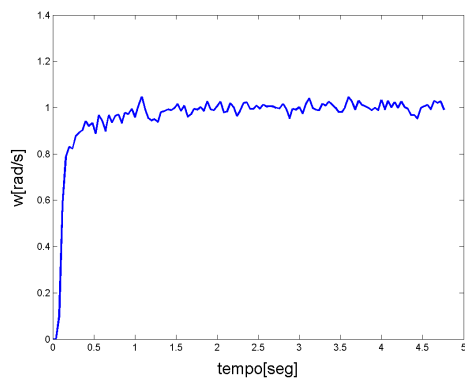
(b) motores 2(i_2) e 4(i_4) - ensaio 1



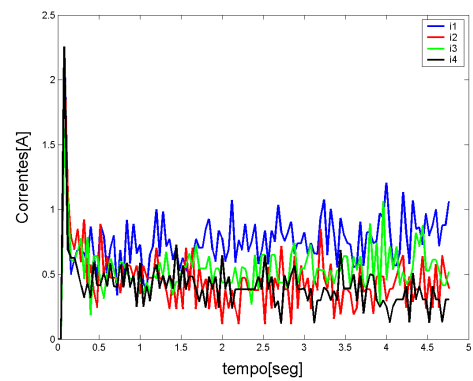
(c) $vn = 1(m/s)$ - ensaio 2



(d) motores 1(i_1) e 3(i_3) - ensaio 2

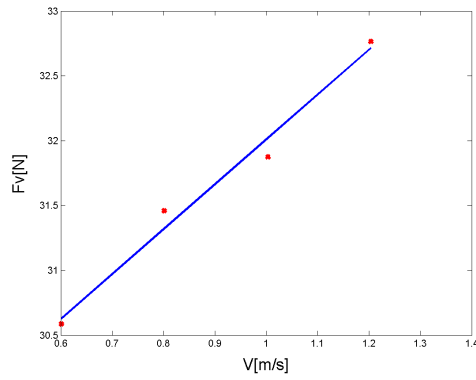


(e) $w = 1 (rad/s)$ - ensaio 3

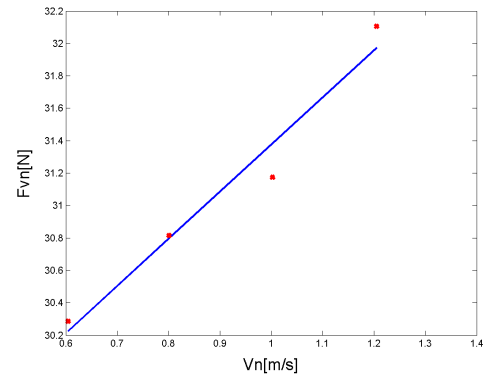


(f) motores 1 a 4(i_1, i_2, i_3, i_4) - ensaio 3

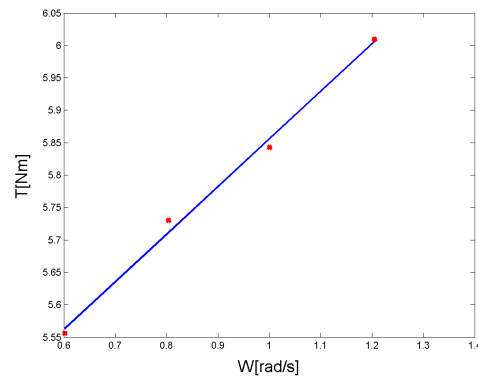
Figura 3.15: Velocidade e correntes para velocidade v - 1º ensaio.



(a) $F_v = f(v)$



(b) $F_{vn} = f(vn)$



(c) $\Gamma = f(w)$

Figura 3.16: Forças e binário *versus* velocidades.

3.3.3.2 Método 2 - Tracção Nula

No segundo método utilizou-se a variação de velocidade com forças nulas para calcular os coeficientes de atritos viscosos (B_v e B_{vn}) e de coulomb (C_v e C_{vn}). Primeiramente aplicou-se uma velocidade constante no robô, então desligou-se os motores, resultando em correntes nulas e consequentemente forças nulas, provocando um decréscimo na velocidade do robô até parar.

A partir das equações 3.31, 3.32 e 3.33 com forças de tracção (F_v e F_{vn}) e binário (Γ) nulos e reorganizando as equações, temos as seguintes equações diferenciais para velocidades positivas:

$$\frac{dv(t)}{dt} = -\frac{B_v}{M}v(t) - \frac{C_v}{M} \quad (3.52)$$

$$\frac{dvn(t)}{dt} = -\frac{B_{vn}}{M}vn(t) - \frac{C_{vn}}{M} \quad (3.53)$$

$$\frac{dw(t)}{dt} = -\frac{B_w}{J}w(t) - \frac{C_w}{J} \quad (3.54)$$

Foram realizados 3 ensaios separadamente, para v , vn e w :

- 1º ensaio → aplicou-se uma velocidade constante v no robô e então desligou-se os motores do robô. A partir da curva de velocidade v , calculou-se a aceleração do robô $\frac{dv(t)}{dt}$.
- 2º ensaio → aplicou-se uma velocidade constante vn no robô e então desligou-se os motores do robô. A partir da curva de velocidade vn , calculou-se a aceleração do robô $\frac{dvn(t)}{dt}$.
- 3º ensaio → aplicou-se uma velocidade constante w no robô e então desligou-se os motores do robô. A partir da curva de velocidade w , calculou-se a aceleração do robô $\frac{dw(t)}{dt}$.

Na Figura 3.17 apresentam-se os gráficos de velocidades dos ensaios realizados. Nos gráficos da figura 3.18 apresentam-se o conjunto de dados da velocidade e a aceleração calculada para realizar os cálculos de estimativa dos coeficientes dos atritos. A acelerações $\frac{dv(t)}{dt}$, $\frac{dvn(t)}{dt}$ e $\frac{dw(t)}{dt}$ foram calculadas a partir das velocidades v , vn e w , utilizando o seguinte método de derivação de Euler (em Franklin *et al.* (1997)):

$$\dot{x}(k) \cong \frac{x(k+1) - x(k)}{T} \quad (3.55)$$

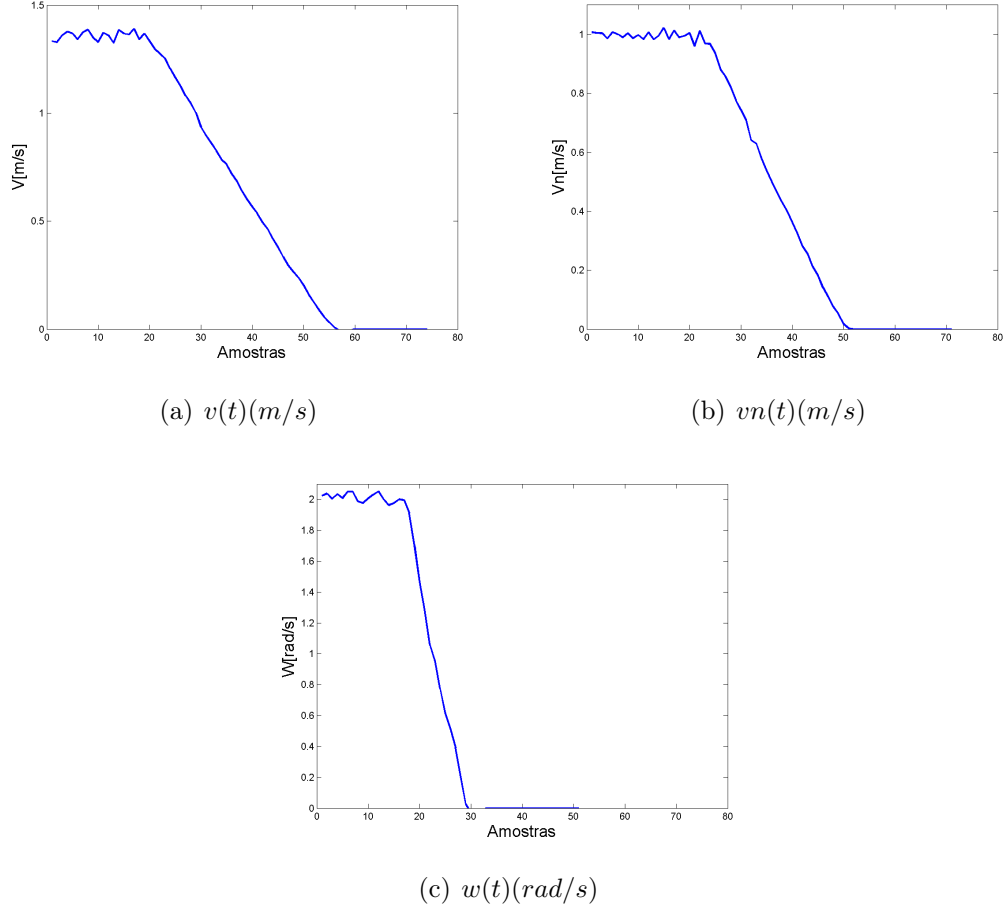


Figura 3.17: Curvas de velocidades.

onde T é o intervalo de amostragem em segundos(0.04seg no maior caso).

Com este método é necessário ter-se o valor da massa M e momento de inércia J do robô para se calcular o coeficiente de atrito viscoso e de coulomb. Como à priori não se tem o valor do momento de inércia J , obteve-se primeiramente uma estimativa de J com o auxílio dos valores dos coeficientes B_w e C_w calculados com o método 1, utilizando-se a equação 3.54 e a curva de velocidade $w(t)$ apresentada na Figura 3.17(c). Para o cálculo do momento de inércia J , pode-se reorganizar a equação 3.54 de forma a isolar J ,

$$J = -\frac{w(t)}{\frac{dw(t)}{dt}}B_w - \frac{C_w}{\frac{dw(t)}{dt}} \quad (3.56)$$

Com os vectores de valores de w e $\frac{dw(t)}{dt}$ (Figura 3.18(c)) e com os coeficientes

3.3 Modelo do Robô Móvel

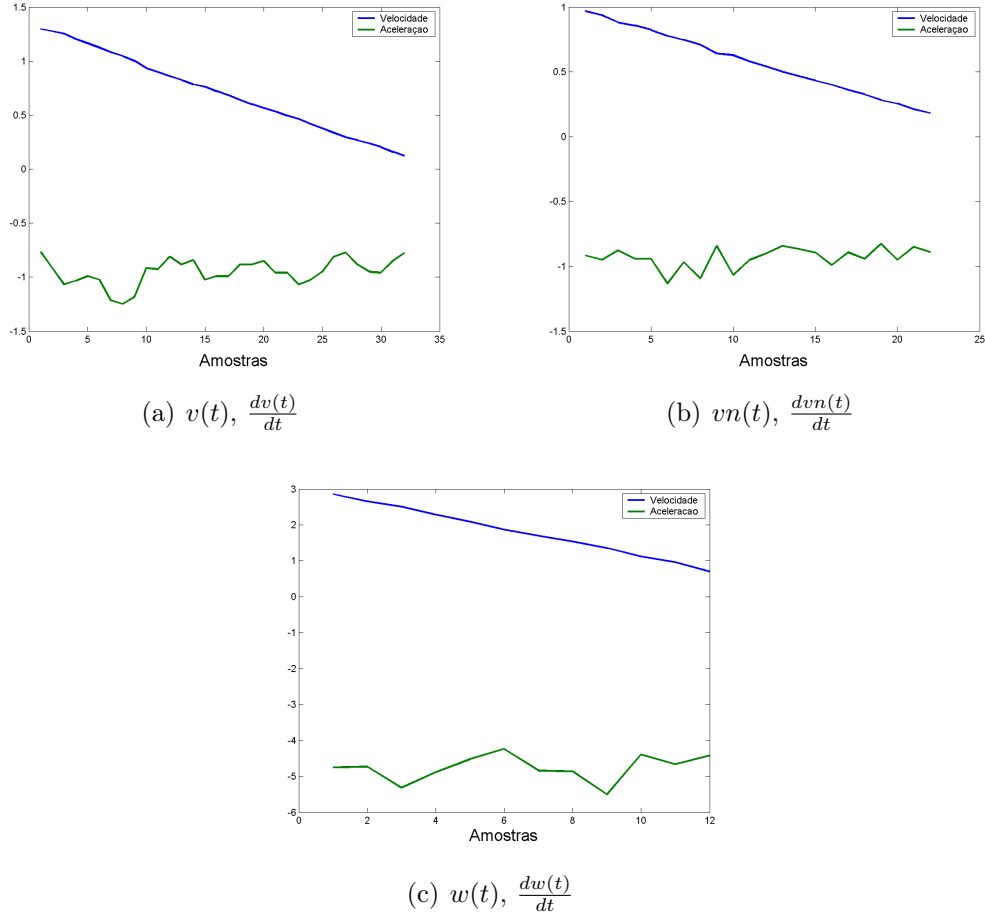


Figura 3.18: Velocidades e acelerações, usadas na estimação.

dos atritos viscoso B_w e de coulomb C_w , calculou-se valores de J , e fazendo a sua média obteve-se o valor de $J = 1.358[kg.m^2]$.

Com os valores de massa do robô e curvas de velocidades e acelerações dos ensaios (Figuras 3.18(a) e 3.18(b)), calculou-se os coeficientes dos atritos viscosos (B_v e B_{vn}) e de coulomb (C_v e C_{vn}), com a aplicação dos mínimos quadrados nas equações 3.52 e 3.53. As equações resultantes com os coeficientes estimados para v e vn são apresentados na Tabela 3.9.

	Equações	$B_v,$ B_{vn}	$C_v,$ C_{vn}
v	$\frac{dv(t)}{dt} = -0.11v(t) - 0.86$	4.01	30.35
vn	$\frac{dvn(t)}{dt} = -0.10vn(t) - 0.87$	3.77	30.76

Tabela 3.9: Equações e coeficientes - método 2.

3.3.3.3 Método 3 - Momento de Inércia

A identificação do momento de inércia J do robô foi realizada de forma experimental. O robô foi suspenso por um fio, para eliminar qualquer atrito do robô com o chão, conforme a Figura 3.19. Aplicou-se um binário ($\Gamma(t)$) conhecido ao corpo do robô, medindo-se a posição angular ($\theta(t)$) através de um sistema de odometria externa, baseado em visão, fixado acima do robô. Foi fixado ao robô um disco de acrílico, por onde passa um fio preso em uma extremidade ao próprio disco e na outra a um objecto com massa mo . O disco de acrílico possui um raio(rm) de $0.25[m]$. Com base na equação que rege o movimento de rotação do

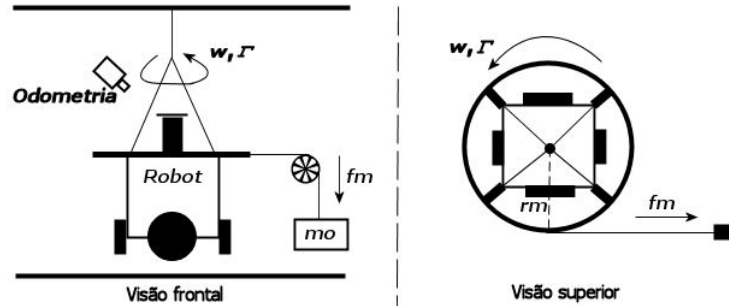


Figura 3.19: Esquemático da experiência.

robô,

$$\Gamma(t) - \Gamma_{Bw}(t) - \Gamma_{Cw}(t) = J \frac{dw(t)}{dt}. \quad (3.57)$$

é possível calcular o valor de J , com os valores dos coeficientes de atritos(B_w e B_{cw}) nulos, obtêm-se:

$$\Gamma(t) = J \frac{dW(t)}{dt} \quad (3.58)$$

Para esta experiência o binário é definido como:

$$\Gamma(t) = f_m(t).rm \quad (3.59)$$

onde,

$$f_m(t) = m \cdot g \quad (3.60)$$

sendo $f_m(t)$ a força aplicada e g a aceleração da gravidade ($\approx 9.8[m/s^2]$). Foram realizados 4 ensaios para identificar o valor de J . Nos ensaios 1, 2 e 3 utilizou-se um objecto(mo) com massa igual a 0.510 Kg . No ensaio 4 utilizou-se um objecto com massa igual a 1 Kg . Sendo assim, validou-se a repetibilidade da experiência com binários iguais e diferentes.

As Figuras 3.20(a), 3.20(c), 3.20(e) e 3.20(g) apresentam as curvas das posições angulares, obtidas pelo sistema de odometria externo. A velocidade angular (w), Figuras 3.20(b), 3.20(d), 3.20(f) e 3.20(h), foram calculadas a partir da derivada da curva das posições angulares. Através do método de mínimos quadrados, calculou-se uma recta que melhor se ajusta na curva de velocidade angular, e partir desta recta retira-se a aceleração angular para cada ensaio. Na Tabela 3.10 apresenta-se os valores de binário, aceleração angular e os valores obtidos do momento de inércia.

	Ensaio 1	Ensaio 2	Ensaio 3	Ensaio 4
Γ	1.2495	1.2495	1.2495	2.4500
$\frac{dW(t)}{dt}$	0.9126	0.8942	0.8947	1.7512
J	1.3691	1.3974	1.3965	1.3990

Tabela 3.10: Momentos de Inércia.

3.3 Modelo do Robô Móvel

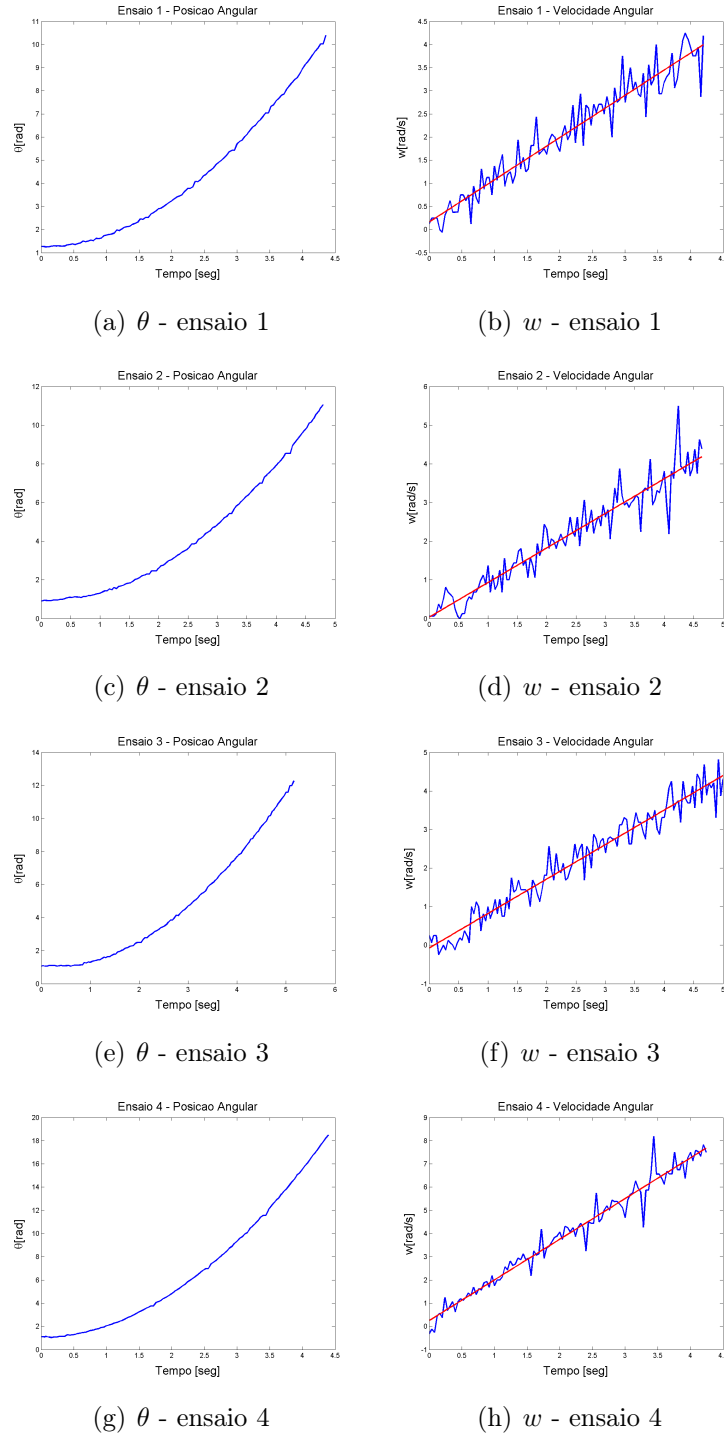


Figura 3.20: Gráficos dos Ensaio.

3.3.3.4 Comparando Métodos

Na Tabela 3.11 apresentam-se os coeficientes dos atritos estimados com os dois métodos, verifica-se que houve uma pequena diferença entre os valores, considerada aceitável, levando-se em conta que os métodos utilizados foram realizados em condições experimentais difíceis. Por exemplo os valores de corrente e velocidades lidos dos motores possuem um ruído considerável, qualquer irregularidade no chão onde foram executados os ensaios podem provocar uma pequena alteração nos parâmetros estimados. Sendo assim serão utilizados no modelo do robô a média dos parâmetros estimados. Na identificação do momento de inércia do robô, Tabela 3.12, verifica-se que se obteve valores similares com os dois métodos.

Atritos	Viscoso			Coulomb		
	B_v	B_{vn}	B_w	C_v	C_{vn}	C_w
Método 1	3.45	2.90	0.73	28.55	28.46	5.12
Método 2	4.01	3.77	-	30.35	30.76	-
Média	3.73	3.34	0.73	29.45	29.61	5.12

Tabela 3.11: Coeficientes de atritos estimados.

	Momentos de inércia
Métodos 1+2	1.358
Métodos 3 (média)	1.390
Média	1.374

Tabela 3.12: Momentos de inércia estimados.

3.3.4 Identificação dos Parâmetros dos Motores

Para a identificação do modelo eléctrico do motor DC, usou-se as seguintes equações diferenciais, conforme descrito na Secção 3.3.1:

$$u(t) = L_a \frac{di_a(t)}{dt} + R_a i_a(t) + K_v w_m(t) \quad (3.61)$$

$$T(t) = K_t i_a(t) \quad (3.62)$$

Com o motor DC em estado de regime permanente, ($L_a \frac{di_a}{dt} = 0$), pode-se simplificar a equação 3.61 como,

$$u(t) = R_a i_a(t) + K_v w_m(t) \quad (3.63)$$

Reorganizando a equação 3.63, temos

$$i_a(t) = \frac{u(t) - K_v w_m(t)}{R_a}. \quad (3.64)$$

Através de um procedimento experimental onde se aplicou um conjunto de tensões ao motor e depois de estabilizar a velocidade, mediu-se a corrente do motor i_a e a velocidade angular do motor w_m em regime permanente, foi possível estimar o valor das constantes K_v e R_a . A equação 3.64 assume o formato de uma equação da recta quando dividida por u , conforme:

$$\begin{aligned} \frac{i_a(t)}{u(t)} &= \frac{1}{R_a} - \frac{K_v w_m(t)}{R_a u(t)}, \\ \frac{w_m(t) K_v}{R_a u(t)} + \frac{i_a(t)}{u(t)} &= \frac{1}{R_a}, \\ \frac{w_m(t) K_v + R_a i_a(t)}{R_a u(t)} &= \frac{1}{R_a}, \\ \frac{R_a u(t)}{w_m(t) K_v + R_a i_a(t)} &= R_a, \\ u(t) &= w_m(t) K_v + R_a i_a(t), \end{aligned}$$

finalizando,

$$\frac{u(t)}{i_a(t)} = K_v \frac{w_m(t)}{i_a(t)} + R_a. \quad (3.65)$$

Usou-se um ajuste de rectas por mínimos quadrados para calcular as constantes K_v e R_a . Na Figura 3.21 apresenta-se $\frac{w_m(t)}{i_a(t)}$ em função de $\frac{u(t)}{i_a(t)}$, e a

3.3 Modelo do Robô Móvel

recta que melhor se adapta aos pontos mensurados. A equação resultante foi: $y = 0.1170x + 3.8299$, sendo a constante $K_v = 0.1170[\text{Volts}/(\text{rad}/\text{seg})]$ e a resistência $R_a = 3.8299[\Omega]$. Estes parâmetros são do motor da roda 1 do robô. O mesmo procedimento foi utilizado para a identificação dos parâmetros dos outros motores do robô.

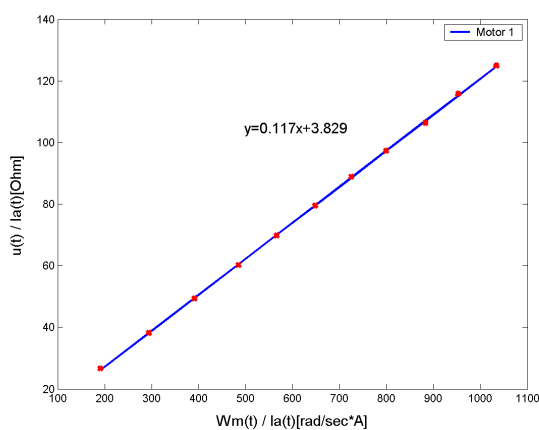


Figura 3.21: Recta que determina os valores de K_v e R_a - motor da roda 1.

A indutância do motor L_a geralmente é desprezada devido a constante eléctrica ser muito pequena comparada com a constante mecânica do motor.

3.3.5 Parâmetros do Modelo do Robô

Os parâmetros estimados para o modelo do robô são apresentados na Tabela 3.13.

Raio das rodas[m]					Redução dos motores			
$b[m]$	r_1	r_2	r_3	r_4	l_1	l_2	l_3	l_4
0.215	0.065	0.098	0.069	0.098	6.75	12.25	6.75	12.25

Resistências[Ω]				Femf[Volts/(rad/seg)]			
R_{a1}	R_{a2}	R_{a3}	R_{a4}	K_{v1}	K_{v2}	K_{v3}	K_{v4}
3.8	4.3	4.3	4.1	0.11	0.10	0.11	0.11

J	M	B_v	B_{vn}	B_w	C_v	C_{vn}	C_w
[kg.m ²]	[kg]	[N/(m/s)]	[N/(m/s)]	[N.m/(rad/s)]	[N]	[N]	[N.m]
1.3	35	3.73	3.34	0.73	29.45	29.61	5.12

Tabela 3.13: Parâmetros.

3.3.6 Resultados Reais e de Simulação

Nesta Secção serão apresentadas simulações do modelo completo do robô, utilizando os parâmetros estimados nas secções anteriores. Utilizou-se o software Matlab/Simulink para implementar e realizar as simulações com o modelo.

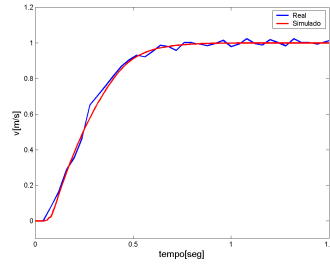
Foram realizadas 3 simulações, usando as seguintes velocidades de referência para o robô:

- Simulação 1 $\hookrightarrow v = 1[m/s]$, $vn = 0[m/s]$, $w = 0[rad/s]$;
- Simulação 2 $\hookrightarrow v = 0[m/s]$, $vn = 1[m/s]$, $w = 0[rad/s]$;
- Simulação 3 $\hookrightarrow v = 0[m/s]$, $vn = 0[m/s]$, $w = 1[rad/s]$.

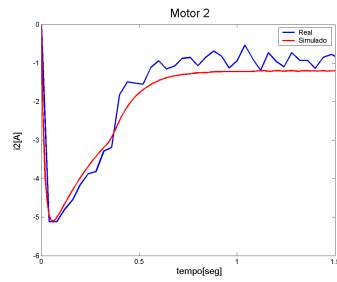
As Figuras 3.22(a), 3.23(a) e 3.24(a) apresentam as velocidades reais e de simulação com o robô. Os resultados de simulação são muito similares aos reais quer em regime transitório quer em regime permanente. Nas Figuras 3.22(b), 3.22(c), 3.23(b) e 3.23(c) pode-se verificar algumas não-linearidades do robô, devido a saturação dos sinais de PWM, mostrados nas Figuras 3.22(d), 3.22(e), 3.23(d) e 3.23(e). As correntes simuladas e reais tem uma boa aproximação, porque as não-linearidades estão modeladas. Na simulação 3, Figura 3.24, os motores não saturam facilmente, comparado às outras simulações, isso deve-se ao facto que são os 4 motores exercendo forças simultaneamente. Tal facto faz com que o regime transitório seja mais rápido.

Foi implementada uma versão linear do modelo, para obter algumas comparações de desempenho. A versão linear obviamente é mais simples, onde os elementos de saturação da tensão e limitação das correntes não estão modelados. Na Figura 3.26 apresentam-se os resultados de simulação com a versão não-linear e linear dos modelos. As velocidades de referência para o robô foram: $v = 1[m/s]$, $vn = 0[m/s]$ e $w = 0[rad/s]$. Analisando os resultados, pode-se verificar que a versão linear do modelo não representou correctamente o comportamento do robô, e por outro lado que a modelização dos elementos não-lineares foram essenciais para o bom desempenho do modelo completo não-linear do robô.

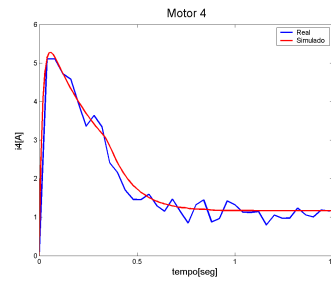
3.3 Modelo do Robô Móvel



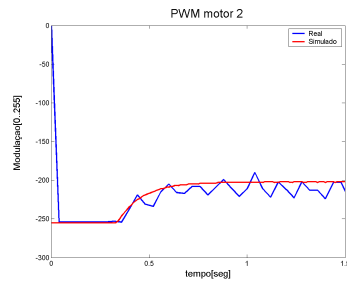
(a) Velocidade $v(m/s)$ real e simulada.



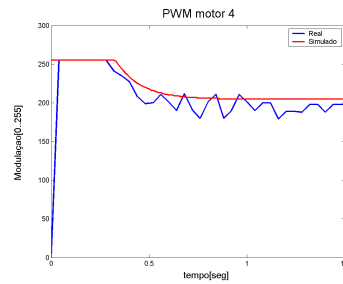
(b) Corrente $i_2(A)$



(c) Corrente $i_4(A)$



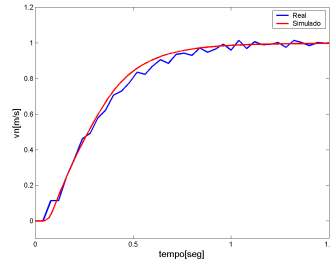
(d) PWM motor 2



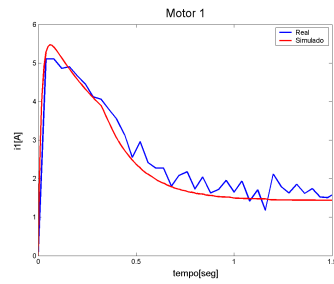
(e) PWM motor 4

Figura 3.22: Velocidade $v = 1$, simulação 1.

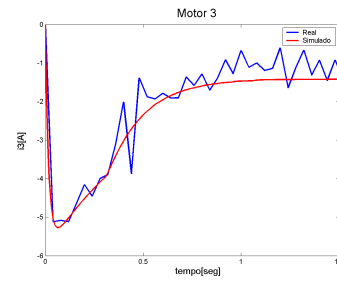
3.3 Modelo do Robô Móvel



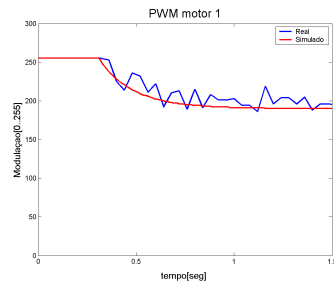
(a) Velocidade $vn(m/s)$ real e simulada.



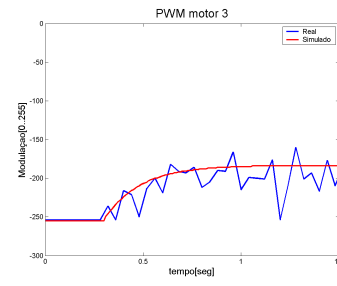
(b) Corrente $i1(A)$



(c) Corrente $i3(A)$



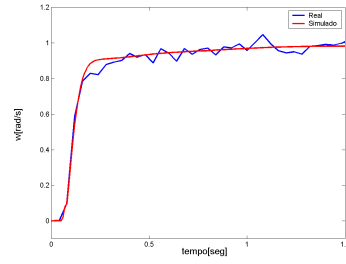
(d) PWM motor 1



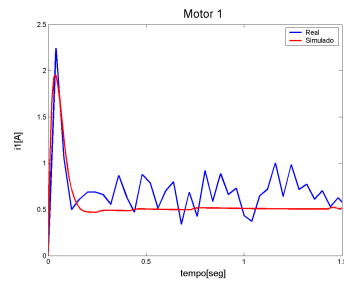
(e) PWM motor 3

Figura 3.23: Velocidade $vn = 1$, simulação 2.

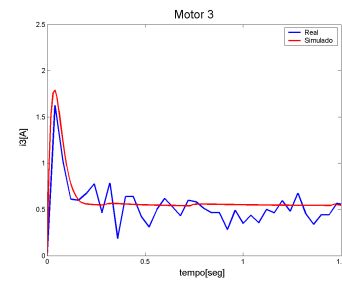
3.3 Modelo do Robô Móvel



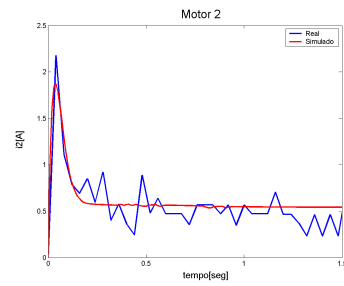
(a) Velocidade $w(rad/s)$ real e simulada.



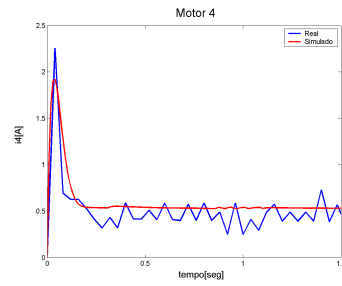
(b) Corrente $i1(A)$



(c) Corrente $i3(A)$



(d) Corrente $i2(A)$



(e) Corrente $i4(A)$

Figura 3.24: Velocidade $w = 1$, simulação 3.

3.3 Modelo do Robô Móvel

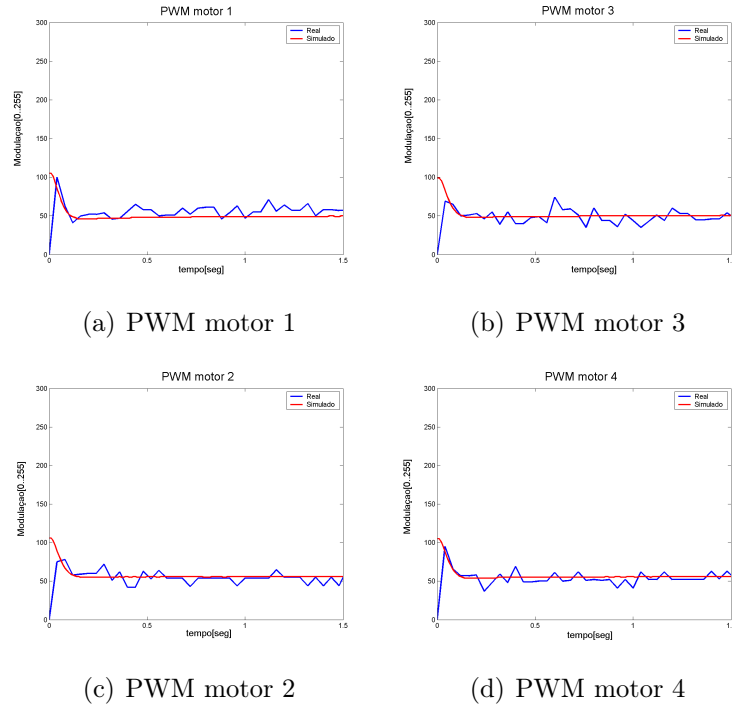


Figura 3.25: Velocidade $w = 1$, simulação 3.

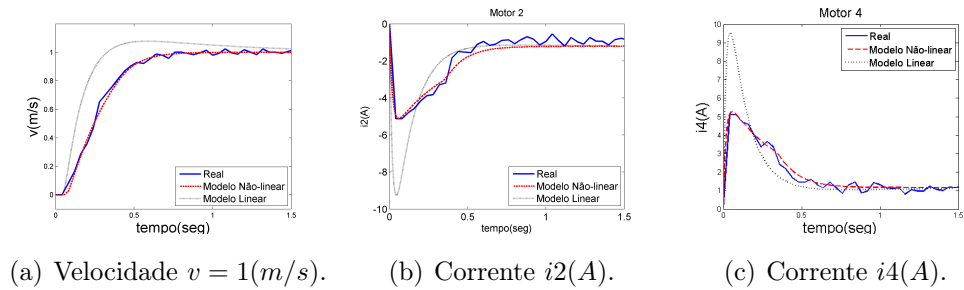


Figura 3.26: Comparação entre a modelização não-linear e a linear.

3.4 Conclusão

Nesta Secção desenvolveu-se a modelização completa de um robô móvel omnidireccional. Inicialmente estimou-se um modelo linear para o processo que engloba o motor, a redução do motor e o encoder, para cada roda do robô. O objectivo principal foi calibrar os controladores PID discretos implementados no microcontrolador do robô. Usou-se técnicas baseadas em estimadores mínimos quadrados e sua variante variável instrumental. Com os controladores PID sintonizados para os quatro motores do robô, partiu-se para um modelo dinâmico e cinemático para o robô. Três métodos para estimar os parâmetros das equações da dinâmica do robô foram apresentados. O modelo estimado consegue representar não-linearidades, como saturação dos motores, limitações de correntes e acção dos atritos nas velocidades do robô. Resultados reais e de simulação foram apresentados. É importante salientar que todos os robôs da equipa são rigorosamente similares, logo os resultados obtidos com um robô podem ser utilizados em toda a equipa. Mas de facto, a maior contribuição são as técnicas e algoritmos utilizados na modelização que podem ser replicados em qualquer estrutura de robô móvel com rodas.

Capítulo 4

Controladores para Robôs Móveis Omnidireccionais

4.1 Introdução

Este Capítulo apresenta estratégias de controlo para robôs omnidireccionais no seguimento de trajectórias. Foram desenvolvidas e testadas duas estratégias de controlo, com estruturas e características diferentes:

- Controlador reactivo $\leftrightarrow$ com uma estrutura baseada em realimentação parcial do estado do robot, acção esta mais relacionada com reacção. Este controlador foi projectado com base em alguns requisitos:
 - aplicável em robôs omnidireccionais;
 - respeitar restrições dinâmicas do robô;
 - analisar e sintetizar as estratégias de controlo utilizadas recentemente em um método flexível com baixo tempo computacional.
- Controlador preditivo $\leftrightarrow$ com a sua estrutura baseada em antecipação ou predição. O projecto baseou-se em alguns itens:
 - características não-lineares do sistema a controlar;
 - possuir um modelo não-linear do sistema;
 - conhecer as referências futuras *a priori* (trajectórias a seguir);

- usar uma nova abordagem para resolver o problema de minimização da função custo do controlador preditivo.

Nas secções 4.2 e 4.3 apresentam-se os controladores. Resultados de simulação e resultados reais com o robô no seguimento de trajectórias também são apresentados, além das conclusões obtidas com o projecto dos respectivos controladores.

4.2 Controlador Reactivo

Nos últimos anos foram abordadas inúmeras estratégias para resolver o problema de controlo de robôs móveis omnidireccionais, que abrange o problema de planeamento, geração e seguimento de trajectórias. Entre os mais relevantes podem-se citar [Muir & Neuman \(1987\)](#) que introduziu uma metodologia para modelar a cinemática de um robô omnidireccional e um algoritmo de controlo por realimentação. [Keigo et al. \(1998\)](#) desenvolveu um controlador com realimentação PI e PD, baseado na acelerações do robô. Foi usado ainda um termo *feed-forward* para minimizar o erro entre a trajectória desejada e a percorrida. [Liu et al. \(2003\)](#) projectou um controlador não-linear chamado *trajectory linearization control* (TLC), baseado na linearização ao longo da trajectória desejada e no modelo dinâmico. [Bianco et al. \(2004\)](#) apresenta uma solução para um controlador contínuo(*smooth control*), baseado em dois conceitos: planeamento cartesiano de trajectos e a geração de comandos para as rodas baseada na inversão do modelo cinemático do robô. [Kalmar-Nagy et al. \(2004\)](#) desenvolveu um algoritmo de controlo e geração de trajectórias que calcula o trajecto com menor tempo baseado na dinâmica do robô e nas características do motor. [Li-Chun et al. \(2006\)](#) aborda o controlo em tempo óptimo, é proposto em um procedimento iterativo para transformar o problema de tempo-óptimo em um problema de programação não-linear, o objecto de optimização é o período de amostragem.

Dando maior ênfase na geração e planeamento de trajectórias pode-se citar [Pignon et al. \(1992\)](#) que introduziu a estruturação de espaços baseado na aplicabilidade de diferentes métodos de planeamento, objectivando melhorar a eficiência do planeamento de trajectórias para robôs móveis. O estruturação do espaço é baseada no crescimento de círculos relativos ao robô. [Munoz et al. \(1994\)](#) usou uma sequência de curvas *splines* para gerar um trajecto. As *splines* continham informação do tempo, determinando a velocidade de navegação desejada. [Soueres & Laumond \(1996\)](#) apresenta uma caracterização completa de pequenos trajectos para robôs. É construída uma partição de espaços em regiões onde o mesmo tipo de trajecto é óptimo para encontrar a origem. [Tanaka et al. \(1998\)](#) propôs um método de geração de trajectórias para controlo dinâmico dos robôs. O método abordado foi o campo potencial artificial. Um algoritmo de geração de trajectórias para um veículo *off-road* é apresentado em [Moore & Flann \(2000\)](#),

usando o algoritmo A^* para determinar os trajectos e um controlo linearizante por realimentação para o seguimento das trajectórias. He *et al.* (2004) propôs uma simplificação das equações de movimento de um robô omnidireccional com quatro rodas. Com o sistema simplificado, calculou-se trajectórias com tempo quasi-óptimo e grande eficiência computacional. Em Purwin & D'Andrea (2005), para calcular trajectórias com tempo mínimo usou-se algoritmos de controlo óptimo com relaxações para diminuir o esforço computacional. Esse método baseou-se nos atritos e na dinâmica do robô para melhorar a performance do controlador.

Com base nessa literatura projectou-se uma nova estratégia de controlo com baixa carga computacional. A estratégia envolve a definição das trajectórias de referência e um controlador baseado em realimentação parcial do estado do robô (x, y, θ) . Além disso, foi desenvolvido um algoritmo que analisa os sinais de controlo aplicados no robô, baseado nas restrições dinâmicas dos motores do robô.

4.2.1 Definição do Controlador

Para os robôs omnidireccionais, onde não existem restrições nas direcções de movimentação, a possibilidade de definir diferentes velocidades lineares e angular para o robot durante o seguimento de uma trajectória é uma ferramenta importante. Uma trajectória pode ser aproximada por segmentos de recta, que por sua vez são definidos por 2 pontos do espaço. Com a utilização de segmentos de recta para definir trajectórias é possível definir as velocidades lineares e a posição angular para o robot em cada ponto do segmento e ajustar estas velocidades ao longo do segmento, tendo em conta os valores correspondentes ao ponto inicial e ao ponto final do segmento.

O controlador apresentado nesta secção, possui uma realimentação parcial do estado do robô (x, y, θ) , pois corrige a posição (x, y) e a orientação do robô (θ) . Na Figura 4.1 ilustra-se as representações dos vectores de velocidades do controlador (v_a, v_d, v_c) , posição do robô (P) e um segmento de recta $(\overline{AB})$, pertencente a trajectória de referência R .

A trajectória de referência R é definidas por posições, orientações e velocidades lineares para o robô no sistema global de coordenadas em duas dimensões $SGC = \{O_G X_G Y_G\}$: $R(i) = [x_r(i) \ y_r(i) \ \theta_r(i) \ v_r(i)]^T$, $i = 0, 1, \dots, M$, onde M é o

número total de pontos da trajectória. A posição do robô é representada pelo

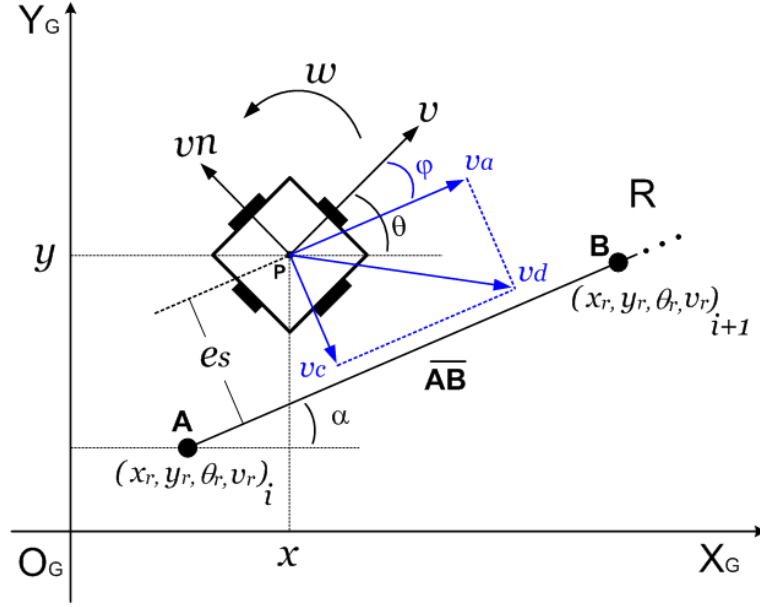


Figura 4.1: Representação esquemática do controlador e do robô móvel.

ponto $P(x, y)$ e o ângulo de orientação θ . Os vectores v e vn , perpendiculares entre si, representam as velocidades lineares do robô e w a velocidade angular do robô.

O ângulo φ é a diferença entre o ângulo α do segmento $\overline{AB}$ e o ângulo θ do robô no sistema de coordenadas SGC:

$$\varphi = \alpha - \theta. \quad (4.1)$$

O vector de velocidade v_d é a velocidade linear desejada para o robô no ponto P . A velocidade v_d pode ter valores diferentes a cada um dos pontos dos extremos dos segmentos de recta, possibilitando que o robô siga a trajectória com diferentes referências de velocidade. A norma da velocidade de referência ao longo do segmento de recta é dada pela seguinte expressão:

$$v_d = v_r(i)(1 - d) + v_r(i + 1)d. \quad (4.2)$$

Sendo $v_r(i)$ e $v_r(i + 1)$ as velocidades lineares de referência nos pontos A e B do segmento de recta, respectivamente. A variável d é a projecção do posição

do robô(x, y) no segmento AB , normalizada para o comprimento do segmento de recta. Os vectores de velocidade do controlador são definidos pelas seguintes equações:

$$v_c = e_s k_1, \quad (4.3)$$

$$v_a = \begin{cases} 0 & , \quad v_d^2 - v_c^2 < 0, \\ \sqrt{v_d^2 - v_c^2} & , \quad v_d^2 - v_c^2 > 0. \end{cases} \quad (4.4)$$

Sendo o vector v_a paralelo a $\overline{AB}$ ($v_a \parallel \overline{AB}$) e o vector v_c perpendicular a $\overline{AB}$ ($v_c \perp \overline{AB}$). A distância entre o robô e o segmento $\overline{AB}$ é dada por e_s e k_1 é um ganho.

Com os valores dos vectores v_c , v_a e o ângulo φ calcula-se as velocidades lineares de referência para o robô v e vn , através de uma matriz de transformação de coordenadas:

$$\begin{bmatrix} v \\ vn \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} v_a \\ v_c \end{bmatrix}.$$

A velocidade angular w de referência do robô é calculada com base na posição angular do robô(θ) e a posição angular de referência (θ_r) em cada ponto do segmento de recta. Ao longo desse segmento de recta a velocidade angular w é ajustada segundo uma variação linear θ_d entre o valor correspondente ao ponto inicial $\theta_r(i)$ e o valor correspondente ao ponto final $\theta_r(i+1)$ da posição angular pretendida no segmento de recta. O controlador da posição angular do robô fica definido pela seguinte equação:

$$w = e_\theta k_2, \quad (4.5)$$

em que:

$$e_\theta = \theta_d - \theta, \quad (4.6)$$

$$\theta_d = \theta_r(i)(1-d) + \theta_r(i+1)d. \quad (4.7)$$

onde k_2 é um ganho, e_θ é o erro entre a posição angular do robô(θ) e a posição angular pretendida(θ_d) e d é a projecção do posição do robô(x, y) no segmento AB , normalizada para o comprimento do segmento de recta.

4.2.2 Restrição nos Sinais de Controlo

Da forma como o controlador está definido até o momento, as velocidades de referência podem exigir velocidades que os motores do robô não consigam atender. Com um mapeamento preciso das limitações dos motores e a partir do modelo cinemático do robô, é possível mapear as limitações de velocidades nas direcções de movimentação do robô. O objectivo é redimensionar as velocidades de referência, se necessário, de forma a não saturar os motores e manter a direcção correcta dos vectores de velocidades de referência (v , vn e w), para não provocar erros no seguimento das trajectórias.

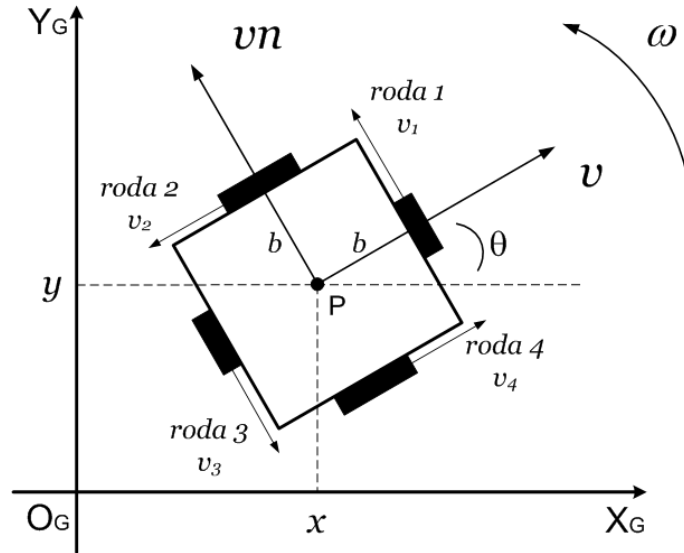


Figura 4.2: Representação do robô móvel.

As velocidades de referência lineares (v , vn) e angular (w) são convertidas para as velocidades lineares das rodas v_1 , v_2 , v_3 e v_4 , conforme as equações :

$$v_1 = vn + bw \quad (4.8)$$

$$v_2 = -v + bw \quad (4.9)$$

$$v_3 = -vn + bw \quad (4.10)$$

$$v_4 = v + bw \quad (4.11)$$

onde b é a distância entre o ponto $P(x, y)$ e as rodas do robô. A partir da Figura 4.2, é possível retirar as equações de movimento do robô levando em

conta as velocidades lineares das rodas do robô e as velocidades lineares (v , vn) e angular(w) do robô,

$$v = \frac{1}{2}(-v_2 + v_4) \quad (4.12)$$

$$vn = \frac{1}{2}(v_1 - v_3) \quad (4.13)$$

$$w = \frac{1}{2b}(v_1 + v_3) \quad (4.14)$$

$$w = \frac{1}{2b}(v_2 + v_4) \quad (4.15)$$

Foi desenvolvido um algoritmo para mapear as restrições de velocidades dos motores, cuja formulação consiste em calcular as velocidades de referência v, vn e w máximas levando em conta as velocidades máximas dos motores do robô. Essas equações podem ser escritas com três variáveis de controle (v, vn e w), relacionadas com as velocidades lineares das rodas (v_1, v_2, v_3 e v_4). Sendo assim, para uma determinada velocidade w , temos para a roda 1, a partir da equação 4.14:

$$v_3 = 2bw - v_1 \quad (4.16)$$

Substituindo 4.16 em 4.13,

$$\begin{aligned} vn &= \frac{1}{2}(v_1 - 2bw + v_1) \\ vn &= v_1 - bw \\ v_1 &\in [-v_{1max}, v_{1max}] \end{aligned} \quad (4.17)$$

onde para $v_1 = v_{1max}$ e $w = 0$ teremos o limite máximo positivo de vn . E para $v_1 = -v_{1max}$ e $w = 0$ teremos o limite máximo negativo de vn .

Identicamente tendo em conta as limitações dos motores restantes:

$$\begin{aligned} v &= -v_2 + bw \\ v_2 &\in [-v_{2max}, v_{2max}] \end{aligned} \quad (4.18)$$

$$\begin{aligned} vn &= -v_3 + bw \\ v_3 &\in [-v_{3max}, v_{3max}] \end{aligned} \quad (4.19)$$

$$\begin{aligned} v &= v_4 - bw \\ v_4 &\in [-v_{4max}, v_{4max}] \end{aligned} \quad (4.20)$$

As equações (4.17...4.20) podem ser analisadas como equações de um plano Anton & Rorres (2000),

$$c_1X + c_2Y + c_3Z + c_4 = 0 \quad (4.21)$$

Substituindo o sistema de coordenadas X, Y, Z por v, vn, w respectivamente, e o coeficiente c_4 por v_i , temos:

$$c_1v + c_2vn + c_3w + v_i = 0 \quad (4.22)$$

com v_i sendo as velocidades lineares das rodas do robô, para $i = 1, 2, 3, 4$. Reorganizando as equações, temos as equações de oito planos, considerando w positivo ou negativo,

$$vn + bw - v_1 = 0 \quad (4.23)$$

$$v - bw + v_2 = 0 \quad (4.24)$$

$$vn - bw + v_3 = 0 \quad (4.25)$$

$$v + bw - v_4 = 0 \quad (4.26)$$

$$vn - bw - v_1 = 0 \quad (4.27)$$

$$v + bw + v_2 = 0 \quad (4.28)$$

$$vn + bw + v_3 = 0 \quad (4.29)$$

$$v - bw - v_4 = 0 \quad (4.30)$$

A Figura 4.3 mostra os oito planos com os seguintes limites de velocidades para as rodas do robô: $v_1 = v_3 = 1(m/s)$ e $v_2 = v_4 = 1.5(m/s)$. Os limites de velocidade para v_1 e v_3 são menores pois o raio das rodas são menores. Gerou-se a figura com velocidades w positivas e negativas, até o instante em que a velocidade vn chegou a zero ($vn = 0$).

Observando a Figura 4.3, verificamos que todas as velocidades impostas ao robô devem estar dentro do sólido cujas faces são os planos referidos. Para dimensionar correctamente um determinado conjunto de velocidades de referência que ultrapassa os limites referidos, deve-se traçar uma recta da origem até o ponto

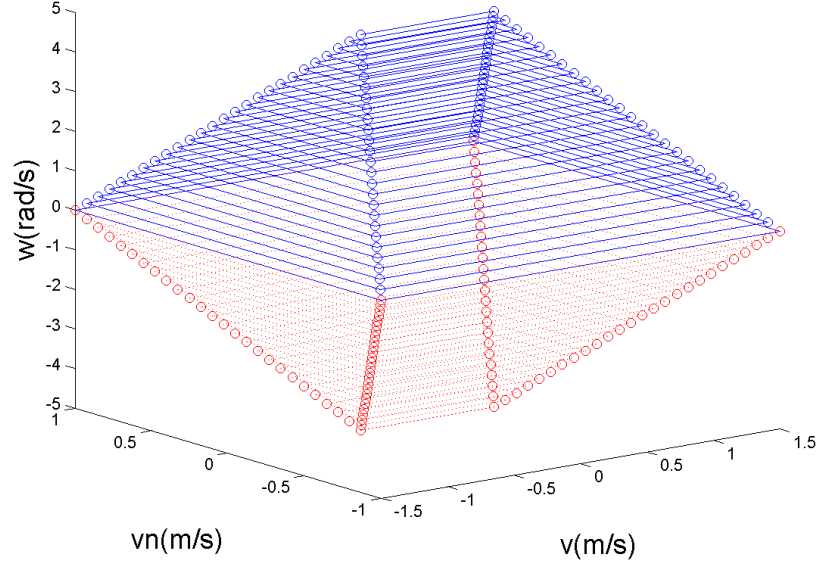


Figura 4.3: Planos das velocidades máximas do robô nos eixos v, vn, w .

da velocidade pretendida. O ponto de intersecção desta recta com um dos planos dará a velocidade máxima possível, mantendo a direcção do deslocamento e respeitando os limites dos motores do robô.

Considere-se então, um ponto $P_{ref} = (v_{ref}, vn_{ref}, w_{ref})$, que representa uma referência de velocidades para o robô, impossível de se atingir. Inicialmente, é traçada uma recta da origem dos eixos de coordenadas (v, vn, w) até o ponto P_{ref} , depois calcula-se o ponto de intersecção ($\bar{P}_{ref}$) entre esta recta e um dos planos, conforme a formulação:

- $G = c_1v + c_2vn + c_3w + v_i = 0$, equação do plano;
- $P_{ref} = (v_{ref}, vn_{ref}, w_{ref})$, o ponto de velocidades de referências;
- $\bar{P}_{ref} = \alpha P_{ref}$, o ponto de velocidades redimensionadas;
- α , o factor de redimensionamento.

Então,

$$\begin{aligned}\alpha P_{ref} &= (\alpha v_{ref}, \alpha v n_{ref}, \alpha w_{ref}) \\ \alpha P_{ref} &\in G\end{aligned}\tag{4.31}$$

substituindo αP_{ref} na equação do plano,

$$\begin{aligned}c_1 \alpha v_{ref} + c_2 \alpha v n_{ref} + c_3 \alpha w_{ref} + v_i &= 0, \\ \alpha (c_1 v_{ref} + c_2 v n_{ref} + c_3 w_{ref}) + v_i &= 0, \\ \alpha &= \frac{-v_i}{c_1 v_{ref} + c_2 v n_{ref} + c_3 w_{ref}}\end{aligned}\tag{4.32}$$

assim determina-se o valor de α . Por exemplo para o plano da equação 4.23, teríamos os seguintes valores,

- $v_i = -v_1$,
- $c_1 = 0$,
- $c_2 = 1$,
- $c_3 = b$.

A seguir apresenta-se um exemplo com o ponto $P_{ref} = (1, -1.2, 2)$, representando as velocidades $v, v n$ e w respectivamente, com os seguintes valores de velocidades máximas para os motores,

- $v_1 = v_3 = 1(m/s)$,
- $v_2 = v_4 = 1.5(m/s)$,
- e o parâmetro $b = 0.2(m)$.

O ponto $P_{ref} = (1, -1.2, 2)$ ultrapassou os limites de velocidades, conforme as Figuras 4.4 e 4.5, após redimensionar as velocidades, obteve-se o ponto redimensionado $\bar{P}_{ref} = (0.625, -0.75, 1.25)$, que é o ponto de intersecção da recta com o plano, com $\alpha = 0.625$.

A Figura 4.5 mostra o plano vn, w onde se pode ver de forma mais clara o ponto de intersecção da recta com o plano.

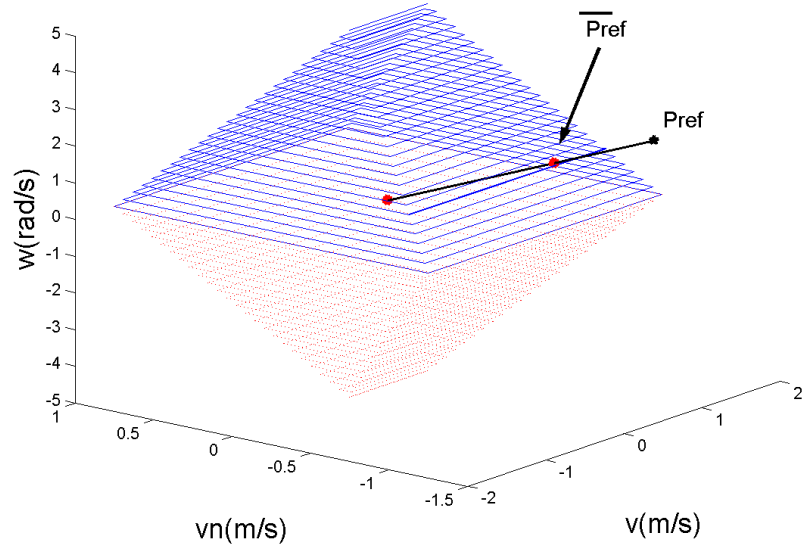


Figura 4.4: Pontos P_{ref} , $\overline{P}_{ref}$.

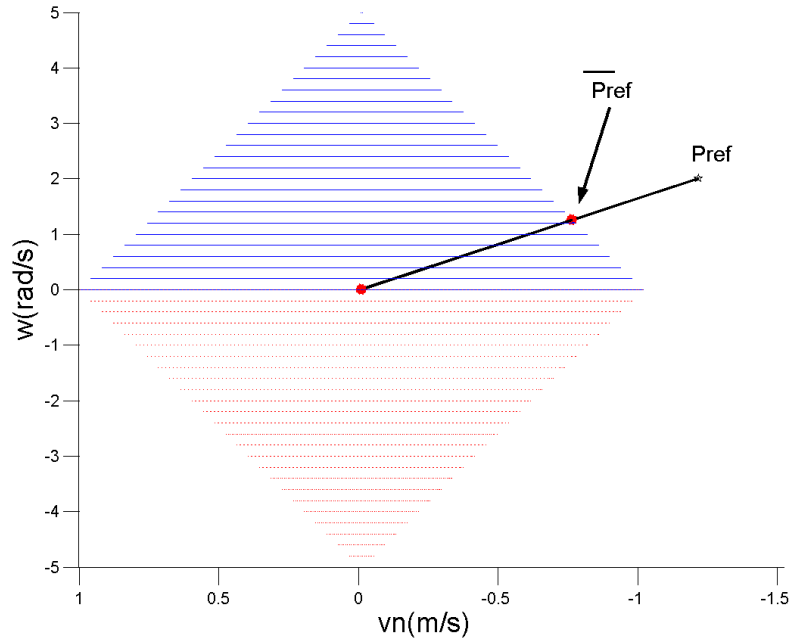


Figura 4.5: Pontos P_{ref} , $\overline{P}_{ref}$.

4.2.3 Optimizaç o dos Par metros do Controlador

Definida a estrutura do controlador, resta escolher ganhos apropriados para k_1 e k_2 de forma que o rob  siga uma traject ria com o melhor desempenho poss vel. Para medir o desempenho no seguimento de uma traject ria, foi definida uma funç o custo C da seguinte forma:

$$C(k_1, k_2) = E_d P_d + E_a P_a + (T_r - T_i) P_t.$$

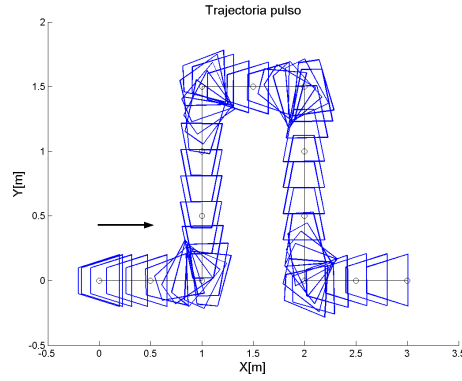
onde:

- $E_d \hookrightarrow$ erro m dio quadr tico(MSE-*Mean Square Error*) correspondente a e_s , no seguimento da traject ria;
- $E_a \hookrightarrow$ erro m dio quadr tico correspondente a e_θ , no seguimento da traject ria;
- $T_i \hookrightarrow$ tempo ideal para percorrer a traject ria;
- $T_r \hookrightarrow$ tempo que o rob  percorreu a traject ria;
- $P_d, P_a, P_t \hookrightarrow$ pesos atribuídos a cada parcela da funç o custo.

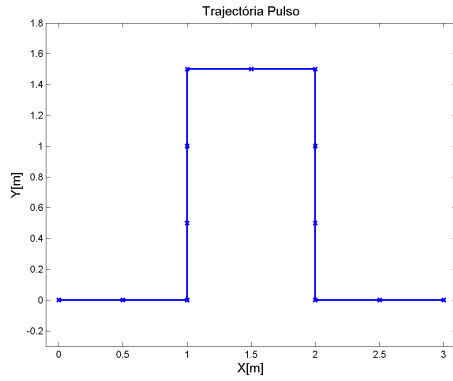
O procedimento para definir os ganhos ideais para k_1 e k_2 foi percorrer uma traject ria pr -definida com diferentes valores de ganhos e obter o valor da funç o custo. O ganhos que resultem no menor valor da funç o custo s o considerados os ideais. Foi utilizado o modelo do rob , descrito na Sec  o 3, para realizar as simulaç es do seguimento da traject ria.

Atrav s de ensaios realizados com o rob  sabe-se que com ganhos k_1 e k_2 com valores acima de 20 apresentam oscilaç es no seguimentos da traject ria. Sendo assim, utilizou-se valores de ganhos entre 0(zero) e 20(vinte) com uma resoluç o de 0.5 para as simulaç es. Utilizou-se uma velocidade linear de refer ncia v_r fixa no seguimento de toda a traject ria. Simulou-se para 3 diferentes velocidades lineares de refer ncia v_r : 1, 0.7 e 0.4 (m/s).   importante salientar que s o 1600 simulaç es para cada velocidade, totalizando um total de 4800 simulaç es. Sem o aux lio de um modelo do rob , ficaria invi vel a realizaç o deste teste. Note-se no entanto que este n mero de simulaç es   realizado pelo computador num espaço de tempo muito curto e portanto n o foi necess rio recorrer-se a algoritmos de minimizaç o mais sofisticados.

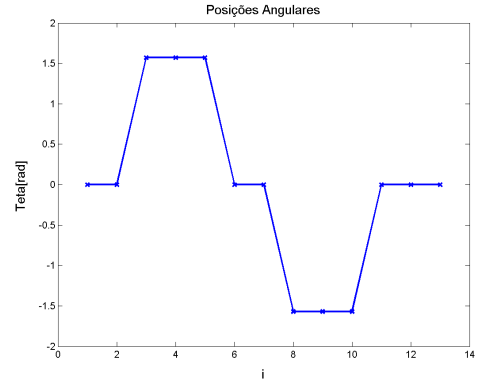
A trajectória escolhida para realizar as simulações possui características especiais como sejam as mudanças bruscas de direcção e orientação do robô, para testar o controlador em situações de difícil seguimento da trajectória. Na Figura 4.6 apresenta-se a trajectória de referência “Pulso”.



(a) Trajectória pulso.



(b) Posições (x_r, y_r) .

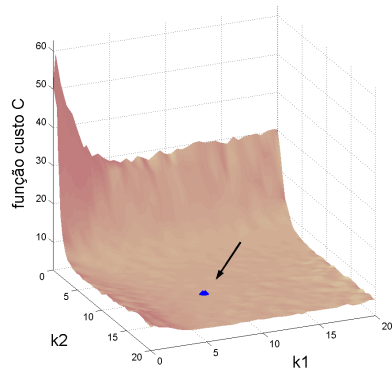


(c) Posições angulares θ_r .

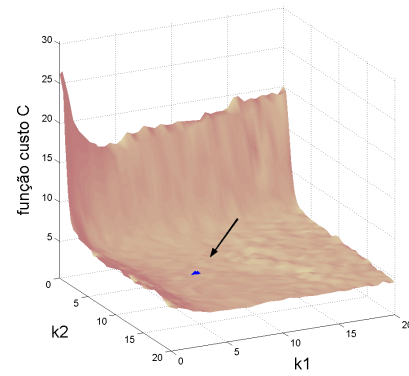
Figura 4.6: Trajectória “Pulso”, para cálculo da função custo.

Nas Figuras 4.7(a), 4.7(b) e 4.7(c) apresentam-se os gráficos das funções custo para as três velocidades de v_r e os ganhos com valores entre 0 e 20 com resolução 0.5. Os valores dos mínimos das funções custo estão na Tabela 4.1, assim como os ganhos k_1 e k_2 respectivos e os pesos (P_d , P_a e P_t) atribuídos aos erros.

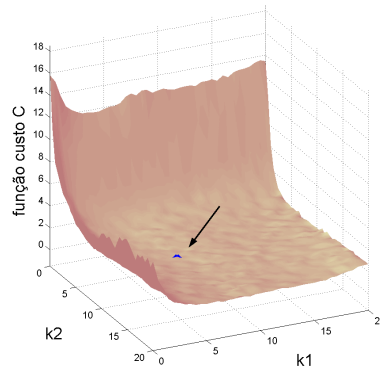
Os valores dos pesos (P_d , P_a e P_t) foram determinados de forma a nivelar a ordem de grandeza dos termos correspondentes a E_d , E_a e $(T_r - T_i)$. Além disso, utilizando os pesos existe a possibilidade de penalizar mais um erro do



(a) $v_r = 1 \text{ m/s}$



(b) $v_r = 0.7 \text{ m/s}$



(c) $v_r = 0.4 \text{ m/s}$

Figura 4.7: Mínimos das funções custo, trajetória "Pulso".

4.2 Controlador Reactivo

$v_r(m/s)$	C	k_1	k_2	P_d	P_a	P_t
1	5.57	8	12.5	1000	10	1
0.7	3.87	7	10.5	1000	10	1
0.4	1.21	7.5	9	1000	10	0.5
média	-	7.5	10.5	-	-	-

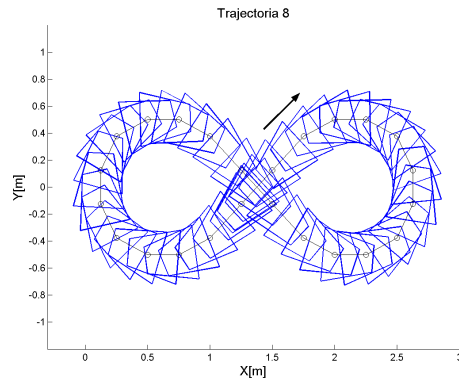
Tabela 4.1: Valores mínimos da função custo para diferentes v_r , trajectória "Pulso".

que o outro, dependendo de como se quer o comportamento do robô. Os valores actualmente utilizados no controlador, são a média entre os ganhos, conforme a Tabela 4.1. Já que não houve grandes diferenças entre os valores dos ganhos com a mudança da velocidade v_r de referência, optou-se por utilizar a média. Testou-se a média dos ganhos em todas as velocidades (0.4, 0.7, 1(m/s)) e não se verificou mudanças significativas que justificassem a utilização de diferentes ganhos para diferentes velocidades de referência v_r .

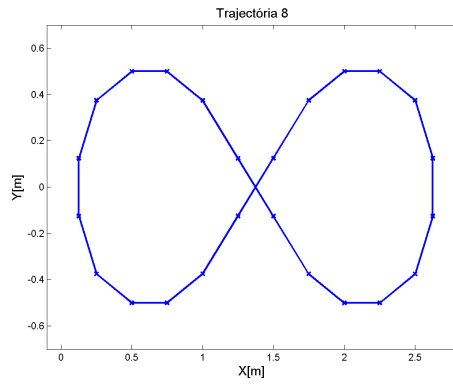
Repetiu-se o procedimento para definir os ganhos ideais para k_1 e k_2 com uma trajectória do tipo "8", Figura 4.8, esta trajectória possui uma característica diferente da primeira trajectória testada, pois o robô não efectua mudanças bruscas de direcção e orientação. Nas Figuras 4.9(a), 4.9(b) e 4.9(c) apresentam-se os gráficos das funções custo para as três velocidades de v_r e os ganhos com valores entre 0 e 20 com resolução 0.5. Na Tabela 4.2, apresenta-se os valores dos mínimos das funções custo, os ganhos k_1 e k_2 respectivos e os pesos (P_d , P_a e P_t) atribuídos aos erros. Manteve-se os mesmos pesos usados com a primeira trajectória, para efeito de comparação. Os resultados para o ganho k_2 foi igual com as duas trajectórias. Com o ganho k_1 houve diferenças mais significativas, o ganho diminuiu com a diminuição da velocidade de referência v_r . Isso deve-se a característica da trajectória "8", que é uma trajectória mais suave, onde o robô segue mais facilmente, provocando um erro menor de E_d comparado com a primeira trajectória.

$v_r(m/s)$	C	k_1	k_2	P_d	P_a	P_t
1	2.84	9	12.5	1000	10	1
0.7	0.72	5	10.5	1000	10	1
0.4	0.22	3	9	1000	10	0.5

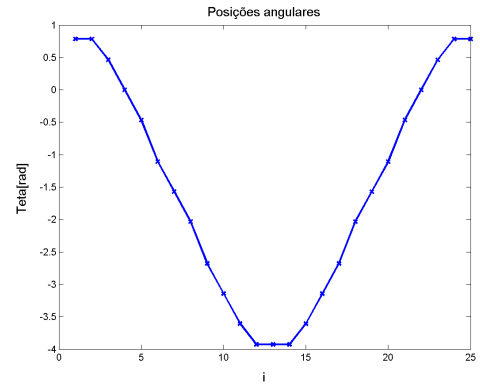
Tabela 4.2: Valores mínimos da função custo para diferentes v_r , trajectória "8".



(a) Trajetória 8.

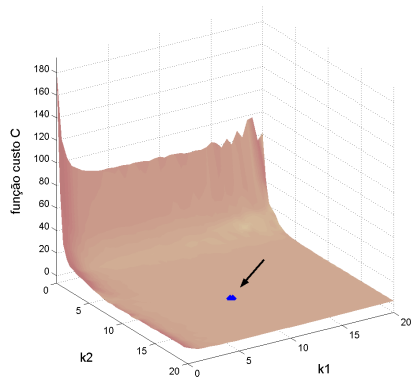


(b) Posições (x_r, y_r) .

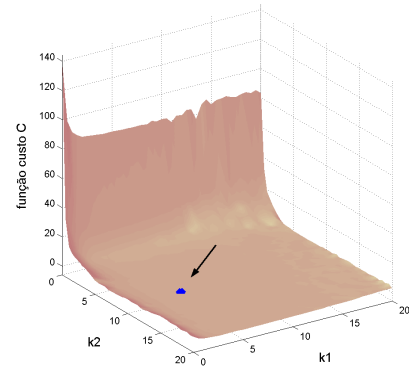


(c) Posições angulares θ_r .

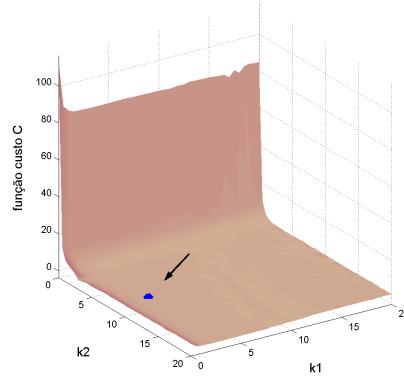
Figura 4.8: Trajetória "8".



(a) $v_r = 1 \text{ m/s}$



(b) $v_r = 0.7 \text{ m/s}$



(c) $v_r = 0.4 \text{ m/s}$

Figura 4.9: Mínimos das funções custo, trajetória “8”.

4.2.4 Resultados Experimentais

Nas Figuras 4.10 e 4.11 apresentam-se resultados reais e de simulação no seguimento da trajectória do tipo “Pulso”, Figura 4.6(a), e tipo “8”, Figura 4.8(a). Os ganhos utilizados foram $k_1 = 7.5$, $k_2 = 10.5$ e a velocidade linear de referência $v_r = 1(m/s)$ fixa para toda a trajectória. Verifica-se que os resultados de simulação e reais são bem parecidos. A maior diferença está nas posições do robô (x,y) , é preciso ter em conta as irregularidades e atritos do chão onde foram executados os ensaios, isso pode causar pequenos desvios, que nos ensaios simulados não ocorrem.

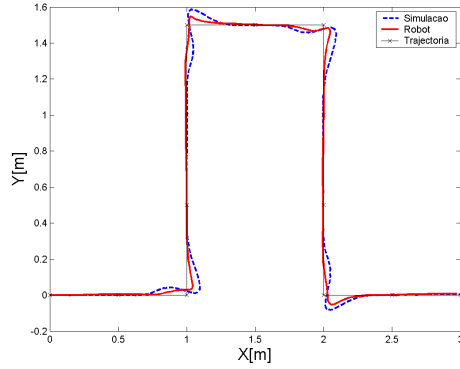
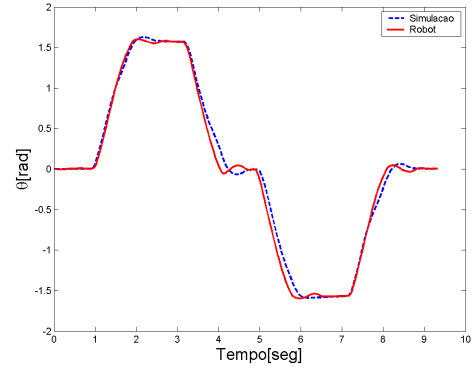
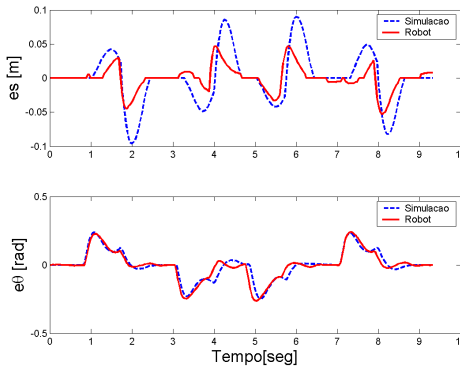
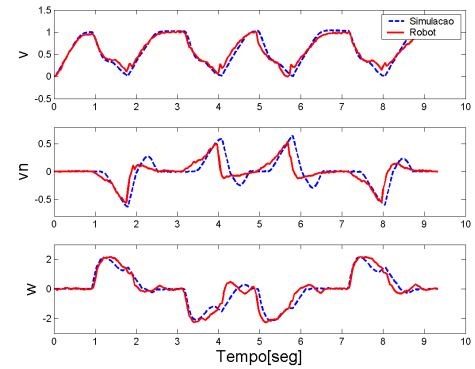
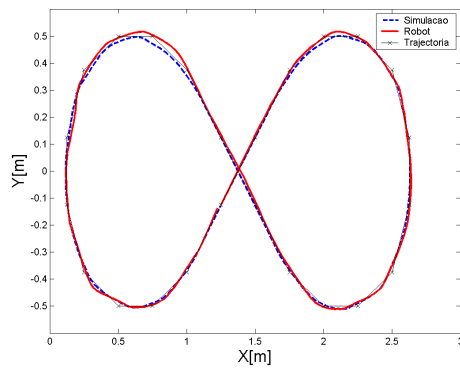
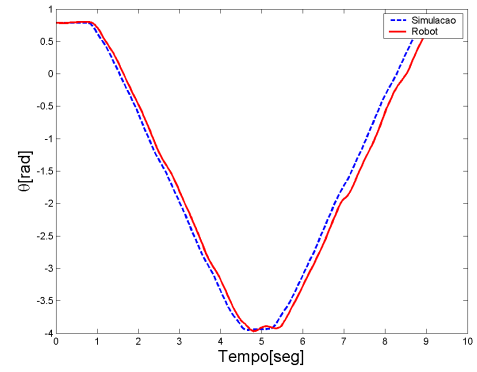

 (a) Posições (x,y) .

 (b) Posições angulares (θ)

 (c) Erros e_s e e_θ .

 (d) $v(m/s)$, $vn(m/s)$, $w(rad/s)$

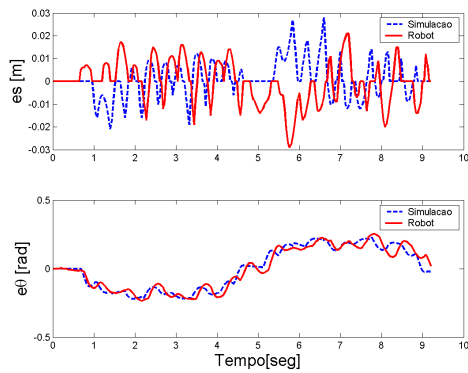
Figura 4.10: Trajectória “Pulso”, simulada e real.



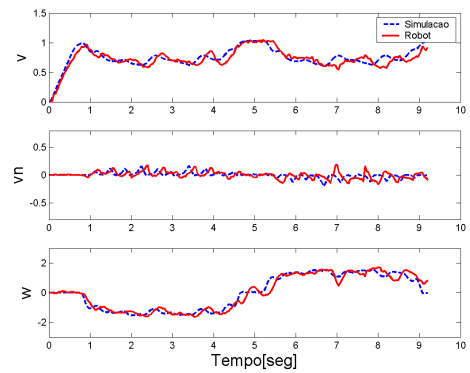
(a) Posições (x, y) .



(b) Posições angulares (θ)



(c) Erros e_s e e_θ .



(d) $v(m/s)$, $vn(m/s)$, $w(rad/s)$

Figura 4.11: Trajectória “8”, simulada e real.

Para testar o algoritmo de restrição dos sinais de controlo realizou-se dois ensaios com o robô. O primeiro ensaio não utilizou o algoritmo, com isso as velocidades de referência calculadas pelo controlador foram directamente aplicadas no robô. No segundo ensaio, utilizou-se o algoritmo. Na Figura 4.13 apresentam-se os resultados do teste, onde o robô realizou o seguimento da trajectória da Figura 4.12. A trajectória possui um segmento de recta, com a sua posição inicial em $(0, 0, 0)[m, m, rad]$ e a posição final em $(-2, 0, \pi)[m, m, rad]$, a velocidade linear de referência para todo o segmento foi definida em $1[m/s]$. A Figura 4.13(a) apresenta o seguimento da trajectória em simulação e a Figura 4.13(b) com o robô. Os resultados mostram que com o uso do algoritmo o robô segue a trajectória com melhores desempenhos quando comparado com os resultados sem o uso do algoritmo. Isso porque são respeitadas as velocidades dos motores do robô. Verifica-se nas Figuras 4.13(c), 4.13(d) e 4.13(f) a acção do algoritmo que limita as amplitudes das velocidades, evitando a saturação dos motores.

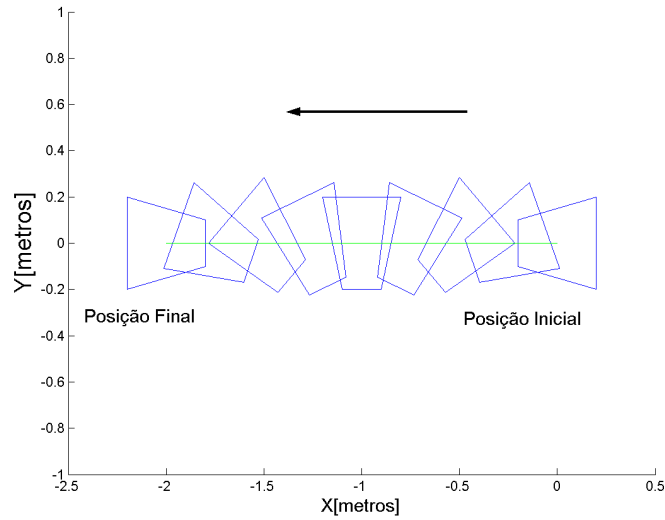
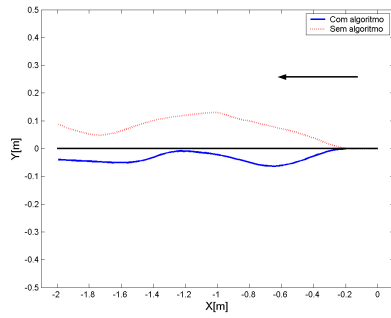
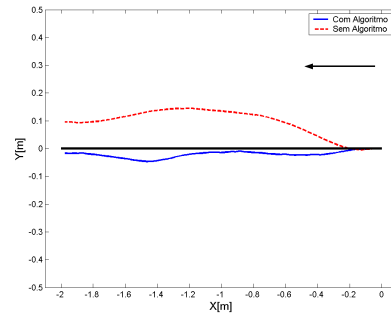


Figura 4.12: Trajectória para teste.

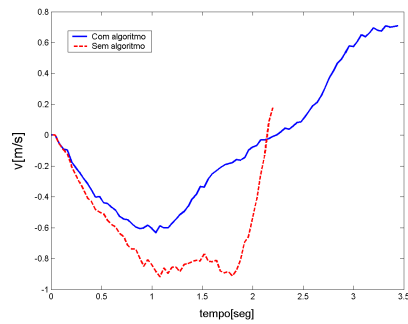
4.2 Controlador Reactivo



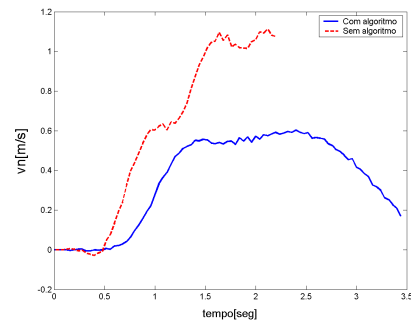
(a) Posições (x,y) - simulação.



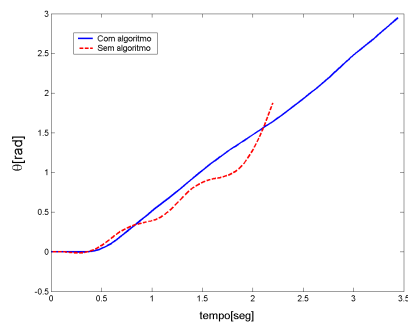
(b) Posições (x,y) - com o robô.



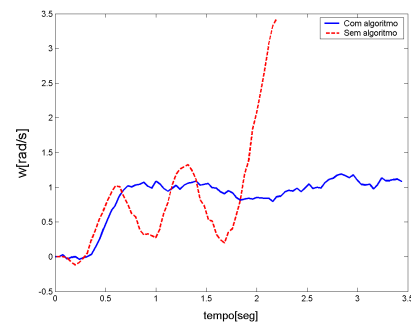
(c) Velocidade linear v - com o robô.



(d) Velocidade linear vn - com o robô.



(e) Posições angulares (θ) - com o robô.



(f) Velocidade angular w - com o robô.

Figura 4.13: Resultados do Algoritmo de restrição.

4.2.5 Conclusões

Após a implementação e análise dos resultados obtém-se algumas conclusões:

- Para robôs omnidireccinais a possibilidade de definir como referência a velocidade de navegação ao longo dos segmentos de recta, além da posição e orientação do robô, é essencial. Principalmente em situações especiais como por exemplo o encaixe de um AGV em prateleiras ou o posicionamento do robô para efectuar um remate para a baliza.
- A possibilidade de ajuste dos ganhos k_1 e k_2 , tornam o controlador flexível para diversos tipos de trajectórias.
- O modelo do robô foi essencial para o projecto e sintonia do controlador.
- Os resultados da simulação e os resultados experimentais foram similares, devido a um bom modelo do robô.
- O mapeamento das restrições dinâmicas dos motores, melhoram o desempenho no seguimento das trajectórias.
- O controlador possui baixo tempo computacional ($\cong 1ms$ com um computador portátil com processador Pentium M, $1.7Ghz$), muito útil em aplicações em tempo-real.
- Em trajectórias com mudanças bruscas de direcção e orientação o controlador apresenta sobrelongação, conforme Figura 4.10, devido a própria característica do robô que não tem a capacidade de prever as próximas referências.

4.3 Controlador Preditivo

Actualmente, o conceito de predição é cada vez mais utilizado em diversos sectores para além do próprio controlo de processos: na manutenção, na previsão de custos, na detecção precoce de avarias, etc. Este conceito tem vindo a complementar cada vez mais a acção de realimentação, acção esta mais relacionada com reacção do que com antecipação ou predição. O controlo por antecipação (*feed-forward*) é naturalmente englobado no conceito de predição. A ideia básica do controlo preditivo (*MPC-Model Predictive Control*) é simples: com base num modelo do processo prevê-se o comportamento do mesmo para as diferentes acções (controlos). Da optimização de uma função custo, dependente dessas acções, obtém-se a acção óptima a aplicar ao processo.

Os conceitos de predição, realimentação e antecipação estão intimamente ligados ao nosso dia-a-dia, uma analogia feita por [Camacho & Bordons \(2004\)](#) é particularmente clara na explicação destes conceitos: conduzir (controlar) um automóvel por realimentação (reacção) seria como conduzi-lo olhando apenas para o retrovisor. Só depois de detectarmos o desvio é que reagiríamos e corrigiríamos a trajectória. A maneira usual como conduzimos é por predição e antecipação. Temos um modelo do automóvel (sabemos como ele reage aos nossos comandos), olhamos para a frente vendo a trajectória a seguir (valores futuros da referência) e tendo em conta as eventuais perturbações (surgiu um carro à nossa frente, um peão que se atravessa, um buraco na estrada ...) e optimizando um critério (tempo, segurança, ...) definimos por predição e antecipação uma série de acções (controlos) a tomar. Tal como no controlo preditivo a estratégia seguida é também de horizonte recuante, ou seja, a todo o momento, face à nova informação que nos chega estamos a reavaliar a situação e a tomar por óptima uma nova sequência de acções, embora, obviamente, só apliquemos a que se torna imediata.

No controlo preditivo existem assim alguns componentes básicos comuns a todo o tipo de sistemas preditivos: o modelo do processo e o modelo ou tipo de perturbações, o horizonte de predição, o horizonte de controlo e o critério a optimizar. As diferenças entre os diversos tipos de controladores revelam-se nas diferentes opções tomadas para cada um destes componentes sendo sempre comum a todos a estratégia de horizonte recuante.

Tal como o controlo por posicionamento de pólos (PP) e o controlo linear

quadrático (LQ), o controlo preditivo pertence a uma classe de controladores baseados em modelos do processo a controlar. Se o processo for linear, se não existirem limitações nos sinais e se a referência ou trajectória a seguir for relativamente simples, então todos os controladores mencionados têm um desempenho semelhante desde que devidamente dimensionados e utilizados todos os seus graus de liberdade. Isto por vezes não é simples pois a relação entre o desempenho desejado para o sistema em malha fechada e os parâmetros de ajuste do controlador nem sempre é linear e evidente o que dificulta bastante a utilização destes controladores. Realmente uma das vantagens do controlador preditivo é a sua relativa facilidade em ser ajustado, além do carácter intuitivo do modo de ajuste dos parâmetros, desde que o modelo do processo seja uma boa aproximação deste. Numerosa literatura refere as vantagens do controlo preditivo algumas das quais se passa a enunciar:

- A já referida facilidade de ajuste dos seus parâmetros e o seu carácter intuitivo.
- O conceito pode facilmente ser extrapolado para sistemas de múltiplas entradas e múltiplas saídas.
- Modelos não lineares de processos podem ser utilizados explicitamente no controlador, contrariamente aos controladores PP ou LQ.
- É uma metodologia que pode ter em conta de uma forma sistemática, durante o projecto do controlador, as limitações dos sinais de controlo e saída do processo. Este é de facto um dos aspectos mais atraentes dos controladores preditivos.
- As perturbações de evolução conhecida podem ser tratadas com relativa facilidade.
- Consegue facilmente controlar uma grande variedade de processos sem ser necessário um cuidado muito especial. Nestes processos incluem-se processos com atrasos, de fase não mínima e instáveis em malha aberta.
- A actuação por antecipação (*feed-forward*), face a perturbações mensuráveis ou variações no sinal de referência, pode ser incluída no controlador de uma forma natural.

- Trajectórias do sinal de referência pré-definidas ou pré-programadas podem ser incluídas na previsão efectuada pelo controlador resultando, obviamente, em desempenhos superiores. Esta é uma situação vulgar na robótica ou na indústria alimentar ou química em que se processam produtos por lotes.

Naturalmente que os controladores preditivos têm também desvantagens. As mais referidas são as que se enumeram de seguida:

- Tal como todos os controladores baseados em modelos, se o modelo for desadequado ao processo ou simplesmente mal conhecido o resultado provavelmente não será satisfatório.
- Devido ao tipo de metodologia aberta, propiciou o fácil aparecimento de numerosas variantes cada qual com o seu pequeno pormenor diferente da outra, GPC (Generalized Predictive Control, [Clarke et al. \(1987\)](#)), DMC (Dynamic Matrix Control, [Cutler & Ramaker \(1980\)](#)), EPSAC (Extended Prediction Self-Adaptive Control, [DeKeyser & Cauwenberghe \(1985\)](#)), PFC (Predictive Functional Control, [Richalet \(1987\)](#)), EHAC (Extended Horizon Adaptive Control, [Ydstie \(1984\)](#)) e UPC (Unified Predictive Control, [Soeterboek \(1992\)](#)[Soeterboek et al. \(1990\)](#)) entre outros. Embora essas diferenças pareçam, por vezes, ser mínimas podem conduzir a desempenhos do sistema em malha fechada bastante distintos o que pode confundir o utilizador e colocá-lo indeciso face a qual controlador aplicar em cada situação. No entanto o controlador UPC engloba em si vários controladores preditivos com os GPC, DMC, EPSAC e EHAC, dando o seu autor em [Soeterboek \(1992\)](#) algumas indicações quanto à metodologia de ajuste de alguns dos parâmetros do controlador.
- A relativamente fraca base de sustentação teórica que garanta e/ou descreva de uma forma clara e genérica a influência dos diferentes parâmetros do controlador na estabilidade, no nível de desempenho e na robustez.

O saldo final no entanto é bastante positivo, sendo este facto claramente demonstrado pelo sucesso de inúmeras aplicações industriais do controlo preditivo. Na robótica móvel ainda mais, pois os robôs móveis estão cada vez mais inseridos

em ambientes dinâmicos, onde a capacidade de prever as acções de controlo é uma ferramenta essencial.

Inúmeras e variadas estratégias de controlo preditivo aplicadas a robôs móveis podem ser encontradas na literatura. Em Ramirez *et al.* (1999) foi implementado um MPC para navegação de robôs com algoritmos genéticos para realizar optimizações não-lineares em tempo real. Xianhua *et al.* (2005) apresenta um método para seguimento de trajectória na qual combina controlo preditivo e lógica fuzzy, onde o controlo preditivo é usado para prever a posição e orientação do robô, enquanto que controlador fuzzy trata das não-linearidades do sistema. Um esquema para seguimento de trajectória baseado em controlo preditivo com redes neuronais é apresentado em Gu & Hu (2002), onde uma rede neuronal *multi-layer back-propagation* é usada para modelar as não-linearidades cinemáticas de um robô. Em Gomez-Ortega & Camacho (1994) uma rede neuronal *multilayer perceptron* foi treinada para reproduzir de uma forma supervisionada o comportamento de um MPC. Em Camacho & Bordons (2004) utilizou-se redes neuronais para obter a minimização da função custo. Outra estratégia usando redes neuronais foi apresentada em Piche *et al.* (2000), onde obtém-se modelos dinâmicos não-lineares a partir de dados empíricos de um algoritmo MPC.

Uma nova abordagem de controlador preditivo não-linear aplicado a robôs móveis omni-direccionais é apresentada nessa secção, onde métodos numéricos de optimização não-linear são usados para obter a minimização da função custo do controlador. A motivação desta abordagem surgiu quando obteve-se bons tempos de minimização ($\cong 20ms$) na minimização da função custo. Para prever o comportamento do robô utilizou-se o modelo não-linear do robô, apresentado na Secção 3. Actualmente com processadores potentes em computadores pessoais, tornou-se possível essa abordagem com algoritmos de optimização em tempo-real.

4.3.1 Estrutura do Controlador

A metodologia dos controladores MPC é baseada no esquema da Figura 4.14: As saídas futuras para um determinado horizonte N (horizonte de predição), são estimadas a cada instante k usando o modelo do processo. A predição das saídas do processo $\hat{Y}(k+j|k)$ para $j = 1...N$ depende dos valores conhecidos para o

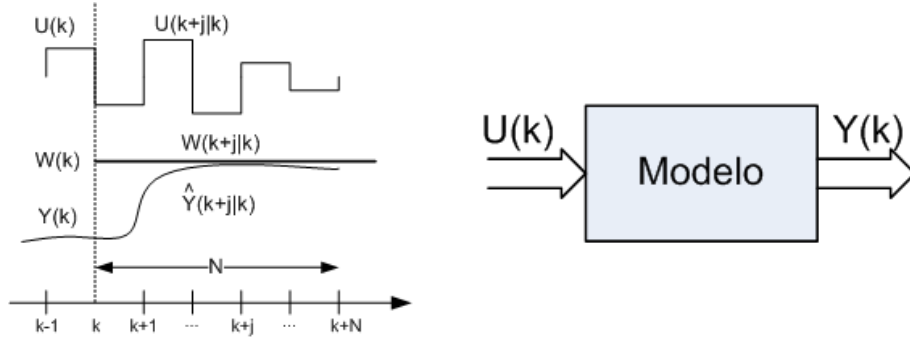


Figura 4.14: Metodologia do controlador.

instante k e dos sinais de controlo futuros $U(k+j|k)$, $j = 0 \dots N-1$, os quais são aplicados ao modelo. O conjunto dos sinais de controlo futuros são calculados pela optimização de um determinado critério (por ex. função custo), visando manter o processo mais próximo possível do sinal de referência $W(k+j)$. O sinal de controlo $U(k|k)$ é enviado para o processo enquanto os próximos sinais de controlo calculados são desprezados, e assim que o próximo instante $Y(k+1)$ é conhecido pode-se repetir o ciclo de controlo.

Com o objectivo de implementar a estratégia do MPC projectou-se a estrutura de controlo apresentada na Figura 4.15:

- **Trajectória Referência** - $R \Rightarrow$ trajectória completa a efectuar;
- **Trajectória Referência Controlador** - $W \hookrightarrow$ sinal de referência em um determinado horizonte de predição;
- **Optimizador** $\hookrightarrow$ calcula os sinais de controlo baseado em critérios definidos na função custo(J);
- **Modelo não linear** $\hookrightarrow$ modelo usado para predição das saídas futuras do processo;
- **Controlador das rodas** $\hookrightarrow$ é o controlador baixo-nível, com os controladores PIDs das rodas do robô. Onde as velocidades lineares desejadas e medidas das rodas são $v_{1..4}$ e $\hat{v}_{1..4}$, respectivamente. A tensão dos motores são $u_{1..4}$ e as correntes medidas dos motores são $i_{1..4}$;
- **Robô** $\Rightarrow$ é o processo a controlar.

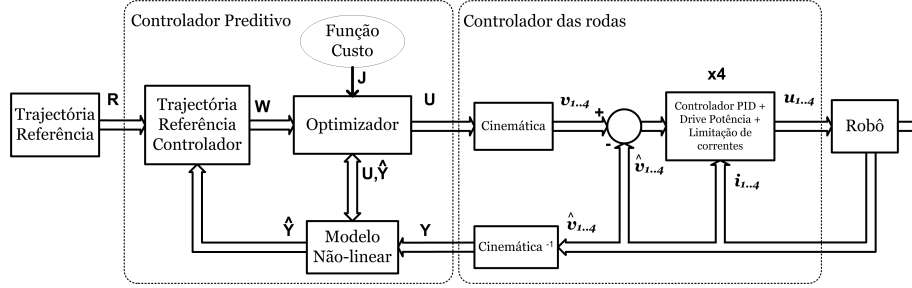


Figura 4.15: Estrutura do Controlador.

4.3.2 Modelo do Processo

O modelo do processo é o principal elemento do MPC, pois através do modelo é possível calcular a predição das saídas do processo nos instantes futuros $\hat{Y}(k+j|k)$. O modelo utilizado no controlador preditivo é o descrito na Secção 3.3. Trata-se de um modelo não-linear que representa o comportamento do robô, com sua dinâmica e restrições nos actuadores, além de permitir o cálculo de predições futuras e o acesso às variáveis de controlo.

Na Figura 4.16 apresenta-se o sistema de coordenadas e parâmetros relacionados com a configuração do robô. A partir desta representação definimos os sinais de entrada e saída do modelo discreto para o robô:

$$Y(k+1) = f(Y(k), U(k)), \quad (4.33)$$

sendo $Y(k) = [x(k) \ y(k) \ \theta(k) \ v(k) \ vn(k) \ w(k)]^T$ o vector estado do robô. Onde $x(k)$ e $y(k)$ determinam a posição do robô e $\theta(k)$ a orientação angular do robô no sistema global de coordenadas $SGC = \{O_G X_G Y_G\}$. Os vectores $v(k)$ e $vn(k)$ são as velocidades lineares do robô e $w(k)$ a velocidade angular do robô. $U(k) = [v_r(k) \ vn_r(k) \ w_r(k)]^T$ são os sinais de controlo do robô; $v_r(k)$ e $vn_r(k)$ são as velocidades lineares pretendidas para o robô e $w_r(k)$ a velocidade angular pretendida para o robô. As velocidades pretendidas para o robô são consideradas constantes em cada intervalo de amostragem.

Definimos como entrada do modelo o vector de velocidades $U(k)$, e como saídas do modelo o vector $Y(k)$, conforme a Figura 4.14.

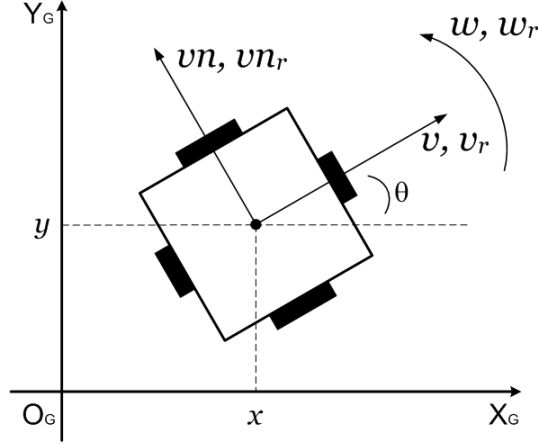


Figura 4.16: Esquemático do robô.

4.3.3 Função Custo

A aplicação em questão trata da navegação de robôs móveis, sendo assim projectou-se uma função custo para penalizar os erros no seguimento de trajetórias, conforme a equação 4.34. A saída futura em um determinado horizonte deve seguir um determinado sinal de referência e ao mesmo tempo, o esforço de controlo necessário deve ser penalizado.

$$\begin{aligned}
 J(N_1, N, N_u) = & \sum_{j=N_1}^N \lambda_1 ([\hat{x}(k+j|k) - x_c(k+j)]^2 + [\hat{y}(k+j|k) - y_c(k+j)]^2) + \\
 & \sum_{j=N_1}^N \lambda_2 [\hat{\theta}(k+j|k) - \theta_c(k+j)]^2 + \\
 & \sum_{j=1}^{N_u} \lambda_3 [\Delta U(k+j-1)]^2
 \end{aligned} \tag{4.34}$$

O primeiro termo de J penaliza o erro de posição, o segundo penaliza o erro na orientação do robô e o terceiro penaliza o esforço nos sinais de controlo do robô. Na função custo J consideram-se:

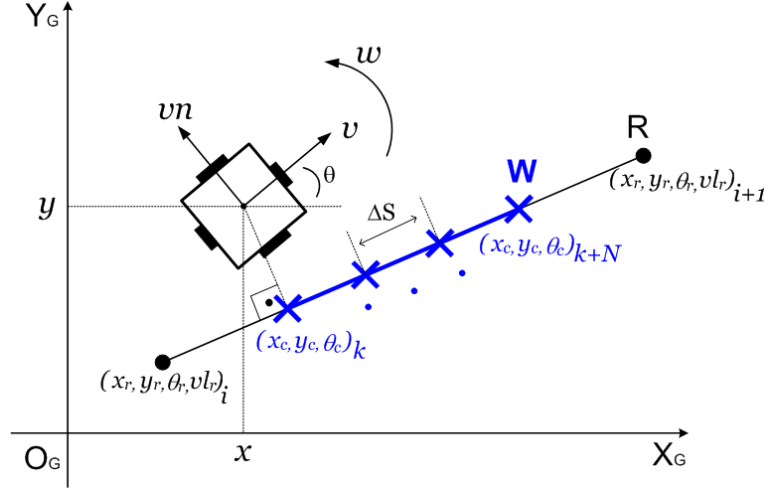
- $\hat{Y}(k+j|k) = [\hat{x}(k+j|k) \ \hat{y}(k+j|k) \ \hat{\theta}(k+j|k)]^T \leftrightarrow$ como a predição futura da posição e orientação do robô no instante $k+j$ dado o sinal Y no instante de tempo k .

- N_1 , N e $N_u \hookrightarrow$ os horizontes de predição mínimo (N_1) e máximo (N) e o horizonte de controlo (N_u). Os valores de N_1 e N são determinados de forma intuitiva, e determinam os limites dos instantes na qual é desejável que a saída siga a referência. O horizonte de controlo N_u não precisa necessariamente coincidir com o máximo horizonte.
- λ_1 , λ_2 e $\lambda_3 \hookrightarrow$ os coeficientes de peso para as predições, usualmente são utilizados valores constantes ou sequências exponenciais. λ_1 é referente ao erro de posição, λ_2 ao erro da posição angular e λ_3 ao esforço nos sinais de controlo.
- $W(k+j) = [x_c(k+j) \ y_c(k+j) \ \theta_c(k+j)]^T \hookrightarrow$ a trajectória de referência para o controlador. Uma das vantagens do controlo preditivo é que as referências futuras são conhecidas *a priori*, assim o processo pode reagir antes que as mudanças tenham efectivamente acontecido, eliminando os efeitos de atrasos na resposta do processo.
- $\Delta U(k+j-1) = [\Delta v_r(k+j-1) \ \Delta v n_r(k+j-1) \ \Delta w_r(k+j-1)]^T \hookrightarrow$ são as variações do controlo em um determinado horizonte de controlo N_u . Ao penalizar-se apenas as variações do controlo e não os valores absolutos estamos implicitamente a introduzir uma integração e a eliminar erros em regime permanente a referências constantes.

4.3.4 Definição da Trajectória de Referência para o Controlador

O controlador MPC precisa das referências futuras de posições (x, y) e orientação (θ) do robô. A partir da posição e orientação actual do robô e da trajectória de referência R é definida a trajectória para o controlador (W) para os próximos N (horizonte de predição) instantes de tempo, conforme a Figura 4.17.

A trajectória de referência R é definida por posições, orientações e velocidades lineares para o robô no sistema global de coordenadas em duas dimensões $SGC = \{O_G X_G Y_G\}$: $R(i) = [x_r(i) \ y_r(i) \ \theta_r(i) \ v l_r(i)]^T$, $i = 1, \dots, M$, onde M é o número total de pontos da trajectória. As posições para o controlador (x_c, y_c) são calculadas inicialmente na intersecção entre a trajectória de referência R e


 Figura 4.17: Trajetória de referência para o controlador, W .

uma perpendicular, traçada a partir da posição do robô (x, y) . Os seguintes N pontos são espaçados igualmente pela trajetória de referência R pelo parâmetro ΔS (metros), o qual é calculado a partir das velocidades lineares de referência v_{lr} .

A orientação desejada θ_c é calculada segundo uma variação linear entre os valores de referência das posições angulares $\theta_r(i)$ e $\theta_r(i+1)$ da trajetória R .

$$\theta_c = \theta_r(i)(1 - d) + \theta_r(i+1)d. \quad (4.35)$$

onde d é a projecção da posição do robô (x, y) no segmento de recta formado pelos pontos $(x_r(i), y_r(i))$ e $(x_r(i+1), y_r(i+1))$, normalizada para o tamanho deste segmento de recta.

4.3.5 Lei de Controle

Para obter os valores de controlo $U(k)$ é necessário minimizar a função custo J . Isso é feito calculando os valores de predição da saída $\hat{Y}(k+j|k)$ em função dos valores passados de entrada e saída e dos sinais de controlo futuros, usando o modelo e substituindo os valores obtidos na função custo.

A Figura 4.18 apresenta um fluxograma do cálculo dos sinais de controlo $U(k) = [v_r(k) \ vn_r(k) \ w_r(k)]^T$, em um período de amostragem k . Onde $v_r(k)$ e $vn_r(k)$ são as velocidades lineares pretendidas para o robô e $w_r(k)$ a velocidade angular pretendida para o robô. O fluxograma segue o seguinte algoritmo:

Algoritmo 1 - Lei de controlo

1. Inicialmente é calculada a trajectória de referência $W(k+j) = [x_c(k+j) \ y_c(k+j) \ \theta_c(k+j)]^T$, com $j = 1...N$, onde N é o horizonte de predição;
 2. Inicia-se o ciclo de optimização, com a velocidades actuais de $U(k)$;
 3. Executa-se a simulação da trajectória de referência obtendo-se a predição $\hat{Y}(k+j|k)$, com $j = 1...N$;
 4. Calcula-se a função custo J ;
 5. Aplica-se o algoritmo de optimização, obtendo-se novos valores para $U(k)$;
 6. Se o critério de paragem for satisfeito vai para o passo 7. Caso contrário, volta para o passo 3;
 7. Finalmente os últimos valores calculados para $U(k)$ são aplicados no robô.
-

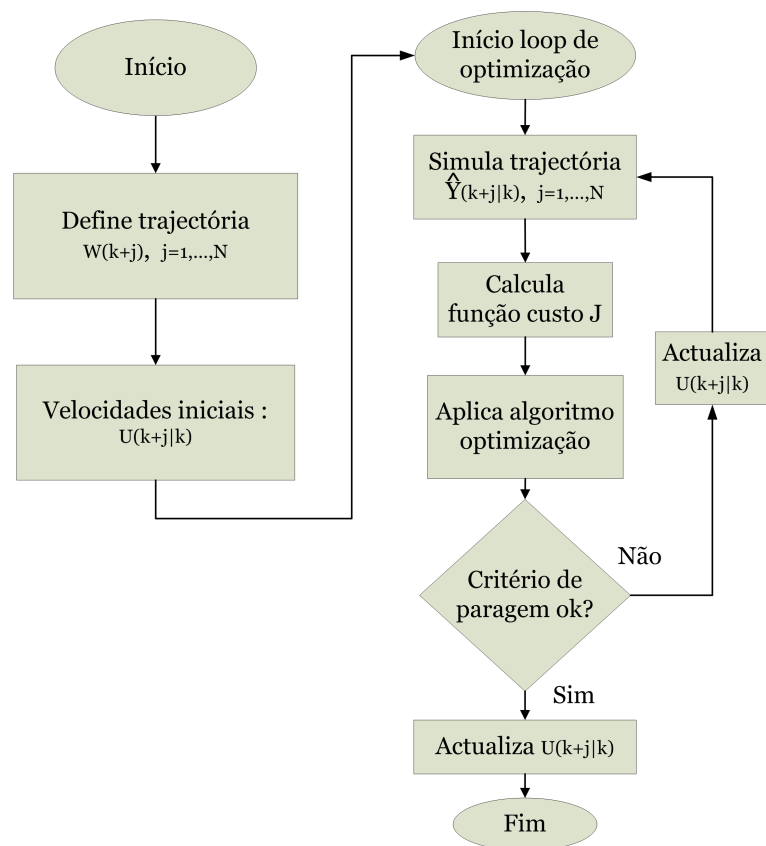


Figura 4.18: Fluxograma da lei de controlo.

4.3.6 Optimiza  o Num rica da Fun  o Custo

Sendo o modelo do processo n o linear, partiu-se para o uso de algoritmos NMPC (*Nonlinear Model Predictive Control*) em tempo real. Um m todo iterativo de optimiza  o num rica tem que ser utilizado, pois n o   poss vel encontrar uma solu  o anal tica. V rios m todos de optimiza  o n o linear podem ser encontrados na literatura Gill *et al.* (1981), Fletcher (1987), Fletcher & Reeves (1964), Grana Drummond & Svaiter (2005), William & Hongchao (2005), William & Hongchao (2006), Jonathan (1994).

Optou-se por utilizar m todos com baixo tempo computacional e flexibilidade de implementa  o. Tr s m todos de optimiza  o foram testados:

- M todo do gradiente (G);
- M todo dos Gradientes Conjugados Fletcher-Reeves (GC_FR);
- M todo dos Gradientes Conjugados Polak-Ribiere (GC_PR).

4.3.6.1 M todo do gradiente

O m todo do gradiente ou *Steepest Descent*   baseado no c culo de gradientes, onde a direc  o de busca adoptada   a direc  o oposta ao vector gradiente ou se baseia nesta informa  o.   um m todo que tem como vantagem principal a sua simplicidade de implementa  o e como desvantagem a sua converg ncia poder ser lenta.

Seja o problema de optimiza  o:

$$\text{Min}\{f(x) : x \in \mathbb{R}^n\}.$$

onde $f(x)$ representa uma fun  o de x . O objectivo   encontrar um valor de x que minimize essa fun  o.

O gradiente da fun  o f   representado por ∇f :

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right], \quad x = [x_1, x_2, \dots, x_n]^T. \quad (4.36)$$

Onde ∇f   um vector em fun  o de x . O gradiente de uma fun  o aponta para a direc  o de m ximo crescimento dessa fun  o. Sendo assim, para minimizar

a função f , os valores de x devem evoluir na direcção oposta. No método do gradiente, segue-se a seguinte fórmula:

$$x_{k+1} = x_k + \alpha d_k, \quad (4.37)$$

sendo $d_k = -\nabla f(x_k)$ a direcção de descida que resulta em $f(x_{k+1}) < f(x_k)$. O escalar α é o tamanho do passo nesta direcção. A seguir apresenta-se um algoritmo para o método do gradiente:

Algoritmo 2 - Método do gradiente

1. Dado x_0 , set $k = 0$;
 2. Calcular $d_k = -\nabla f(x_k)$. Se $d_k \leq \epsilon$, então parar;
 3. Actualiza $x_{k+1} = x_k + \alpha d_k$;
 4. Incrementa $k = k + 1$, vai para o passo 2.
-

O parâmetro ϵ determina o critério de paragem do algoritmo.

4.3.6.2 Método dos Gradientes Conjugados

O método dos Gradientes Conjugados é um caso particular do método das direcções conjugadas, em que elas são obtidas através do gradiente. Através dos sucessivos gradientes podemos construir as direcções que serão conjugadas. Estendendo o método para funções não-lineares, apresenta-se no algoritmo 3 a proposta de Fletcher-Reeves. Outro método, proposto por Polak-Ribiere, difere apenas na escolha de β , conforme o algoritmo 4.

Os métodos dos gradientes conjugados possuem algumas vantagens importantes:

- O gradiente é na maioria das vezes diferente de zero e linearmente independente em relação aos vectores direcção anteriores;
- O cálculo do algoritmo é levemente mais complicado que o método do gradiente, mas converge mais rapidamente;
- Por ser baseado em gradientes, a minimização geralmente possui uma boa convergência.

Algoritmo 3 - Método dos Gradientes Conjugados - Fletcher-Reeves

1. Dado x_0 , calcular $d_0 = -\nabla f(x_0)$, $k = 0$;
2. Calcular $x_{k+1} = x_k + \alpha d_k$, onde α é o tamanho do passo que minimiza $f(x_k + \alpha d_k)$;
3. Calcular $\nabla f(x_{k+1})$, se $\nabla f(x_{k+1}) \leq \epsilon$, parar. Senão calcular

$$\beta_{k+1} = \frac{\nabla f(x_{k+1})^T \nabla f(x_{k+1})}{\nabla f(x_k)^T \nabla f(x_k)};$$

4. Calcular nova direcção $d_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1} d_k$;
 5. Incrementa $k = k + 1$, vai para o passo 2.
-

Algoritmo 4 - Método dos Gradientes Conjugados - Polak-Ribiere

1. Dado x_0 , calcular $d_0 = -\nabla f(x_0)$, $k = 0$;
2. Calcular $x_{k+1} = x_k + \alpha d_k$, onde α é o tamanho do passo que minimiza $f(x_k + \alpha d_k)$;
3. Calcular $\nabla f(x_{k+1})$, se $\nabla f(x_{k+1}) \leq \epsilon$, parar. Senão calcular

$$\beta_{k+1} = \frac{\nabla f(x_{k+1})^T (\nabla f(x_{k+1}) - \nabla f(x_k))}{\nabla f(x_k)^T \nabla f(x_k)};$$

4. Calcular nova direcção $d_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1} d_k$;
 5. Incrementa $k = k + 1$, vai para o passo 2.
-

4.3.7 Resultados Experimentais

Na Secção 4.3.8 apresentam-se resultados de simulações e na secção 4.3.9 apresentam-se resultados reais com o robô no seguimento de trajectórias.

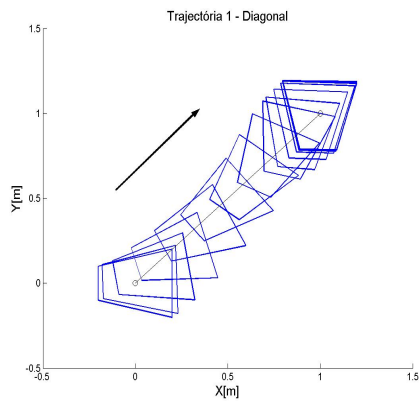
Os resultados de simulações foram realizados com o objectivo principal de testar os diferentes métodos de optimização aplicados ao controlador e também definir uma primeira calibração nos parâmetros do controlador, como horizontes de predição e controlo (N_1 , N e N_u), pesos (λ_1 , λ_2 e λ_3) e critério de paragem da optimização (ϵ).

Os resultados de simulações fornecem-nos informações importantes, como por exemplo os tempos de cálculo utilizados pelos algoritmos de optimização. Esta informação é extremamente importante para definir os parâmetros do controlador quando o mesmo for usado no robô. O robô utiliza um tempo de amostragem de 40 mili-segundos(*ms*), sendo que uma parte significativa deste tempo é utilizada para o processamento de imagem. Sendo assim, é necessário dimensionar o controlador para atender essas condições.

Nas Sub-Secções a seguir serão apresentados os ensaios realizados com horizontes de predição iguais a 8, 10 e 12 amostras. Testou-se outros horizontes de predição, mas os resultados com horizontes de predição maiores que 12 o robô apresentou uma tendência de evitar os pontos de controlo das trajectórias de referência, e com horizontes de predição menor que 8 os algoritmos de optimização necessitam de muitas iterações para minimizar a função custo.

4.3.8 Resultados de Simulações

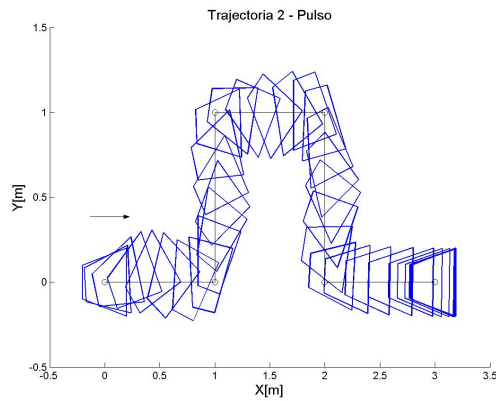
Nos ensaios a seguir foram utilizados dois tipos de trajectórias, com características diferentes. Na primeira trajectória, Figura 4.19 e Tabela 4.3, o robô executa uma translação e rotação ao mesmo tempo no seguimento de uma recta. Na segunda trajectória, Figura 4.20 e Tabela 4.4, o robô executa mudanças bruscas na direcção do movimento e na sua orientação.



índice i	x_r (m)	y_r (m)	θ_r (rad)	vl_r (m/s)
1	0	0	0	1
2	1	1	1.57	1

Figura 4.19: Trajectória "Diagonal".

Tabela 4.3: Parâmetros da trajectória.



índice i	x_r (m)	y_r (m)	θ_r (rad)	vl_r (m/s)
1	0	0	0	1
2	1	0	1.57	1
3	1	1.5	0	1
4	2	1.5	-1.57	1
5	2	0	0	1
6	3	0	0	1

Figura 4.20: Trajectória "Pulso".

Tabela 4.4: Parâmetros da trajectória.

4.3.8.1 Ensaio simulado 1 - ($N = 10$)

No Ensaio simulado 1 foram utilizados os seguintes parâmetros para o controlador:

N_1	N	N_u	λ_1	λ_2	λ_3	ϵ	$\Delta S[m]$
1	10	1	2	1	1	$1e-3$	0.04

Tabela 4.5: Parâmetros ensaio simulado 1.

Optou-se por testar o seguimento da trajectória com um peso maior no erro das posições do robô ($\lambda_1 = 2$) comparado ao peso nos erros das posições angulares do robô ($\lambda_2 = 1$) e ao peso do esforço nos sinais de controlo ($\lambda_3 = 1$). Inicialmente foi definido um horizonte de predição de 10 amostras ($N = 10$). A taxa de amostragem que o robô utiliza é $40ms$, sendo assim é feita uma predição de 0.4 segundos a cada intervalo de amostragem. A partir de ΔS , a velocidade de navegação do robô fica definida em 1 metro por segundo (m/s).

O horizonte de controlo N_u ficou definido como unitário devido a carga computacional. A cada incremento no horizonte de controlo N_u , resultaria em um aumento no tempo computacional para os cálculos dos algoritmos de optimização, devido aos cálculos das derivadas, necessárias para obter o gradiente. O valor do critério de paragem $\epsilon = 1e-3$ foi definido de forma empírica, através de testes preliminares.

Os ensaios foram realizados com os três métodos de optimização: método do gradiente (G), método dos Gradientes Conjugados Fletcher-Reeves (GC_FR) e o método dos Gradientes Conjugados Polak-Ribiere (GC_PR).

Na Tabela 4.6 apresentam-se os seguintes resultados do seguimento das trajectórias, Figuras 4.21(a) e 4.22(a), no ensaio 1 :

- os números mínimo, máximo e a média de iterações do algoritmo de optimização para o seguimento da trajectória;
- o tempo médio do algoritmo de optimização para o seguimento da trajectória;
- o MSE(Mean Square Error) do erro da posição(x,y) e do erro da posição angular(θ) do robô no seguimento da trajectória.

Trajectória diagonal						
Método	Número de iterações			Tempo Optimiz.	MSE(Erro)	
	Máximo	Mínimo	Média	Média(ms)]	x, y	θ
G	31	4	11.21	10.79	5.18e-005	0.0010
GC_FR	18	4	9.23	8.84	8.37e-005	0.0011
GC_PR	13	4	7.86	7.32	7.68e-005	0.0011
Trajectória pulso						
Método	Número de iterações			Tempo Optimiz.	MSE(Erro)	
	Máximo	Mínimo	Média	Média(ms)	x, y	θ
G	40	4	17.85	16.75	4.53e-004	0.0023
GC_FR	23	4	12.47	11.93	4.24e-004	0.0027
GC_PR	22	4	10.70	10.27	4.39e-004	0.0027

Tabela 4.6: Trajectória diagonal e pulso - ensaio simulado 1 ($N = 10$).

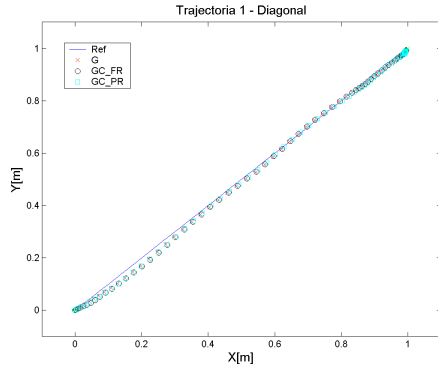
Analisando a Tabela verifica-se que o método GC_PR apresentou uma convergência mais rápida na optimização das trajectórias. Não foi significativa a diferença entre os erros no seguimento da trajectória para os diferentes métodos de optimização. Também se verificou que o tempo gasto para uma iteração do algoritmo de optimização é aproximadamente $1ms$ para todos os métodos, devido aos algoritmos possuírem praticamente o mesmo grau de complexidade na implementação. Os testes foram realizados com computador portátil com um processador Pentium M, 1.7 Ghz.

O critério de paragem ϵ é um factor importante para o número de iterações, geralmente os algoritmos possuem uma contribuição maior para a solução nas primeiras iterações e uma contribuição desprezível nas últimas iterações, como se pode verificar nas últimas iterações nas Figuras 4.21(d) e 4.22(d). Por esse motivo até poderia ser usado um ϵ com valor mais alto, evitando as últimas iterações que pouco acrescentam para a minimização da função custo.

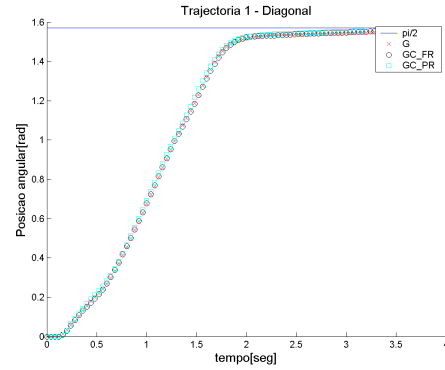
As Figuras 4.21(a) e 4.21(b) apresentam a trajectória(x,y) e a posição angular (θ) realizada em simulação pelo robô, para os três métodos de optimização. A Figura 4.21(c) apresenta os erros no seguimento da trajectória. Os erros apresentados nas Figuras 4.21(c) e 4.22(c), representam a distância entre a posição (x,y) do robô e uma perpendicular da trajectória de referência, conforme a Figura 4.17.

A Figura 4.21(d) apresenta uma optimização realizada no instante de amostragem 1.56 segundos, instante de máximo número de iterações com o método

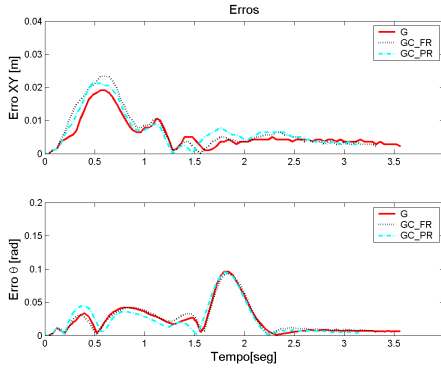
dos gradientes (G). O valor da função custo J (eixo das ordenadas), decresce com o número de iterações (eixo das abcissas). O método de optimização GC_PR apresenta uma convergência mais rápida para a solução.



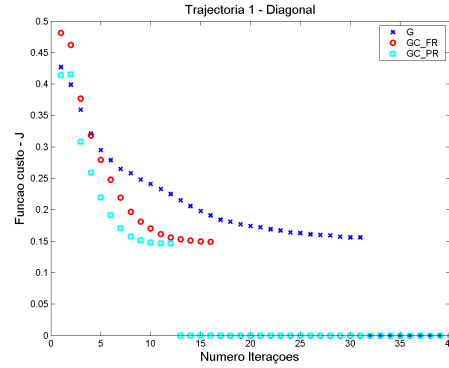
(a) Posições (x,y) .



(b) Posição angular θ .



(c) Erros (x,y,θ) .



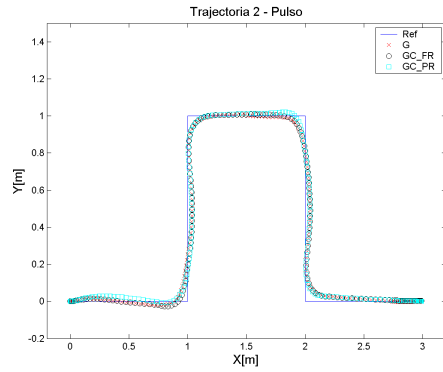
(d) Optimização no instante 1.56 segundos.

Figura 4.21: Trajectória Diagonal - ensaio simulado 1 ($N = 10$).

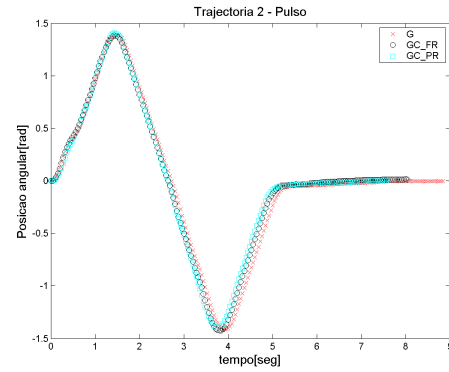
Na Figura 4.22 apresentam-se os gráficos para o seguimento da trajectória pulso. Trata-se de uma trajectória mais complexa, com mudanças bruscas e constantes na direcção do robô. Analisando a Figura 4.22(a) consegue-se verificar perfeitamente o efeito da predição no controlo do robô, quando o mesmo antecipa a acção nos pontos de mudanças bruscas da trajectória, como por exemplo na posição $x = 1$ e $y = 1$.

Da mesma forma que na trajectória diagonal, o método de optimização GC_PR apresentou uma convergência mais rápida para a solução, conforme a Figura

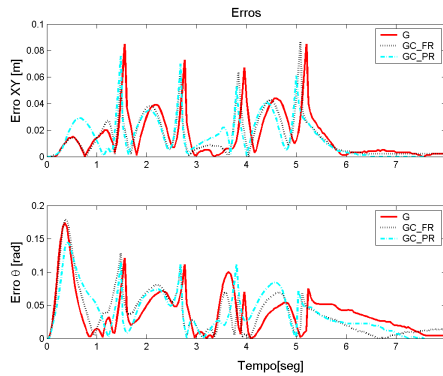
4.22(d).



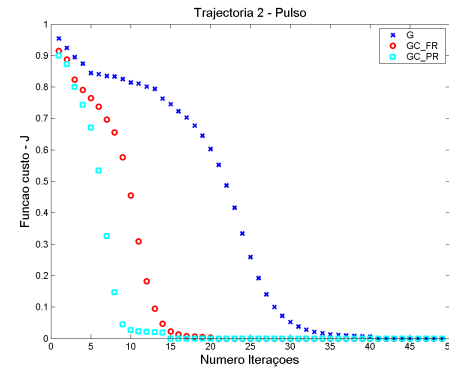
(a) Posições (x, y) .



(b) Posição angular θ .



(c) Erro (x, y, θ) .



(d) Optimização no instante 0.92 segundos.

Figura 4.22: Trajectória Pulso - ensaio simulado 1 ($N = 10$).

4.3.8.2 Ensaio simulado 2 - ($N = 8$)

No Ensaio simulado 2 foram utilizados os seguintes parâmetros para o controlador:

N_1	N	N_u	λ_1	λ_2	λ_3	ϵ	$\Delta S[m]$
1	8	1	2	1	1	$1e-3$	0.04

Tabela 4.7: Parâmetros ensaio simulado 2.

Neste ensaio mudou-se o horizonte de predição ($N = 8$), que passou para 8 amostras, resultando em uma predição de 0.32 segundos por intervalo de amostragem. Os outros parâmetros continuam igual ao ensaio simulado 1. Foram realizadas simulações para os três métodos de otimização no seguimento das duas trajectórias (diagonal e pulso).

Na Tabela 4.8 apresentam-se os resultados do seguimento das trajectórias, Figuras 4.23(a) e 4.24(a), no ensaio 2.

Trajectória diagonal						
Método	Número de iterações			Tempo Optimiz.	MSE(Erro)	
	Máximo	Mínimo	Média	Média(ms)	x, y	θ
G	37	4	13.20	11.19	4.73e-005	5.88e-004
GC_FR	24	4	11.01	9.27	4.59e-005	4.10e-004
GC_PR	18	4	9.98	8.26	4.27e-005	8.23e-004
Trajectória pulso						
Método	Número de iterações			Tempo Optimiz.	MSE(Erro)	
	Máximo	Mínimo	Média	Média(ms)	x, y	θ
G	45	4	18.66	15.84	3.61e-004	0.0016
GC_FR	30	4	13.81	11.89	3.95e-004	0.0024
GC_PR	27	4	11.89	10.34	4.11e-004	0.0026

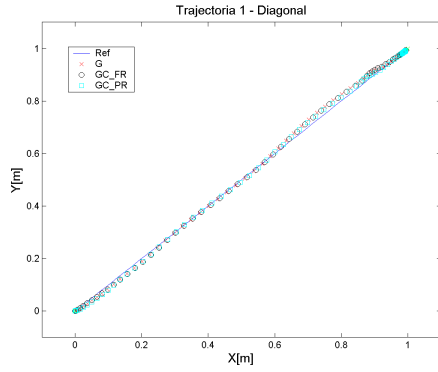
Tabela 4.8: Trajectória diagonal e pulso - ensaio simulado 2 ($N = 8$).

Analisando a Tabela verifica-se que os números de iterações dos algoritmos de optimização aumentaram. Isso tem lógica, pois o tamanho do horizonte predição afecta directamente o valor da função custo. Não foi significativa a diferença entre os erros no seguimento da trajectória para os diferentes métodos de optimização.

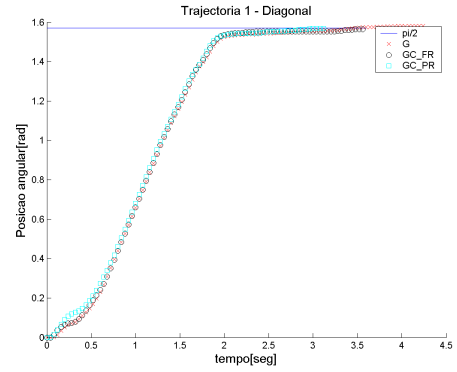
O método GC_PR continua a apresentar uma convergência mais rápida na optimização das trajectórias, além disso em algumas optimizações os métodos

com gradientes conjugados (GC_FR e GC_PR) tendem a uma solução melhor. Por exemplo, na Figura 4.23(d) o valor da função custo tem o mesmo valor inicial para os três métodos, pois trata-se da otimização realizada no primeiro instante de amostragem, e convergem para valores mais baixos nos métodos com gradientes conjugados que conseguiram ultrapassar um patamar existente com derivada quase nula.

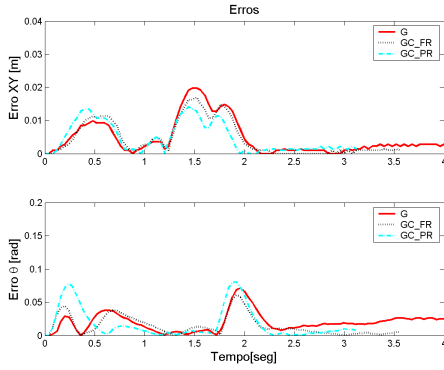
As Figuras 4.23(a) e 4.23(b) apresentam a trajetória (x,y) e a posição angular (θ) realizada em simulação pelo robô, para os três métodos de otimização. A Figura 4.23(c) apresenta os erros no seguimento da trajetória.



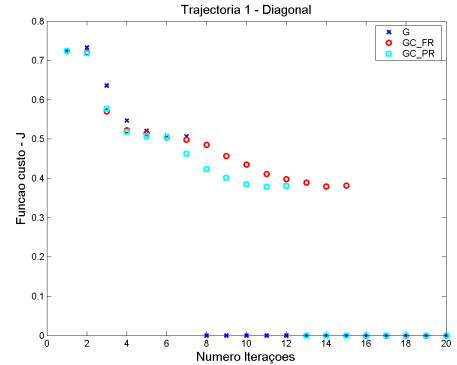
(a) Posições (x,y) .



(b) Posição angular θ .



(c) Erro (x,y,θ) .

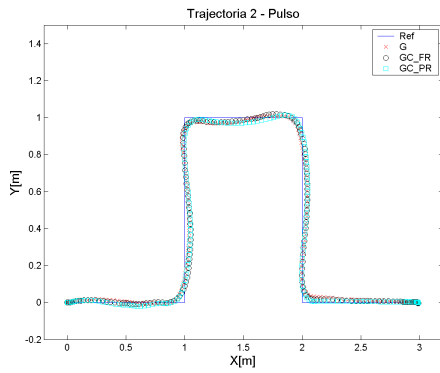


(d) Otimização no instante 1 (0.04 seg).

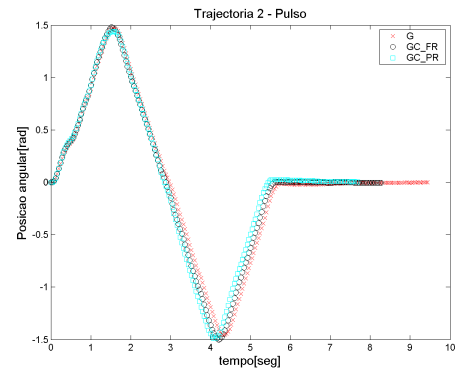
Figura 4.23: Trajetória Diagonal - ensaio simulado 2 ($N = 8$).

Na Figura 4.24 apresenta-se os gráficos para o seguimento da trajetória pulso, para o ensaio 2. As trajetórias realizadas pelo robô com o horizonte de previsão

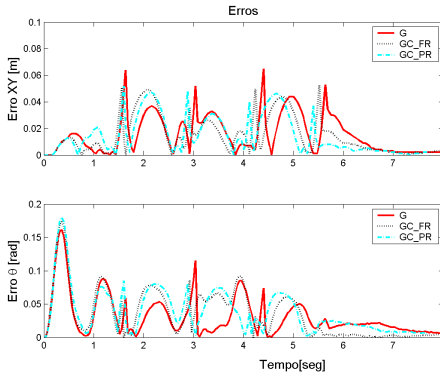
$N = 8$, não parecem tão suaves como as trajetórias com $N = 10$ (ensaio 1), basta comparar as Figuras 4.22(a) e 4.24(a). Outro factor importante é que o número de iterações dos algoritmos de optimização aumentaram com o horizonte de predição menor. Assim embora cada iteração seja ligeiramente mais rápida pois temos que calcular menos saídas futuras, verifica-se que o tempo de cálculo dos sinais de controlo do robô é maior devido a termos mais iterações.



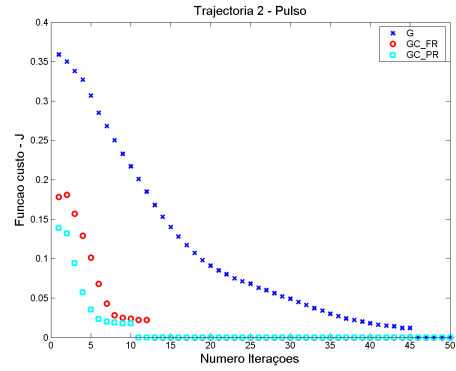
(a) Posições (x,y) .



(b) Posição angular θ .



(c) Erro (x,y,θ) .



(d) Optimização no instante 0.92 segundos.

Figura 4.24: Trajetória Pulso - ensaio simulado 2 ($N = 8$).

4.3.8.3 Ensaio simulado 3 - ($N = 12$)

No Ensaio simulado 3 foram utilizados os seguintes parâmetros para o controlador:

N_1	N	N_u	λ_1	λ_2	λ_3	ϵ	$\Delta S[m]$
1	12	1	2	1	1	$1e-3$	0.04

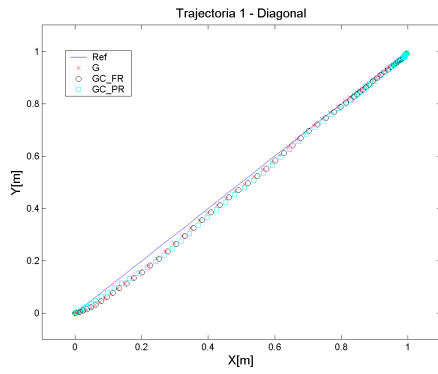
Tabela 4.9: Parâmetros ensaio simulado 3.

Neste ensaio usou-se um horizonte de predição de 12 amostras ($N = 12$), resultando em uma predição de 0.48 segundos por intervalo de amostragem. O objectivo maior deste ensaio é confirmar a tendência do comportamento do controlador: com um horizonte de predição maior, diminui o número de iterações dos algoritmos de optimização, mas aumenta o erro no seguimento das trajectórias.

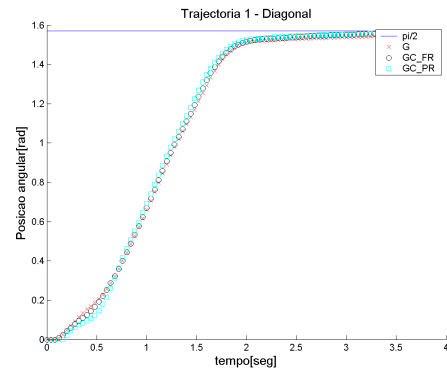
Trajectória diagonal						
Método	Número de iterações			Tempo Optimiz.	MSE(Erro)	
	Máximo	Mínimo	Média	Média(ms)	x, y	θ
G	20	4	10.63	10.51	1.13e-004	0.0010
GC_FR	13	4	8.08	7.90	1.79e-004	0.0013
GC_PR	12	4	6.31	6.20	1.77e-004	0.0017
Trajectória pulso						
Método	Número de iterações			Tempo Optimiz.	MSE(Erro)	
	Máximo	Mínimo	Média	Média(ms)	x, y	θ
G	27	4	13.68	14.20	4.43e-004	0.0035
GC_FR	19	4	9.32	9.51	6.17e-004	0.0029
GC_PR	15	4	7.67	7.90	7.61e-004	0.0034

Tabela 4.10: Trajectória diagonal e pulso - ensaio simulado 3 ($N = 12$).

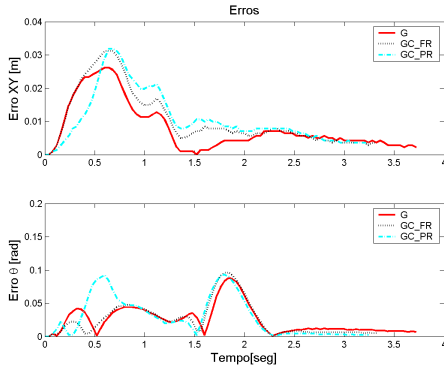
Analisando a Tabela 4.10 e as Figuras 4.25(a) e 4.26(a), confirma-se a tendência, houve uma diminuição no número de iterações dos algoritmos, mas o erro no seguimento das trajectórias aumentou. Com o aumento da predição, o comportamento do robô tende a evitar os pontos de controlo da trajectória de referência, suavizando a trajectória, o que é o esperado em um controlador preditivo. Nas Figuras 4.25(d) e 4.26(d) apresentam-se gráficos com o número de iterações em cada amostragem no seguimento das trajectórias.



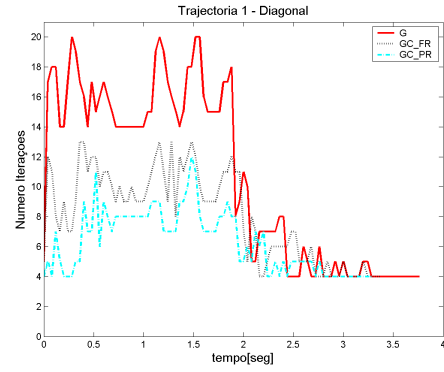
(a) Posições (x,y) .



(b) Posição angular θ .

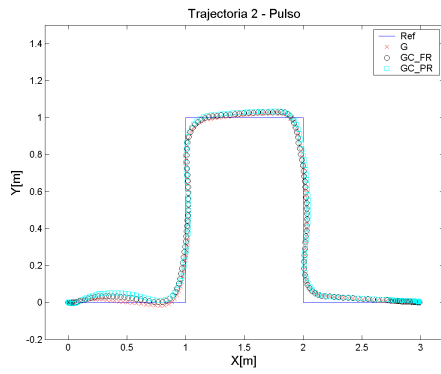


(c) Erro (x,y,θ) .

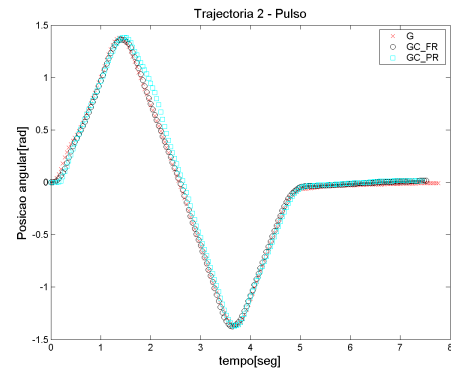


(d) Número de iterações em cada amostragem.

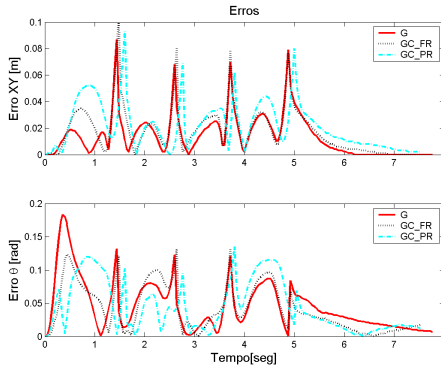
Figura 4.25: Trajetória Diagonal - ensaio simulado 3 ($N = 12$).



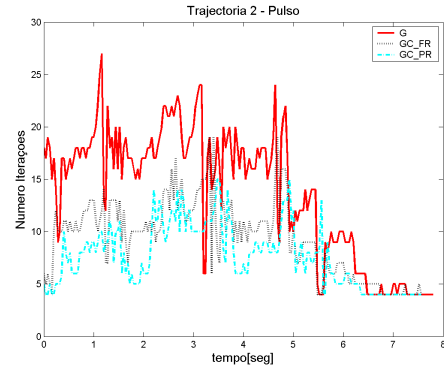
(a) Posições (x,y) .



(b) Posição angular θ .



(c) Erro (x,y,θ) .



(d) Número de iterações em cada amostragem.

Figura 4.26: Trajetória Pulso - ensaio simulado 3 ($N = 12$).

4.3.9 Resultados Reais

A partir dos resultados obtidos com as simulações, partiu-se para o teste do controlador no robô efectivamente. Para realizar os testes com o robô foi necessário inserir uma restrição no tempo utilizado pelo controlador preditivo para calcular os sinais de controle. Foi definido como $20(L_{max_ite} = 20)$ o limite máximo de iterações dos algoritmos de optimização. Isso significa que a optimização da função custo será truncada na iteração 20, caso não tenha encontrado um mínimo. Nas próximas sub-seções serão apresentados dois ensaios realizados pelo robô no seguimento das trajectórias "Diagonal" e "Pulso", Figuras 4.19 e 4.20.

4.3.9.1 Ensaio real 1 - ($N = 10$)

No ensaio real 1 foram utilizados os seguintes parâmetros para o controlador:

N_1	N	N_u	λ_1	λ_2	λ_3	ϵ	$\Delta S[m]$
1	10	1	2	1	1	$1e-3$	0.04

Tabela 4.11: Parâmetros ensaio real 1.

Neste ensaio optou-se por utilizar um horizonte de predição de 10 amostras ($N = 10$), porque se verificou nos resultados de simulação que os optimizadores conseguem uma boa minimização da função custo nestas condições. Os ensaios simulados com um horizonte de predição menor que 10 amostras, mostraram que o número de iterações dos algoritmos de optimização são elevados, considerando a restrição de $L_{max_ite} = 20$.

Na Tabela 4.12 e nas Figuras 4.27 e 4.28, apresentam-se os resultados do seguimento das trajectórias, para os três métodos de optimização.

Para as duas trajectórias, o método dos gradientes conjugados GC_PR é o mais rápido na optimização da função custo, como ocorreu nos ensaios simulados. Este método apresenta erros levemente maiores que os outros métodos no seguimento da trajectória.

A restrição no número máximo de iterações que o optimizador pode realizar, provocou truncamentos na minimização da função custo. Na Tabela 4.13 apresenta-se a quantidade de truncamentos que ocorreram ao longo do seguimento das trajectórias e o número de amostras em que o robô percorreu as trajectórias. O método do gradiente (G) precisa de mais iterações para encontrar um mínimo

Trajectória diagonal						
Método	Número de iterações			Tempo Optimiz.	MSE(Erro)	
	Máximo	Mínimo	Média	Média(ms)	x, y	θ
G	20	4	9.13	7.42	1.02e-004	0.0021
GC_FR	20	4	7.98	6.65	7.78e-005	0.0030
GC_PR	16	4	7.31	6.21	1.68e-004	0.0043
Trajectória pulso						
Método	Número de iterações			Tempo Optimiz.	MSE(Erro)	
	Máximo	Mínimo	Média	Média(ms)	x, y	θ
G	20	4	13.15	11.19	2.45e-004	0.0040
GC_FR	20	4	10.21	8.57	2.94e-004	0.0049
GC_PR	20	4	8.75	7.25	3.17e-004	0.0059

Tabela 4.12: Trajectória diagonal e pulso - ensaio real 1 ($N = 10$).

local, e consequentemente sofreu mais com os truncamentos. Por exemplo, na trajectória diagonal ocorreu 4 truncamentos em 115 possíveis, e na trajectória pulso 48 em 213 possíveis.

Trajectória diagonal		
Método	Número de truncamentos	Número de amostras
G	4	115
GC_FR	1	89
GC_PR	0	69
Trajectória pulso		
Método	Número de truncamentos	Número de amostras
G	48	213
GC_FR	10	196
GC_PR	1	184

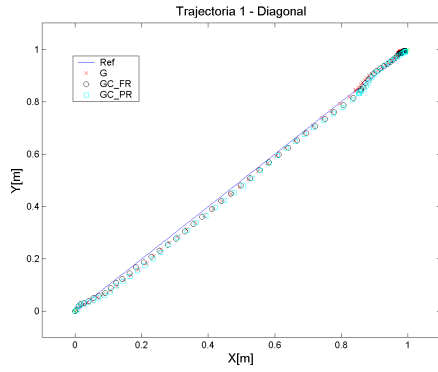
Tabela 4.13: Truncamentos - ensaio real 1.

Na Figura 4.27(d) a optimização utilizando o método do gradiente(G) foi truncada na iteração 20, conforme a restrição no número de iterações. Verifica-se que a minimização da função custo não atingiu um mínimo local, pois precisaria de mais algumas iterações.

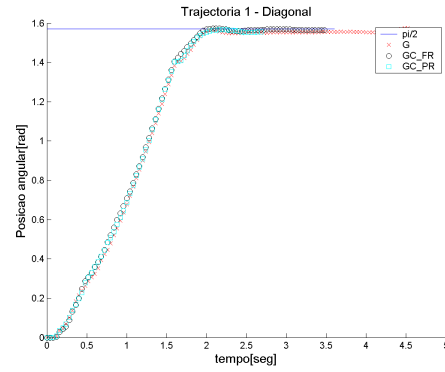
A Figura 4.28(d) apresenta a optimização no único instante em que houve o

4.3 Controlador Preditivo

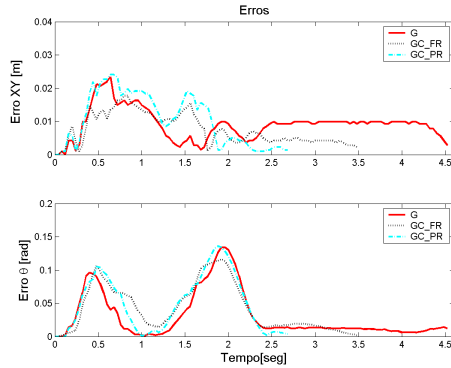
truncamento na utilização do método GC_PR para a trajectória pulso, verifica-se que as últimas iterações pouco acrescentam para a minimização da função custo, isso significa que o truncamento não trouxe prejuízo para o cálculo dos sinais de controlo. Tanto na simulação como nos ensaios reais verificou-se que as primeiras iterações do optimizador possuem uma minimização maior da função custo.



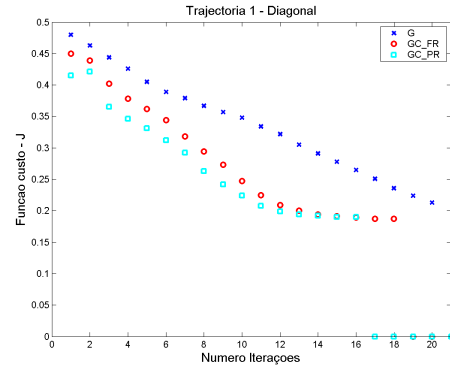
(a) Posições (x, y) .



(b) Posição angular θ .



(c) Erros (x, y, θ) .



(d) Optimização no instante 1.52 segundos.

Figura 4.27: Trajectória diagonal - ensaio real 1 ($N = 10$).

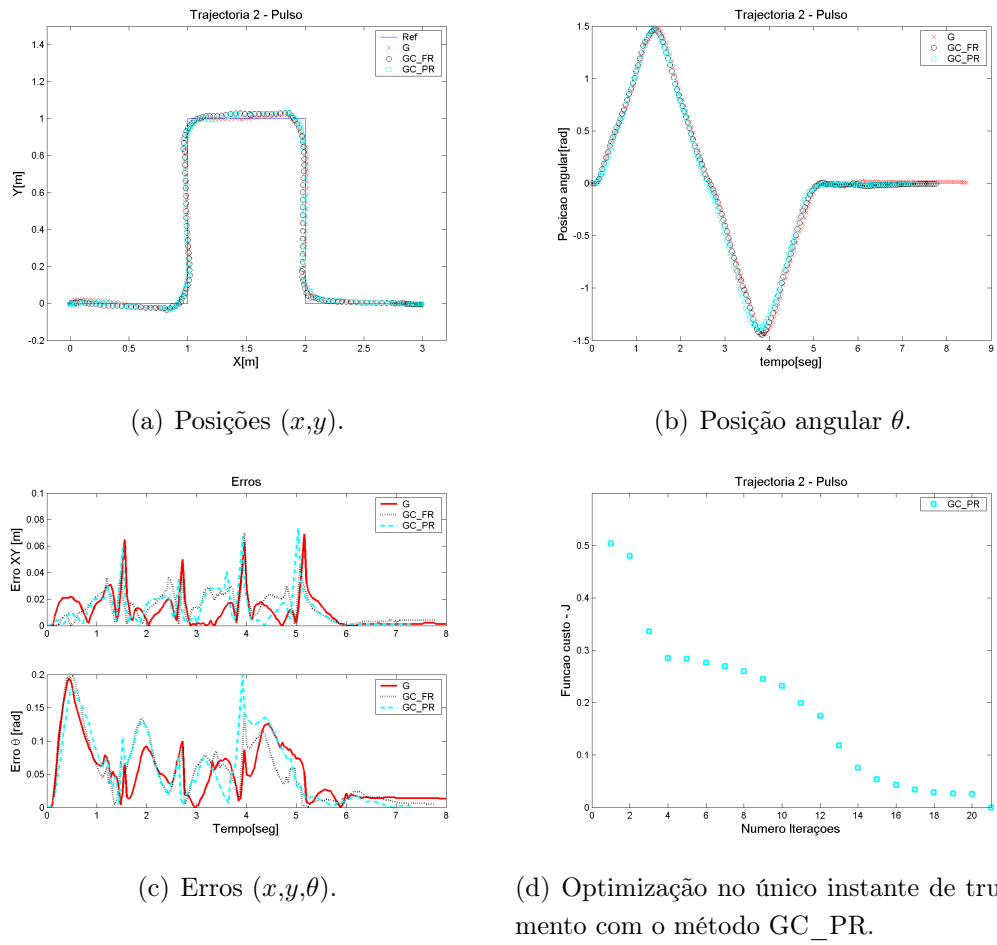


Figura 4.28: Trajetória pulso - ensaio real 1 ($N = 10$).

4.3.9.2 Ensaio real 2 - ($N = 12$)

No ensaio real 2 foram utilizados os seguintes parâmetros para o controlador:

N_1	N	N_u	λ_1	λ_2	λ_3	ϵ	$\Delta S[m]$
1	12	1	2	1	1	$1e-3$	0.04

Tabela 4.14: Parâmetros ensaio real 2.

Na Tabela 4.15 e nas Figuras 4.29 e 4.30, apresentam-se os resultados do seguimento das trajectórias, para os três métodos de optimização.

Neste ensaio optou-se por utilizar um horizonte de predição maior ($N = 12$), para verificar o desempenho do optimizador e o erro no seguimento das trajectórias. Nos ensaios simulados, verificou-se que um horizonte de predição maior que 12, provocaria erros excessivos no seguimento da trajectória, devido a quantidade elevada no número de predições.

Trajectória diagonal						
Método	Número de iterações			Tempo Optimiz.	MSE(Erro)	
	Máximo	Mínimo	Média	Média(ms)	x, y	θ
G	20	4	9.37	8.23	9.59e-005	0.0071
GC_FR	16	4	7.32	6.53	1.58e-004	0.0061
GC_PR	14	4	6.80	6.13	1.51e-004	0.0091
Trajectória pulso						
Método	Número de iterações			Tempo Optimiz.	MSE(Erro)	
	Máximo	Mínimo	Média	Média(ms)	x, y	θ
G	20	4	12.01	10.76	4.17e-004	0.0049
GC_FR	20	4	8.45	7.63	4.83e-004	0.0048
GC_PR	14	4	6.93	6.27	6.24e-004	0.0050

Tabela 4.15: Trajectória diagonal e pulso - ensaio real 2 ($N = 12$).

Houve uma diminuição no número de iterações comparado com o ensaio 1, mas um erro maior no seguimento das trajectórias. Com o aumento do horizonte de predição é provável que o erro no seguimento da trajectória tenha um aumento, principalmente em trajectórias com mudanças bruscas de direcção, como a trajectória pulso. O comportamento do robô tende a antecipar as acções de controlo, evitando os pontos de mudanças bruscas.

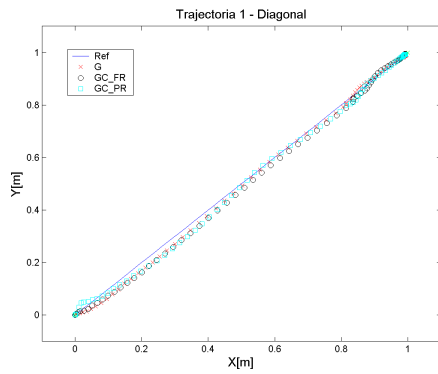
Nas Figuras 4.29(d) e 4.30(d) apresentam-se as velocidades lineares (v, vn) e a velocidade angular (w) do robô no seguimento das trajectórias.

Na Tabela 4.16 apresenta-se a quantidade de truncamentos no seguimento das trajectórias para os métodos de optimização. Verifica-se que o número de truncamentos diminuiu comparado com o ensaio real 1, pois a média de iterações dos algoritmos de optimização também diminuiu.

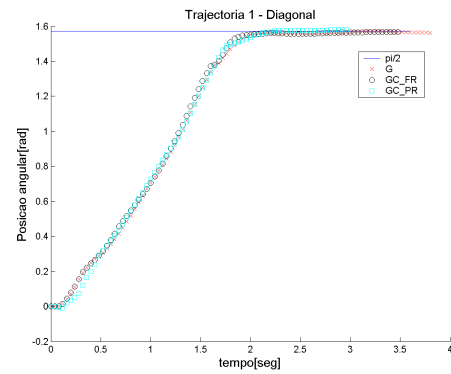
Trajectória diagonal		
Método	Número de truncamentos	Número de amostras
G	2	97
GC_FR	0	89
GC_PR	0	76
Trajectória pulso		
Método	Número de truncamentos	Número de amostras
G	16	189
GC_FR	1	182
GC_PR	0	180

Tabela 4.16: Truncamentos - ensaio real 2.

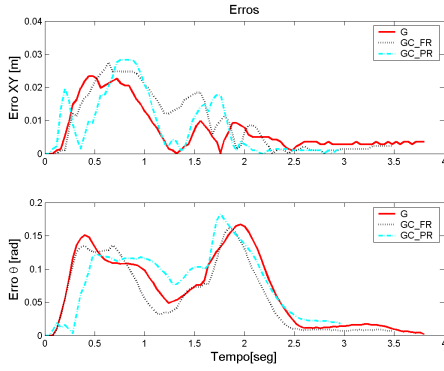
4.3 Controlador Preditivo



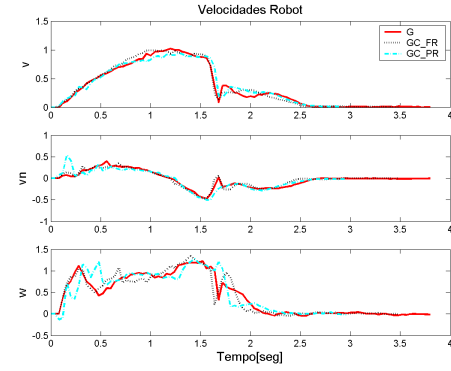
(a) Posições (x,y) .



(b) Posição angular θ .



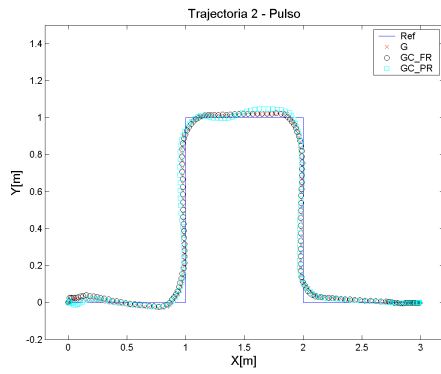
(c) Erros (x,y,θ) .



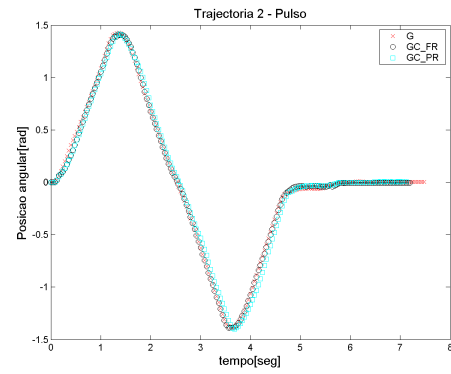
(d) Velocidades do robô.

Figura 4.29: Trajectória diagonal - ensaio real 2.

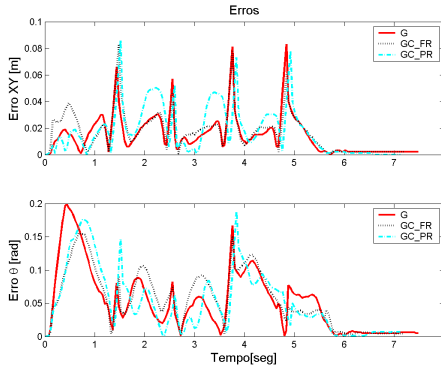
4.3 Controlador Preditivo



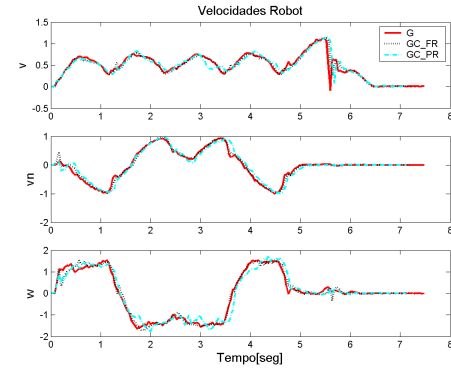
(a) Posições (x,y) .



(b) Posição angular θ .



(c) Erros (x,y,θ) .



(d) Velocidades do robô.

Figura 4.30: Trajetória pulso - ensaio real 2 ($N = 12$).

4.3.10 Conclusões

Analisando os resultados do controlador preditivo não-linear no robô omnidireccional, obtém-se algumas conclusões:

- Os métodos de optimização dos gradientes conjugados são mais rápidos na minimização da função custo, em relação ao método do gradiente.
- A diferença entre os erros no seguimento da trajectória para os diferentes métodos de optimização não foram significativas. Ainda mais tendo-se em conta o efeito de predição no controlo do robô no seguimento de trajectórias com mudanças bruscas de direcção, onde o robô evita estes pontos de mudanças bruscas.
- Os algoritmos de optimização possuem uma contribuição maior para a minimização da função custo nas primeiras iterações e uma contribuição desprezível nas últimas iterações.
- O número de iterações dos algoritmos de optimização aumentam com o horizonte de predição menor, consequentemente aumentando o tempo de cálculo dos sinais de controlo do robô em cada amostragem.
- Um horizonte de predição entre 8 e 12 amostras ($8 \leq N \leq 12$) é um bom compromisso entre o número de iterações dos algoritmos de optimização para minimizar a função custo e o comportamento do robô no seguimento das trajectórias. Com horizontes de predição maiores que 12, os seguimentos das trajectórias apresentam erros excessivos, pois o robô tem a tendência de evitar os pontos de controlo das trajectórias de referência, devido a predição dos instantes futuros. Com um horizonte de predição menor que 8, os algoritmos de optimização necessitam de muitas iterações para minimizar a função custo.
- A estratégia de controlo é aplicável em tempo real.
- Os resultados reais estão concordantes com os simulados. Verificou-se que os resultados de simulação e reais possuem o mesmo comportamento para variações dos horizontes de predição e para os diferentes métodos de optimização.

- Fazendo uma comparação com um controlador reactivo, Secção 4.2, pode-se verificar a diferença no comportamento do robô, conforme a Figura 4.31. O controlador preditivo tem a característica de antecipar as acções de controlo, enquanto o reactivo apresenta uma sobre passagem no seguimento da trajectória, pois as acções de controle acontecem depois de detectar os erros.

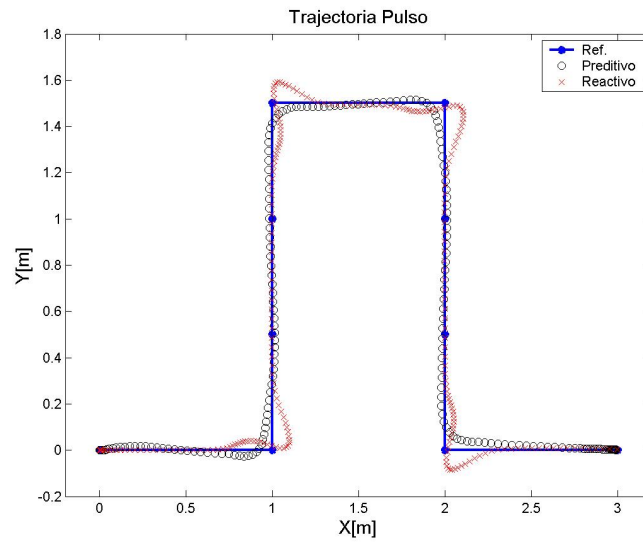


Figura 4.31: Comparação entre o controlador preditivo e o controlador reactivo.

Capítulo 5

Arquitectura de Coordenação e Cooperação para Sistemas Multi-Robô

5.1 Introdução

Sistemas com agentes autónomos em ambientes complexos e dinâmicos requerem mecanismos de cooperação, coordenação e constante reconfiguração. É essencial o desenvolvimento de ferramentas que permitam o funcionamento destes sistemas, por um longo tempo e sem interrupções. A aplicação é muito vasta, basicamente em todos os sistemas em que existam robôs autónomos ou similares (AGVs - Automated Guided Vehicles). A aplicação abrange desde a cooperação em missões de busca e salvamento até a cooperação em linhas de produção e armazéns distribuídos onde diversos AGVs tem que cooperar para executar tarefas de transporte e armazenagem de produtos. Planeamento é um componente essencial para a coordenação de agentes autónomos, enquanto algumas formas de coordenação são puramente reactivas a situações correntes, de forma mais ampla o sucesso da coordenação está na antecipação e como as acções dos agentes autónomos vão afectar o ambiente(Durfee (1996)). Desta forma, uma arquitectura flexível que permita rápida reconfiguração e planeamento das actividades dos agentes envolvidos, além da cooperação entre eles é fundamental.

O futebol robótico é a aplicação utilizada como plataforma de teste e de-

envolvimento para a arquitectura de cooperação, pois trata-se de um ambiente dinâmico onde a coordenação espacial e comportamental entre os robôs torna-se essencial para alcançar os objectivos. Conforme [Kitano *et al.* \(1997a\)](#) o jogo de futebol é um ambiente multi-agente em tempo-real muito atraente, do ponto de vista de inteligência artificial distribuída e pesquisa de multi-agentes. Além disso, como citado no Capítulo 2, a equipe de futebol robótico da Faculdade de Engenharia possui uma larga experiência neste tipo de competições e a necessidade de um ambiente de coordenação e cooperação para os robôs futebolistas tornou-se fundamental.

Na literatura a respeito de cooperação de sistemas multi-robôs aplicado ao futebol robótico pode-se citar algumas abordagens como [Fujii *et al.* \(2005\)](#) que apresenta um método de controle cooperativo tendo em conta o nível de realização do objectivo. O objectivo é definido a partir das necessidades essenciais para realizar uma tarefa. Neste método, o comportamento cooperativo entre os agentes é definido pelo grau de avaliação para alcançar um objectivo e compartilhando o resultado de avaliação entre eles. [Beaudry *et al.* \(2005\)](#) apresenta uma arquitectura constituída principalmente por 5 níveis hierárquicos: Decisão principal, Padrão, Função, Comportamento, Acção. Nesta arquitectura, cada máquina deve ser constituída de um número arbitrário de níveis de nodos de decisão, conduzindo a nodos de comportamento que activam acções específicas para implementar o comportamento desejado. Em [Balch & Arkin \(1998\)](#), uma abordagem baseada em comportamento para a formação dos robôs é apresentada. [Riedmiller & Merke \(2002\)](#) utilizam uma decomposição hierárquica que decompõe a tarefa global em possíveis sub-tarefas. A divisão das tarefas são baseadas em características individuais e ferramentas básicas que habilitam o funcionamento do time ou um nível tático.

Este Capítulo está dividido da seguinte forma: na Secção 5.2 introduz-se uma breve revisão bibliográfica sobre sistemas multi-agentes e a definição de alguns conceitos importantes que serão necessários no decorrer do capítulo. Na Secção 5.3 descreve-se a estrutura da arquitectura de coordenação e cooperação. Na Secção 5.4 apresenta-se a implementação da arquitectura no contexto do futebol robótico. Conclusões e trabalhos futuros são apresentados na Secção 5.5.

5.2 Conceptualização

Esta Secção tem o objectivo de inserir alguns conceitos mais utilizados na literatura a respeito de sistemas multi-robô. Na literatura mais precisamente no campo da Inteligência Artificial, de uma forma mais genérica poderíamos usar o termo “sistemas multi-agente” no lugar de “sistemas multi-robô”, pois um robô autónomo pode ser classificado como um agente. Nesse trabalho foi adoptado o termo “sistemas multi-robô” pelo simples facto que os agentes em questão são os robôs autónomos descritos no Capítulo 2.

Conforme Wooldridge (2002) não existe uma definição universalmente aceite para o termo agente, mas é um consenso geral que a autonomia é uma característica essencial para um agente. A definição de um agente inteligente apresentada por Wooldridge (2002) foi: “*An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.*” Com as seguintes capacidades, conforme Wooldridge & Jennings (1995):

- **Reactividade** $\hookrightarrow$ agentes inteligentes são capazes da percepção do seu ambiente e respondem rapidamente às alterações que nele ocorrem, para satisfazer seus objectivos;
- **Pró-Actividade** $\hookrightarrow$ agentes inteligentes são capazes de exibir comportamento direccionado por objectivos, tomando a iniciativa para satisfazer seus objectivos;
- **Habilidade Social** $\hookrightarrow$ agentes inteligentes são capazes de interagir com outros agentes (e possivelmente com humanos) para satisfazer seus objectivos.

Outras definições de agentes inteligentes encontradas na literatura:

- “*An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.*” Russel & Norvig (1995)
- “Um Agente é um sistema computacional, situado num dado ambiente, que tem a percepção desse ambiente através de sensores, tem capacidade de

decisão, age de forma autónoma nesse ambiente através de actuadores, e possui capacidades de comunicação de alto-nível com outros agentes e/ou humanos, de forma a desempenhar uma dada função para a qual foi projectado." **Reis (2003)**

A respeito de sistemas multi-agente, uma definição clara é apresentada por **Lesser (1995)**: "*In general multiagent systems are computational systems in which several semi-autonomous agents interact or work together to perform some set of tasks or satisfy some set of goals.*". **Stone & Veloso (2000)** apresentam o seguinte conceito: "*Multiagent Systems is the subfield of Artificial Intelligence that aims to provide both principles for construction of complex systems involving multiple agents and mechanisms for coordination of independent agents' behaviors.*" As principais razões para o uso de sistemas multi-agente(SMA) conforme **Stone & Veloso (2000)** são:

- Alguns domínios requerem SMA
- Programação simplificada
- Paralelismo
- Estudo da inteligência
- Robustez
- Distribuição geográfica
- Escalabilidade
- Efectividade no custo

Os agentes podem ser homogéneos ou heterogéneos, e necessitam de uma infra-estrutura que permita cooperação entre eles. A Figura 5.1(de **Wooldridge (2002)**) apresenta uma estrutura típica que compõe um sistema multi-agente.

A comunicação entre os agentes podem ser divididas em duas arquitecturas básicas(**Reis (2003)**,**Huhns & Stephens (1999)**): Directa e Assistida. Na comunicação directa os agentes são responsáveis por sua comunicação com os outros agentes, sem intervenção de outro agente. Na comunicação assistida existe um agente facilitador responsável pela comunicação entre os agentes. A Figura 5.2 apresenta um esquema dos dois tipos de comunicação.

A coordenação de agentes cooperativos é uma propriedade do sistema de agentes para executar alguma actividade em um ambiente compartilhado. A cooperação é a coordenação entre agentes que não são antagónicos, enquanto que a negociação é a coordenação entre agentes competitivos (ou agentes *self-interested*)(**Huhns & Stephens (1999)**). **Jennings (1996)** apresenta o conceito de

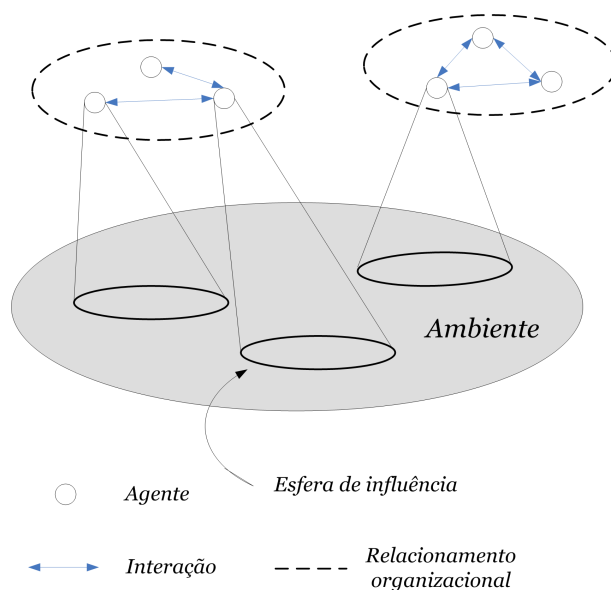


Figura 5.1: Estrutura típica de um sistema multi-agente.

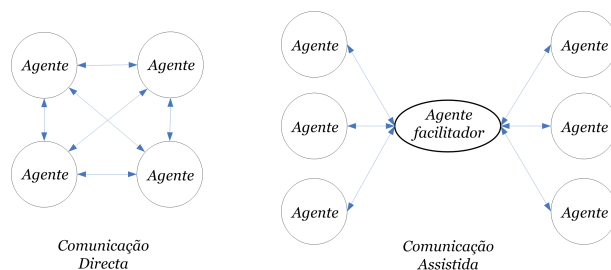


Figura 5.2: Arquitectura de comunicação entre agentes.

coordenação entre agentes cooperativos como sendo o processo pelo qual os agentes raciocinam sobre suas acções locais e as acções dos outros agentes de forma a assegurar que a comunidade actue de maneira coerente. Além disso [Jennings \(1996\)](#) apresenta três razões principais para as acções entre agentes serem coordenadas :

1. Existem dependências entre as acções dos agentes;
2. Existe a necessidade de encontrar restrições globais;
3. Nenhum dos agentes tem competência suficiente, recursos, ou informações para resolver todo o problema.

Conforme Wooldridge (2002), o problema de coordenação está relacionado com administrar inter-dependências entre as actividades dos agentes: algum mecanismo de coordenação é essencial se as actividades que os agentes podem executar interagem de qualquer forma. Existem várias técnicas para coordenação de agentes cooperativos encontradas na literatura. Conforme Wooldridge (2002) algumas das principais abordagens desenvolvidas para coordenação dinâmica de actividades são:

- Planeamento global parcial(PGP) $\leftrightarrow$ Durfee (1996) desenvolveu essa técnica, na qual denominou planeamento global parcial. O principal princípio é a partilha de informações na cooperação dos agentes de forma a atingir conclusões comuns sobre a resolução dos problemas. O planeamento é parcial porque o sistema não gera um plano para todo o problema, mas é global porque os agentes formam planos não-locais a partir da partilha de planos locais para atingir soluções para todo o problema.
- Intenções conjuntas $\leftrightarrow$ esta abordagem está relacionada com técnicas de coordenação de humanos. Para além do objectivo global da equipa, partilhado por todos os agentes, surgem os objectivos de cooperação com os outros agentes da equipa. Estes objectivos consistem em informar os restantes elementos: para atingir os objectivos globais, da impossibilidade de realizar os objectivos globais ou da não viabilidade em atingir os objectivos globais.
- Modelização mútua $\leftrightarrow$ esta abordagem também é relacionada com técnicas de coordenação de humanos. Esta ideia foi primeiramente apresentada em Genesereth *et al.* (1986), abordada como “*cooperation without communication*”. A ideia foi: caso os agentes que interagem dividem o mesmo ponto de vista de um cenário, então pode-se prever qual a acção racional que cada agente deve ter.
- Normas e leis sociais $\leftrightarrow$ é baseado nas técnicas dos humanos para coordenar actividades, e uma das mais importantes é o uso de normas e leis sociais. As normas e leis sociais podem existir em sistemas multi-agente de duas formas:

- Projectadas *off-line* $\mapsto$ nesta abordagem as leis sociais são projectadas off-line e implantadas nos agentes.
- Emergentes $\mapsto$ nesta abordagem as leis emergem do próprio sistema com a participação do grupo de agentes.
- Planeamento multi-agente e sincronização $\leftrightarrow$ o planeamento multi agente deve ter em consideração o facto que as actividades dos agentes podem interferir umas com as outras. Suas actividades devem ser coordenadas. Conforme [Durfee \(1999\)](#), existem três possibilidades para o planeamento multi-agente:
 - Planeamento centralizado para planos distribuídos $\mapsto$ um sistema centralizado de planeamento desenvolve um plano para um grupo de agentes, na qual a divisão e ordenação das tarefas é definida. O agente “*master*” distribui os planos para os agentes “*slaves*”, que executam sua parte do plano.
 - Planeamento distribuído $\mapsto$ um grupo de agentes coopera para formar um plano centralizado. Tipicamente, os agentes serão especialistas em diferentes aspectos. Entretanto, os agentes que formam os planos não serão os mesmos que executarão as tarefas.
 - Planeamento distribuído para planos distribuídos $\mapsto$ um grupo de agentes coopera para formar planos individuais de acções, coordenando dinamicamente suas actividades ao longo do processo. Os agentes podem ser competitivos, podendo existir conflitos que serão resolvidos através de negociações.

Conforme [Jennings \(1996\)](#), três dos mecanismos mais comuns para administrar o processo de coordenação em Inteligência Artificial Distribuída(IAD) são:

- Estrutura Organizacional $\leftrightarrow$ No contexto de sistemas IAD uma estrutura organizacional pode ser vista como um padrão de informação e relações de controlo entre os agentes. Essas relações de controlo são responsáveis por definir a autoridade relativa dos agentes e formatar os tipos de interacção social que possam ocorrer. Outros autores utilizam terminologias diferentes. [Singh \(1990\)](#), [Reis \(2003\)](#), [Conceição et al. \(2006b\)](#) a noção de “estratégia” é

empregada para designar a especificação do comportamento global de um agente ou grupo de agentes. Em [Werner \(1989\)](#), [Reis \(2003\)](#), [Conceição *et al.* \(2006b\)](#) o termo “papéis” é usado para designar o comportamento de um agente individual.

- Troca de informação nível-meta (*meta-level*) $\hookrightarrow$ nesta técnica os agentes enviam entre si informações sobre o nível de controlo de suas prioridades correntes.
- Planeamento multi-agente $\hookrightarrow$ com esta abordagem os agentes geralmente formam um plano na qual especificam suas acções futuras e interacções para atingir um objectivo particular. Planos multi-agente são tipicamente construídos para evitar inconsistências ou conflitos nas acções dos agentes. Existem dois tipos básicos para planeamento multi-agente:
 - Centralizado $\mapsto$ os planos(papéis) de cada agente são desenvolvidos separadamente e um coordenador central identifica as potenciais cooperações de modo a evitar possíveis conflitos. Finalmente os objectivos são divididos em planos individuais na qual os agentes podem executar as suas tarefas de forma sincronizada.
 - Distribuído $\mapsto$ os planos são desenvolvidos por vários agentes, pode acontecer que nenhum agente possua uma visão global das actividades da comunidade, e consequentemente podem detectar e solucionar interacções indesejáveis tornando significativamente mais difícil a resolução de problemas.

5.3 Descrição da Arquitectura

Nesta Secção descreve-se a arquitectura de coordenação e cooperação para coordenar robôs autónomos na execução de tarefas cooperativas. A arquitectura é baseada na definição de uma estratégia que utiliza diferentes tácticas, operações coordenadas, papéis, tarefas e acções. Os papéis permitem a configuração de um comportamento individual pela execução de tarefas específicas usando um determinado conjunto de acções. A comunicação entre os robôs é do tipo assistida, onde um agente facilitador (Supervisor) administra as informações entre os

robôs. Esta arquitectura foi baseada nas linhas de pesquisa sobre Estrutura Organizacional aplicada na coordenação em Inteligência Artificial Distribuída (Singh (1990), Reis (2003), Conceição *et al.* (2006b)) e Planeamento multi-agente com o planeamento centralizado (Durfee (1999) e Jennings (1996)). Com esta configuração de arquitectura busca-se flexibilidade e eficiência na operação de sistemas multi-robôs em ambientes dinâmicos. Além disso a arquitectura pode ser utilizada como uma plataforma base para aplicação e desenvolvimento de pesquisa em Inteligência Artificial, como um ambiente de testes para diferentes algoritmos nos seus diversos ramos de pesquisa.

A arquitectura de controlo e cooperação está dividida em dois níveis principais (Supervisor e Decisão):

1. Supervisor (Agente facilitador) $\hookrightarrow$ é a central de controlo, responsável pelo planeamento e resolução dos problemas. Possui dois níveis hierárquicos:
 - Estratégia e táticas $\mapsto$ são os modos de operação;
 - Operações coordenadas $\mapsto$ são as tarefas cooperativas;
2. Decisão $\hookrightarrow$ são os níveis implementados nos robôs, está relacionado com a execução das tarefas. Está dividido em três níveis hierárquicos:
 - Papéis $\mapsto$ funções atribuídas a cada robô;
 - Tarefas $\mapsto$ tarefas desempenhadas a partir de um determinado papel;
 - Acções $\mapsto$ acções de controlo de baixo nível para desempenhar as tarefas.

Para exemplificar de forma mais prática o funcionamento da arquitectura, pode-se usar o contexto de AGVs de transporte em armazéns de armazenamento. Desta forma, para cada nível hierárquico tem-se o seguinte planeamento:

- Estratégia e táticas $\mapsto$ Modo de operação seguro, económico ou normal;
- Operações coordenadas $\mapsto$ tarefas cooperativas como as inicializações dos robôs ou na cooperação para transportar e empilhar produtos;
- Papéis $\mapsto$ funções de cada robô: AGV em transporte, AGV em carga, AGV ocioso, etc;

- Tarefas $\mapsto$ tarefas como transportar para porta A, ir para porta B, etc.;
- Acções $\mapsto$ acções para desempenhar as tarefas: desvio de obstáculos, planeamento e geração de trajectórias, controladores de trajectórias, accionamento dos actuadores.

Na Figura 5.3 apresenta-se um diagrama comportamental-hierárquico dos agentes no ambiente de cooperação.

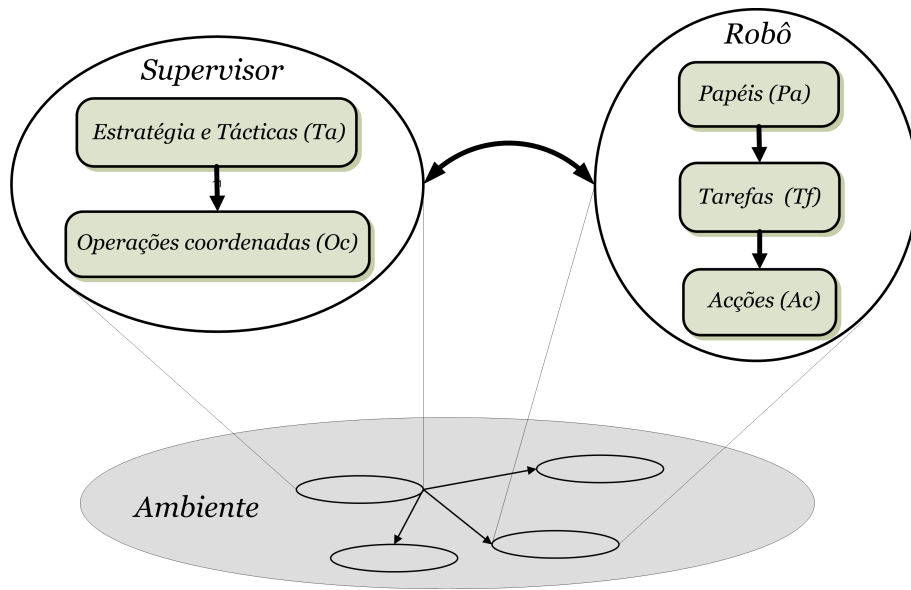


Figura 5.3: Diagrama comportamental-hierárquico da arquitectura.

5.3.1 Estratégia e Tácticas, Operações Coordenadas

A coordenação estratégica é composta por um conjunto de tácticas que são responsáveis pelo comportamento do sistema multi-robôs. O Supervisor é um agente privilegiado, pois é o ponto de referência do sistema. O Supervisor actua como um agente centralizador, que decide qual a melhor táctica para cada situação, a partir de um conjunto de regras de activação.

Estas regras de activação dependem da aplicação em questão, por exemplo no futebol robótico, o supervisor tem a função de decidir a estratégia da equipa, que por sua vez é composta por um conjunto de tácticas. As tácticas (Ta) possuem

um conjunto de operações coordenadas(Oc), que definem o comportamento e posicionamento dos robôs a partir das situações no jogo. Geralmente são atribuídas Ocs a situações do jogo como início de jogo, chutos de canto, laterais e etc.

Por exemplo em uma situação de pontapé de baliza, a uma determinada tática pode usar a Oc_1 (por exemplo um remate directo a baliza) e outra tática pode ser mais cautelosa usando a Oc_2 (por exemplo um passe ao companheiro de equipa). Outra forma bastante usual para a escolha de táticas no futebol está baseada no resultado actual do jogo, se a equipa está em desvantagem no marcador e o jogo está no fim, opta-se por uma tática mais ofensiva no decorrer normal do jogo, como por exemplo usar uma Oc_n que atribui papéis de atacantes para todos os robôs. No caso onde não há Ocs activas, ou no caso de falha na comunicação entre os robôs e o supervisor, os robôs se comportam baseados em seus próprios papéis pré-definidos.

5.3.2 Papéis, Tarefas e Acções

Os papéis, tarefas e acções formam os níveis hierárquicos de coordenação dos robôs. Os papéis (Pa) são as funções desempenhadas pelos robôs, essas funções são atribuídas a partir de cada Operação coordenada(Oc). As funções que cada robô assume são delegadas pelo Supervisor conforme um conjunto de condições, como por exemplo: suas características físicas, a partir de sua posição no ambiente, etc. A partir da definição de um papel, são atribuídas tarefas(Tf) para o robô, que são executadas por um conjunto de acções. As acções(Ac) são responsáveis pelo controle à nível mais baixo, como por exemplo, seguimento de uma trajectória, desvio de um obstáculo, controlo de um actuador, etc. De uma forma resumida pode-se dizer que os papéis definem o comportamento dos robôs através da atribuição de tarefas, que são executadas por um conjunto de acções.

Por exemplo no domínio do futebol robótico, é necessário fornecer aos robôs informações relativas ao seu posicionamento e comportamento, tendo-se em conta a situação actual do jogo. Essas informações são passadas aos jogadores pelo Supervisor que possui as informações de posicionamento dos robôs. Para exemplificar vamos considerar a situação do jogo em que temos um lateral a favor da equipa. A equipa deve assumir uma operação coordenada que executa um lateral. Nesta Oc , cada jogador possui um papel, que por sua vez executará tarefas,

5.4 Implementação da Arquitectura Aplicada ao Domínio do Futebol Robótico

como por exemplo passar a bola ou receber a bola de um companheiro de equipa. As acções são as ferramentas necessárias para executar a tarefa, como efectuar o passe, ir até a bola ou posicionar-se para receber o passe. A execução de uma determinada operação coordenada pode ser abortada, isso acontece no caso de algo correr errado. Esta decisão pode ser baseada em um conjunto de regras, como por exemplo:

- a bola não chega ao companheiro de equipa;
- o oponente intercepta a bola;
- o tempo de execução da *Oc* termina.

5.4 Implementação da Arquitectura Aplicada ao Domínio do Futebol Robótico

A arquitectura de coordenação e cooperação está dividida em dois módulos principais(Supervisor e Decisão), como explicado na Secção 5.3. Desta forma, para implementar a arquitectura optou-se pelo desenvolvimento duas aplicações informáticas, também denominados Supervisor e Decisão. Na Figura 5.4 apresenta-se um fluxograma do funcionamento dos ciclos de controlo da arquitectura. O fluxograma possui três módulos principais:

1. Referee box $\leftrightarrow$ é uma aplicação informática usada para auxiliar o juiz do jogo(humano) a controlar o jogo de futebol robótico. São enviados comandos por uma rede local a cada time. Os comando são diferentes para cada situação do jogo: início do jogo (k ou K, depende do time que vai sair com a bola), lateral(t ou T, depende para qual time foi o lateral), parar o jogo(S), etc.
2. Supervisor - *Loop Principal* $\leftrightarrow$ o *loop* de controlo principal do Supervisor é definido por cinco níveis principais:
 - Define Tácticas $\mapsto$ a partir de informações da situação do jogo define-se qual táctica a equipa vai adoptar, por exemplo: normal, ataque ou defesa.

5.4 Implementação da Arquitectura Aplicada ao Domínio do Futebol Robótico

- Define Operação Coordenada $\mapsto$ a partir das informações recebidas dos robôs por uma rede local sem fio(*wireless*), da tática adoptada e do comando da Referee box é possível definir a Operação coordenada mais apropriada. Geralmente para cada situação do jogo (lateral, início do jogo,...) existe uma Operação coordenada correspondente.
 - Mensagem *broadcast* $\mapsto$ nesta etapa são enviados aos robôs os papéis que devem assumir, assim como informações importantes como por exemplo a posição da bola no campo (por vezes o robô não está a ver a bola).
 - Exibe informações $\mapsto$ são exibidas as informações recebidas dos robôs, táticas e operações coordenadas adoptadas, posições da bola(posição que cada robô estima), comandos recebidos da Referee box e outras informações necessárias para supervisionar o comportamento da equipa.
3. Decisão - *Loop* Principal $\leftrightarrow$ o ciclo de controlo principal é composto por quatro etapas:
- Lê sensores $\mapsto$ nesta etapa as informações dos sensores são processadas, obtendo a estimação da posição do robô e da bola.
 - Controlo central $\mapsto$ a partir da definição do papel do robô, são executadas as tarefas e acções. Os controladores de trajectórias, algoritmos de desvio de obstáculos, accionamento do sistema de chuto estão implementados neste nível.
 - Mensagem *broadcast* $\mapsto$ são enviados aos robôs e ao supervisor informações do robô(posição no ambiente, posição da bola, trajectória em execução, ...).
 - Exibe informações $\mapsto$ são exibidas os principais *logs* do robô (posição, velocidades, posição da bola, papel actual, ...).

Nas próximas duas sub-secções(5.4.1 e 5.4.2) apresentam-se algumas características dos *softwares* desenvolvidos.

5.4 Implementação da Arquitectura Aplicada ao Domínio do Futebol Robótico

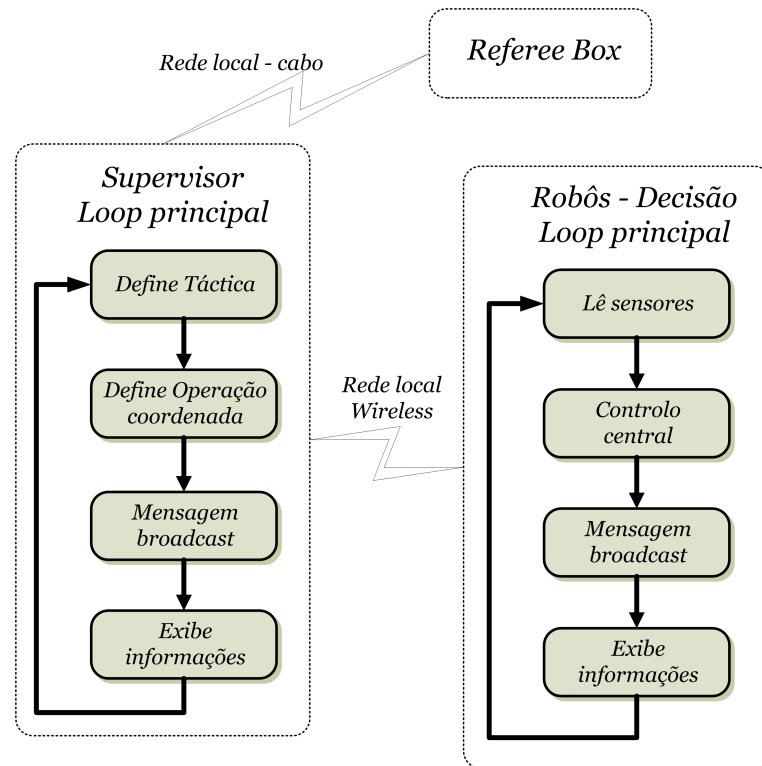


Figura 5.4: Fluxograma do *Loop* de controlo da arquitectura.

5.4.1 Supervisor

O software “Supervisor” é responsável pela coordenação e cooperação da equipa de robôs, através dos módulos “Estratégia e táticas” e “Operações coordenadas”. Além disso, ainda possui as habilidades de comunicação com o ambiente externo para receber as requisições para tarefas dos robôs. Na Figura 5.5 apresenta-se aplicação “Supervisor” e algumas das suas características são listadas abaixo, conforme a numeração na figura:

1. Informações dos robôs $\leftrightarrow$ exibe informações actuais dos robôs como papéis, tarefas e acções.
2. Ambiente $\leftrightarrow$ exibe a posição dos robôs no ambiente, neste caso o campo de futebol. Também mostra leituras dos sensores de cada robô, como por exemplo a posição da bola identificada por cada robô.
3. Central de controlo $\leftrightarrow$ são os comandos para ligar/desligar a comunicação

5.4 Implementação da Arquitectura Aplicada ao Domínio do Futebol Robótico

com os robôs, assim como inserir ou retirar os robôs da equipa.

4. Comandos Refbox $\hookrightarrow$ exhibe os comandos recebidos pela Referee Box, ou em caso de testes simula os comandos de uma Referee Box.
5. Tácticas $\hookrightarrow$ exhibe a táctica actual.
6. Operações Coordenadas $\hookrightarrow$ exhibe a operação coordenada actual.
7. Papéis $\hookrightarrow$ exhibe o papel actual de cada robô.
8. Configurações gerais $\hookrightarrow$ engloba configurações de rede, comandos de configuração para os robôs, parametrização do ambiente, entre outros.



Figura 5.5: Software "Supervisor".

5.4.2 Decisão

O *software* “Decisão” é responsável pelo comportamento do robô, nele estão implementadas as camadas “Papéis”, “Tarefas” e “Acções”. Na Figura 5.6 apresenta-se a aplicação “Decisão” e algumas das suas características são listadas abaixo, conforme a numeração na figura:

1. Informações dos robôs $\leftrightarrow$ exibe informações de *logs* do robô: posição no ambiente, velocidades, imagens processadas (bola, balizas, obstáculos,...), entre outras.
2. Ambiente $\leftrightarrow$ exibe o robô no ambiente, neste caso o campo de futebol. Também mostra leituras dos sensores de cada robô, como por exemplo a posição da bola identificada por cada robô.
3. Central de controlo $\leftrightarrow$ são os comandos para ligar/desligar o robô.
4. Comandos $\leftrightarrow$ comandos para teste do robô, como referências de velocidades e posição.
5. Controlo baixo nível $\leftrightarrow$ módulo que comunica com o microcontrolador do robô, configura parâmetros dos controladores de baixo nível (PID).
6. Controlo alto nível $\leftrightarrow$ módulo que define o estado do robô: se o robô está em teste, simulação, em funcionamento a esperar comandos do Supervisor, etc.
7. Papéis $\leftrightarrow$ exibe o papel actual do robô.
8. Configurações gerais $\leftrightarrow$ módulo que permite configuração dos sistemas de localização, configuração dos actuadores, parametrização de controladores, entre outros.

5.4 Implementação da Arquitectura Aplicada ao Domínio do Futebol Robótico

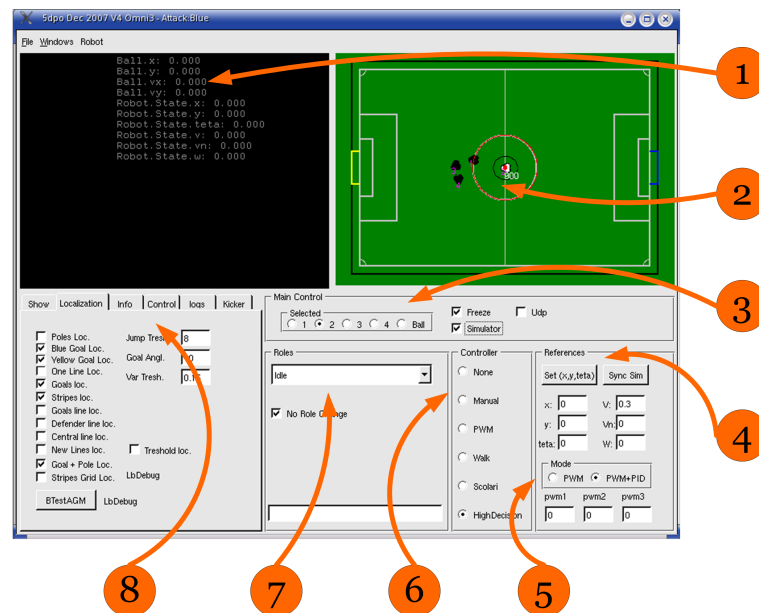


Figura 5.6: Software "Decisão".

5.5 Conclusões

Este Capítulo apresentou uma arquitectura que permite a cooperação de sistemas multi-robô na execução de tarefas complexas em ambientes dinâmicos. A arquitectura proposta vem sendo utilizada na liga *middle-size* de futebol robótico da Universidade do Porto, e uma estrutura similar desta arquitectura nas ligas *small-size*, *Sony 4legged*, *simulation 2D*, *simulation 3D* e *rescue simulation*, revelando-se uma estrutura muito prática no sentido de permitir a cooperação entre os robôs de uma forma sistemática e estruturada. Outra aplicação em desenvolvimento da arquitectura são os robôs móveis industriais (AGVs), na qual é responsável pela coordenação do transporte de material entre estações de trabalho e sistemas automáticos de armazenamento.

Capítulo 6

Conclusões e Trabalhos Futuros

6.1 Síntese e Conclusões do Trabalho

O objectivo principal deste trabalho foi o estudo de estratégias para o controlo, coordenação e cooperação de um sistema multi-robôs. Esse estudo foi focado em robôs móveis com locomoção do tipo omnidireccional, e seguiu uma linha de pesquisa que abordou a modelização, o controlo individual e a coordenação e cooperação dos robôs, respectivamente. A tese foi estruturada de modo a apresentar no final de cada Capítulo as conclusões dos resultados obtidos de forma detalhada. Sendo assim, num contexto geral apresenta-se a seguir uma síntese do trabalho e suas principais conclusões:

- Desenvolvimento de um modelo completo para os robôs. O modelo engloba a motorização, a dinâmica e a cinemática dos robôs. O modelo estimado representa não-linearidades como: limitações de corrente e de tensão nos motores, e a acção dos atritos na dinâmica dos robôs. Três métodos para estimar os parâmetros das equações da dinâmica do robô foram desenvolvidos, usou-se técnicas baseadas em estimadores mínimos quadrados e sua variante variável instrumental.

O modelo estimado foi essencial na calibração inicial dos controladores PID das rodas e no projecto dos controladores de trajectórias dos robôs. Os resultados de simulação e também a comparação com um modelo linear evidenciam a eficiência do uso de um modelo não-linear para representar de forma realista o comportamento dos robôs;

- Desenvolvimento de duas estratégias de controlo para o seguimento de trajectórias dos robôs móveis omnidireccionais: controlador reactivo e controlador preditivo.

A primeira estratégia utiliza a realimentação parcial do estado do robô e um algoritmo para mapear as restrições das velocidades de movimentação do robô, através do modelo cinemático do robô e dos limites de velocidade dos motores. Buscou-se com esta estratégia um controlador flexível com baixo tempo computacional.

A segunda estratégia aborda o conceito de antecipação ou predição, ideal para este tipo de aplicação onde são conhecidas as referências futuras, que são as trajectórias a seguir.

Analisando os resultados com os controladores pode-se verificar que o controlador preditivo tem a característica de antecipar as acções de controlo, evitando erros demasiados em situações de mudanças bruscas de direcção e orientação nas trajectórias. O mesmo desempenho não foi obtido pelo controlador reactivo, o que é natural devido à sua acção reactiva, baseada no erro do estado do robô. Outro ponto de análise é o tempo computacional do cálculo dos sinais de controlo, essencial em aplicações tempo-real. O controlador reactivo obteve óptimos tempos de cálculo, até porque foi projectado com esta característica. Levando-se em conta o compromisso entre tempo computacional e desempenho, o controlador preditivo obteve melhores resultados que o controlador reactivo, considerando a aplicação em questão que é o seguimento de trajectórias.

- Desenvolvimento de uma arquitectura para a coordenação e a cooperação de sistemas multi-robô. Buscou-se uma arquitectura flexível que permita rápida reconfiguração e planeamento das tarefas dos robôs em ambientes dinâmicos. O futebol robótico foi a aplicação utilizada como plataforma de teste e desenvolvimento, pois trata-se de um ambiente dinâmico onde a coordenação, cooperação e controlo individual dos robôs é essencial para alcançar os objectivos. A arquitectura vem sendo utilizada na equipa da liga *Middle-Size* de futebol robótico da Universidade do Porto em competições da iniciativa RoboCup. Outra aplicação que utiliza esta arquitectura é um AGV de carga também desenvolvido na Universidade. A definição de

papéis permite a configuração de um comportamento individual do robô, desempenhando tarefas através de um conjunto de acções. Com uma estrutura de comunicação assistida, uma central(Supervisor) administra os pedidos de transporte por um aplicação web, e delega as funções para o robô.

6.2 Trabalhos Futuros

A pesquisa realizada para este trabalho proporcionou novas perspectivas de desenvolvimento:

- Aplicação das técnicas de modelização e das técnicas de controlo nas novas equipas(versão 2007/2008) de robôs omnidireccionais da equipa 5dpo-2000 da Universidade do Porto;
- Desenvolvimento de técnicas de modelização em tempo-real, muito útil em aplicações onde robôs tem que navegar em terrenos irregulares;
- Estudo de algoritmos de controlo de tracção dos robôs baseados em características dinâmicas do robô e características do terreno de navegação;
- A implementação e teste de novas técnicas de optimização numérica não-linear na estrutura do controlador preditivo;
- A pesquisa de novas abordagens da função custo utilizada no controlador preditivo, inserindo componentes relacionadas ao desvio de obstáculos dinâmicos e obstáculos estáticos;
- Aplicação da arquitectura de coordenação e cooperação do sistema multi-robôs em ambientes industriais;
- O desenvolvimento de pesquisa em Inteligência Artificial(AI) utilizando a arquitectura de coordenação e cooperação do sistema multi-robôs como estrutura base.

6.3 Considerações Finais

Num âmbito geral, o objectivo final do trabalho foi a execução de tarefas cooperativas em ambientes dinâmicos pelos robôs. O trabalho foi planeado em 3 etapas principais: inicialmente obteve-se informações importantes a respeito das características cinemáticas e dinâmicas do robô para a estimação de um modelo mais realista possível. Após obter um bom modelo dos robôs projectou-se os controladores para o seguimento de trajectórias, com uma vasta flexibilidade de calibração para várias situações distintas. Desta forma, finalizou-se uma parcela importante na execução das tarefas cooperativas, que é o controlo individual dos robôs. Por fim, com simuladores e controladores afinados partiu-se para o projecto de uma arquitectura de coordenação e cooperação com flexibilidade de reconfiguração, devido à exigência que os ambientes dinâmicos requerem.

Todas as etapas projectadas do trabalho foram finalizadas, o sucesso em cada etapa foi essencial para o desenvolvimento da etapa seguinte. Um ponto que é importante realçar, é que todas as etapas do trabalho foram testadas em situações reais de operação, ou de forma experimental ou na real aplicação para o qual foram projectadas. O trabalho não se limitou a simulações, mesmo que estas tenham sido essenciais para o projecto, testes e implementação final das técnicas e algoritmos de modelização e controlo.

Robótica é uma área multidisciplinar, sendo necessário a aplicação de um vasto conjunto de conceitos e técnicas para alcançar os objectivos propostos. Este trabalho disponibiliza uma importante contribuição através de um conjunto de técnicas e algoritmos, testadas e implementadas, que poderão ser usadas em problemas relacionados a robótica.

Anexo A

Locomoção de robôs móveis com rodas

Neste Anexo são apresentadas as principais configurações a respeito do projecto mecânico de locomoção dos robôs móveis com rodas(Braunl (2003)).

A.1 Omnidireccionais

Os robôs omnidireccionais ou holonômicos são os que possuem a habilidade de se mover em qualquer direcção em um plano 2D, independentemente da orientação do robô. Os robôs denominados não-holonômicos não são capazes de se movimentar em todas direcções. A maioria deles não se movimentam na direcção perpendicular de suas rodas de tracção. Sendo assim o que permite a um robô ser omnidireccional são suas rodas, Figura A.1. Existem vários projectos mecânicos para rodas omnidireccionais, nos projectos mais recentes cada roda é coberta com várias “rodinhas” que rodam livremente, permitindo que roda se movimente em todas direcções.

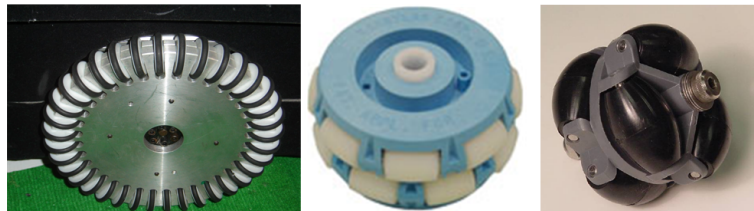


Figura A.1: Rodas omnidireccionais.

A.2 Tracção simples

Conceptualmente, este tipo de configuração é o mais simples para robôs móveis, pois necessita apenas de um conjunto de roda com tracção e um servo de posição para direccionar a roda. O robô ainda necessita de duas rodas passivas, sendo que três pontos de contacto são necessários. A Figura A.2 apresenta a configuração da roda para diferentes direcções.

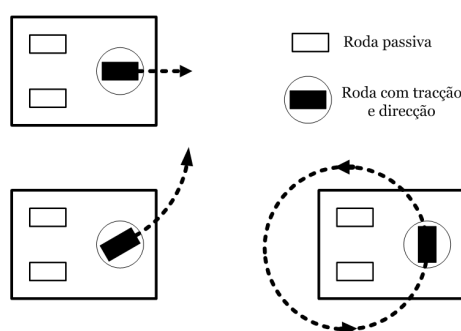


Figura A.2: Tracção simples.

A.3 Tracção diferencial

Robôs com tracção diferencial tem duas rodas fixadas nos lados esquerdo e direito do robô, com tracção individual em cada roda. Sendo que três pontos de apoio são necessários, este tipo de projecto necessita mais uma ou duas rodas passivas adicionais. Este tipo de tracção é mais simples mecanicamente que a tracção simples, pois não necessita de um servo de posição nas rodas de tracção. A tracção diferencial com uma roda passiva não pode ter as rodas de tracção no centro do robô, por razões de estabilidade. Sendo assim, não é possível um movimento de rotação sobre o centro do robô. Com duas rodas passivas, já é permitido realizar esse movimento, mas introduz um problema de contacto na superfície, devido as seus quatro pontos de contacto. A Figura A.3 mostra exemplos de movimentação da tracção diferencial.

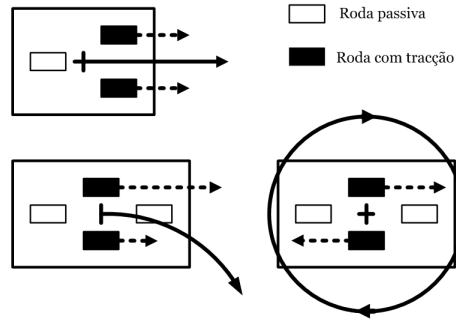


Figura A.3: Tracção diferencial.

A.4 Tracção síncrona

Nesta configuração cada roda do robô possui um conjunto com tracção e direcção (Figura A.4). O robô com tracção síncrona é quase um robô omnidireccional, já que o robô pode se movimentar para qualquer direcção. Entretanto, o robô tem que parar e realinhar as rodas dependendo da sua movimentação. Esta abordagem foi inicialmente desenvolvida para que o robô consiga cobrir completamente o ambiente onde se encontra.

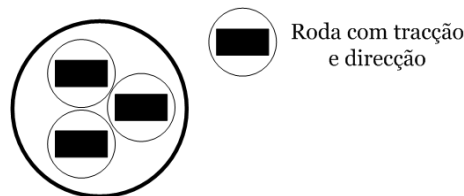


Figura A.4: Tracção síncrona.

A.5 Ackermann

Esta configuração é similar a dos automóveis, possui duas rodas de tracção combinadas, e duas rodas de direcção combinadas(Figura A.5). Esta configuração tem algumas desvantagens comparada com a tracção diferencial: o robô necessita de um raio mínimo para efectuar curvas; as rodas de tracção podem derrapar em curvas.

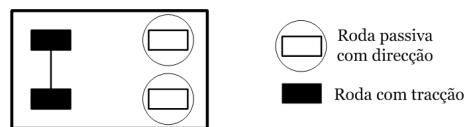


Figura A.5: Ackermann.

Referências

- ALBAGUL, A. & WAHYUDI (2004). Dynamic modeling and adaptive traction control for mobile robots. *International Journal of Advanced Robotic Systems*, **1**, 149–154. [37](#)
- ANTON, H. & RORRES, C. (2000). *Elementary Linear Algebra - Applications version*. John Wiley Sons Inc. [70](#)
- ASTROM, K.J. & HAGGLUND, T. (1995). *PID controllers : Theory, design, and tuning*. International Society for Measurement and Con, 2nd edn. [32](#)
- BALCH, T. & ARKIN, R. (1998). Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, **14**, 926–939. [122](#)
- BEAUDRY, J., MARLEAU, S. & FOURNIER, P. (2005). Robofoot epm team description. *Robocup 2005*. [122](#)
- BIANCO, C.G.L., PIAZZI, A. & ROMANO, M. (2004). Smooth control of a wheeled omnidirectional robot. In *5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles - IAV2004*, Lisbon-Portugal. [64](#)
- BRAUNL, T. (2003). *Embedded Robotics*. Springer. [12](#), [143](#)
- CAMACHO, E.F. & BORDONS, C. (2004). *Model Predictive Control*. Springer-Verlag, 2nd edn. [85](#), [88](#)
- CANUDAS DE WIT, C. & TSOTRAS, P. (1999). Dynamic tire friction models for vehicle traction control. In *38th IEEE Conference on Decision and Control, CDC*, vol. 4, 3746–3751. [37](#)

- CARVALHO, J.M.D. (1993). *Dynamical Systems and Automatic Control*. Prentice Hall. [18](#)
- CLARKE, D.W., MOHTADI, C. & TUFFS, P.S. (1987). Generalized predictive control. *Automatica*, **23**, 137–148. [87](#)
- CONCEIÇÃO, A.S., MOREIRA, A.P. & COSTA, P.J. (2005). Control and model identification of a mobile robot’s motor based in least squares and instrumental variable methods. In *Proceedings of the 11th IEEE International Conference on Methods and Models in Automation and Robotics - MMAR2005*, vol. 1, 479–484, Miedzydroje, Poland. [15](#)
- CONCEIÇÃO, A.S., MOREIRA, A.P. & COSTA, P.J. (2006a). Trajectory tracking for omni-directional mobile robots based on restrictions of the motor’s velocities. In *8th International IFAC Symposium on Robot Control - Syroco 06*, Bologna, Italy. [15](#)
- CONCEIÇÃO, A.S., MOREIRA, A.P., COSTA, P.J. & REIS, L.P. (2006b). Architecture of cooperation for multi-robot systems. In *First IFAC Workshop on Multivehicle Systems (MVS’06)*, Salvador, Brazil. [127](#), [128](#), [129](#)
- CUTLER & RAMAKER (1980). Dynamic matrix control—a computer control algorithm. In *American Control Conference*, San Francisco, CA. [87](#)
- DEKEYSER, R. & CAUWENBERGHE, A.V. (1985). Extended prediction self-adaptive control. In *7th IFAC Symposium on Identification and System Parameter Estimation*, 1255–1260, York, UK. [87](#)
- DUDEK, G. & JENKIN, M. (2000). *Computational Principles of Mobile Robotics*. Cambridge University Press. [1](#), [2](#), [38](#)
- DURFEE, E.H. (1996). Planning in distributed artificial intelligence. In G. O’Hare & N. Jennings, eds., *Foundations of Distributed Artificial Intelligence*, 231–246, John Wiley & Sons, Inc. [121](#), [126](#)
- DURFEE, E.H. (1999). Distributed problem solving and planning. In G. Weiss, ed., *Multiagent Systems A Modern Approach to Distributed Artificial Intelligence*, 121–164, The MIT Press. [127](#), [129](#)

- FLETCHER, R. (1987). *Practical methods of optimization*. John Wiley & Sons. [96](#)
- FLETCHER, R. & REEVES, C.M. (1964). Function minimization by conjugate gradients. *The Computer Journal*, **7**, 149–154. [96](#)
- FRANKLIN, G.F., POWELL, J. & WORKMAN, M. (1997). *Digital control of dynamic systems*. Addison Weley Longman, Inc, Boston. [35](#), [46](#)
- FUJII, H., OTSUKA, F., SEMA, N., KAWASHIMA, M., SUGIYAMA, E., NIIZUMA, S., KAWAKAMI, K., MAKISHIMA, S. & YOSHIDA, K. (2005). Eigen team description. *Robocup 2005*. [122](#)
- GENESERETH, M., GINSBERG, M. & ROSENCEIN, J. (1986). Cooperation without communication. In *Proceedings of the 5th National Conference on Artificial Intelligence, AAAI86*, 51–57, Philadelphia, PA. [126](#)
- GILL, P.E., MURRAY, W. & WRIGHT, M.H. (1981). *Practical Optimization*. Academic Press Inc. [96](#)
- GOLDA, D., IAGNEMMA, K. & DUBOWSKY, S. (2004). Probabilistic modeling and analysis of high-speed rough-terrain mobile robots. In *IEEE International Conference on Robotics and Automation, ICRA*, vol. 1, 914–919 Vol.1. [37](#)
- GOMEZ-ORTEGA, J. & CAMACHO, E.F. (1994). Neural network mbpc for mobile robots path tracking. *Robotics and Computer Integrated Manufacturing Journal*, **11**, 271–278. [88](#)
- GOODWIN, G.C. & PAYNE, R.L. (1997). *Dinamic System Identification: Experiment Design and Data Analysis*, vol. 136. Academic Press. [19](#), [20](#), [28](#)
- GRANA DRUMMOND, L.M. & SVAITER, B.F. (2005). A steepest descent method for vector optimization. *Journal of Computational and Applied Mathematics*, **175**, 395–414. [96](#)
- GU, D. & HU, H. (2002). Neural predictive control for a car-like mobile robot. *Robotics and Autonomous Systems*, **39**, 73–86. [88](#)

- HAMERLAIN, F., ACHOUR, K., FLOQUET, T. & PERRUQUETTI, W. (2005). Higher order sliding mode control of wheeled mobile robots in the presence of sliding effects. In *44th IEEE Conference on Decision and Control and European Control Conference. CDC-ECC*, 1959–1963. [37](#)
- HE, Z., LIN, L. & MA, X. (2004). Design, modeling and trajectory generation of a kind of four wheeled omni-directional vehicle. *IEEE International Conference on Systems, Man and Cybernetics*, 6125–6130. [65](#)
- HUHNS, M.N. & STEPHENS, L.M. (1999). Multiagent systems and societies of agents. In G. Weiss, ed., *Multiagent Systems A Modern Approach to Distributed Artificial Intelligence*, 79–120, The MIT Press. [124](#)
- IAGNEMMA, K., SHINWOO, K., SHIBLY, H. & DUBOWSKY, S. (2004). Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers. *IEEE Transactions on Robotics*, **20**, 921–927, 1552-3098. [37](#)
- INVACARE (2007). *Invacare Corporation*, vol. acessado em julho/2007. <http://www.invacare.com>. [2](#)
- JAE HEON, C., BYUNG-JU, Y., WHEE KUK, K. & HOGIL, L. (2003). The dynamic modeling and analysis for an omnidirectional mobile robot with three caster wheels. In *IEEE International Conference on Robotics and Automation, ICRA*, vol. 1, 521–527. [37](#)
- JENNINGS, N.R. (1996). Coordination techniques for distributed artificial intelligence. In G. O’Hare & N. Jennings, eds., *Foundations of Distributed Artificial Intelligence*, 187–210, John Wiley & Sons, Inc. [124](#), [125](#), [127](#), [129](#)
- JONATHAN, R.S. (1994). An introduction to the conjugate gradient method without the agonizing pain. Tech. rep., Carnegie Mellon University, 865018. [96](#)
- JUN, T., WATANABE, K. & SHIRAISHI, Y. (1996). Design and traveling experiment of an omnidirectional holonomic mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems , IROS*, vol. 1, 66–73. [37](#)

- KALMAR-NAGY, T., D'ANDREA, R. & GANGULY, P. (2004). Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. *Robotics and Autonomous Systems*, **46**, 47–64. [64](#)
- KEIGO, W., YAMATO, S., SPYROS, G.T., JUN, T. & TOSHIO, F. (1998). Feedback control of an omnidirectional autonomous platform for mobile service robots. *Journal of Intelligent and Robotic Systems*, **V22**, 315–330, 10.1023/A:1008048307352. [37](#), [64](#)
- KITANO, H., ASADA, M., KUNIYOSHI, Y., NODA, I. & OSAWA, E. (1997a). Robocup: The robot world cup initiative. *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, 340–347. [122](#)
- KITANO, H., KUNIYOSHI, Y., NODA, I., ASADA, M., MATSUBARA, H. & OSAWA, E. (1997b). Robocup: A challenge problem for AI. *AI Magazine*, **18**, 73–85. [10](#)
- KUO, B.C. (1995). *Automatic Control Systems*. John Wiley & Sons, Inc. [39](#), [40](#), [41](#)
- LESSER, V. (1995). Multiagent systems: An emerging subdiscipline of ai. *ACM Computing Surveys*, **27**, 340–342. [124](#)
- LI-CHUN, L., CHIA-NAN, K., TSONG-LI, L. & CHIA-JU, W. (2006). Time-optimal control of an omni-directional mobile robot. In *1ST IEEE Conference on Industrial Electronics and Applications*, 1–6, Singapore. [64](#)
- LIU, Y., WU, X., ZHU, J.J. & LEW, J. (2003). Omni-directional mobile robot controller design by trajectory linearization. *Proceedings of the American Control Conference*, **4**, 3423 – 3428. [64](#)
- MOBILEROBOTS (2007). *MobileRobots Inc*, vol. acessado em julho/2007. <http://www.mobilerobots.com/>. [2](#)
- MOORE, K.L. & FLANN, N.S. (2000). A six-wheeled omnidirectional autonomous mobile robot. *IEEE Control Systems Magazine*, **20**, 53–66, 0272-1708. [64](#)

- MOREIRA, A.P., COSTA, P., CONCEIÇÃO, A.S., SOUSA, A. & MARQUES, P. (2006). 5dpo-2000 team description for robocup 2006. *Team Description Papers, RoboCup 2006*. [11](#)
- MSL (2007). *Middle Size Robot League Rules and Regulations for 2007*, vol. Final Version - 10.5 20070116. RoboCup - Technical Committee. [10](#)
- MUIR, P. & NEUMAN, C. (1987). Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. In *IEEE International Conference on Robotics and Automation*, vol. 4, 1772–1778. [64](#)
- MUNOZ, V., OLLERO, A., PRADO, M. & SIMON, A. (1994). Mobile robot trajectory planning with dynamic and kinematic constraints. In *IEEE International Conference on Robotics and Automation*, vol. 4, 2802–2807. [64](#)
- MURPHY, R.R. (2000). *Introduction to AI robotics*. The MIT Press. [2](#)
- PICHE, S., SAYYAR-RODSARI, B., JOHNSON, D. & GERULES, M. (2000). Non-linear model predictive control using neural networks. *Control Systems Magazine, IEEE*, **20**, 53–62, 0272-1708. [88](#)
- PIGNON, P., HASEGAWA, T. & LAUMOND, J.P. (1992). Basic algorithms for space structuring in path planning for mobile robots. In *IEEE International Conference on Robotics and Automation*, 2495–2500 vol.3. [64](#)
- PIN, F.G. & KILLOUGH, S.M. (1994). A new family of omnidirectional and holonomic wheeled platforms for mobile robots. *Robotics and Automation, IEEE Transactions on*, **10**, 480–489, 1042-296X. [12](#), [37](#)
- PURWIN, O. & D’ANDREA, R. (2005). Trajectory generation for four wheeled omnidirectional vehicles. In *American Control Conference*, vol. 7, 4979–4984. [65](#)
- RAMIREZ, D.R., LIMON, D., GOMEZ-ORTEGA, J. & CAMACHO, E.F. (1999). Nonlinear mbpc for mobile robot navigation using genetic algorithms. In *International Conference on Robotics & Automation*, vol. 3, 2452–2457 vol.3, Detroit, Michigan. [88](#)

- REIS, L.P. (2003). Coordination in multi-agent systems: Applications in university management and robotic soccer. [124](#), [127](#), [128](#), [129](#)
- RICHALET, J. (1987). Predictive functional control: Application to fast and accurate robots. In *IFAC 10th World Congress*, Munich. [87](#)
- RIEDMILLER, M. & MERKE, A. (2002). Using machine learning techniques in complex multi-agent domains. In I. Stamatescu, W. Menzel & U. Ratsch, eds., *Perspectives on Adaptivity and Learning.*, Springer. [122](#)
- ROBOCUP (2007). *Robocup Official Site*, vol. acessado em junho/2007. <http://www.robocup.org/>. [9](#)
- RUSSEL, S. & NORVIG, P. (1995). *Artificial Intelligence a Modern Approach*. Prentice-Hall, Inc. [123](#)
- SAR (2007). *SAR - Soluções de Automação e Robótica*, vol. acessado em julho/2007. <http://www.industry.sarobotica.pt/>. [3](#)
- SILVA, A.P.G.S.E., OLIVEIRA, D.F.C. & CARVALHO, P.M.F. (2007). *Veículo Auto-Guiado para Transporte e Armazenamento de Pequenos Volumes*. Trabalho de Conclusão de Curso - FEUP. [3](#)
- SINGH, M.P. (1990). Group intentions. In *In Proceedings of the 10th International Workshop on Distributed Artificial Intelligence (DAI).*, MCC Tech. Rep. ACT-AI-355-90, Bandera, Texas. [127](#), [129](#)
- SOETERBOEK, A.R.M., VERBRUGGEN, H.B., VAN DEN BOSCH, P.P.J. & BUTLER, H. (1990). On the unification of predictive control algorithms. In *29th IEEE Conference on Decision and Control*, 1451–1456 vol.3. [87](#)
- SOETERBOEK, R. (1992). *Predictive control- A unified approach*. New York. [87](#)
- SOUERES, P. & LAUMOND, J.P. (1996). Shortest paths synthesis for a car-like robot. *IEEE Transactions on Automatic Control*, **41**, 672–688, 0018-9286. [64](#)
- STONE, P. & VELOSO, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, **8**, 929–5593 (Print) 1573–7527 (Online). [124](#)

REFERÊNCIAS

- SUNGYONG, P. & MINOR, M.A. (2004). Modeling and dynamic control of compliant framed wheeled modular mobile robots. In *IEEE International Conference on Robotics and Automation, ICRA 04*, vol. 4, 3937–3943. 37
- TAI-YU, W. & CHING-CHIH, T. (2004). Adaptive trajectory tracking control of a wheeled mobile robot via lyapunov techniques. In *30th Annual Conference of IEEE Industrial Electronics Society, IECON*, vol. 1, 389–394. 37
- TANAKA, Y., TSUJI, T., KANEKO, M. & MORASSO, P.G. (1998). Trajectory generation using time scaled artificial potential field. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 223–228. 64
- TECNOGIAL (2007). *Tecnogial Lda - Construções Mecânicas*, vol. acessado em junho/2007. <http://www.hotfrog.pt/Empresas/Tecnogial-Lda>. 12
- THE MATHWORKS, I. (1994-2006a). *Communications Toolbox - Documentation*, vol. acessado em junho/2007. <http://www.mathworks.com/access/helpdesk/help/helpdesk.html>. 20
- THE MATHWORKS, I. (1994-2006b). *Neural Network Toolbox - Documentation*, vol. acessado em junho/2007. <http://www.mathworks.com/access/helpdesk/help/helpdesk.html>. 25
- VACCARO, R.J. (1995). *Digital Control - A State-Space Approach*. McGraw-Hill. 18, 32, 34
- VENTURE, G., KHALIL, W., GAUTIER, M. & BODSON, P. (2004). Identification of the dynamic parameters of a car: simulation and experimental results. In *IFAC symposium AAC*, Salerno, Italia. 37
- WERNER, E. (1989). Cooperating agents: A unified theory of communication and social structure. In L. Gasser & M.N. Huhs, eds., *Distributed Artificial Intelligence*, vol. 2, 3–36, CA/Pilman, London. 128
- WILLIAM, W.H. & HONGCHAO, Z. (2005). A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. on Optimization*, 16, 170–192, 1085581. 96

- WILLIAM, W.H. & HONGCHAO, Z. (2006). Algorithm 851: Cg_descent, a conjugate gradient method with guaranteed descent. *ACM Transactions on Mathematical Software*, **32**, 113–137, 1132979. [96](#)
- WILLIAMS, I., R. L., CARTER, B.E., GALLINA, P. & ROSATI, G. (2002). Dynamic model with slip for wheeled omnidirectional robots. *IEEE Transactions on Robotics and Automation*, **18**, 285–293, 1042-296X. [37](#)
- WOOLDRIDGE, M. (2002). *An introduction to multiagent systems*. John Wiley & Sons. [123](#), [124](#), [126](#)
- WOOLDRIDGE, M. & JENNINGS, N.R. (1995). Intelligent agent: theory and practice. *Knowledge Engineering Review*, **10**, 115–152. [123](#)
- XIANHUA, J., MOTAI, Y. & XINGQUAN, Z. (2005). Predictive fuzzy control for a mobile robot with nonholonomic constraints. In *12th International Conference on Advanced Robotics, ICAR '05*, 58–63. [88](#)
- YDSTIE, B. (1984). Extended horizon adaptive control. In *9th IFAC World. Congress*, 133–137, Budapest. [87](#)