

UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA



DISSERTAÇÃO DE MESTRADO

**ESTUDO COMPARATIVO APLICADO AO PLANEJAMENTO
E CONTROLE DE TRAJETÓRIAS DE ROBÔS MÓVEIS**

FRANCISCO VANIER DE ANDRADE
ORIENTADOR: Dr. BISMARCK CLAURE TORRICO
CO-ORIENTADOR: Dr. OTACÍLIO DA MOTA ALMEIDA

Fortaleza - Ce
Outubro - 2011

Francisco Vanier de Andrade

ESTUDO COMPARATIVO APLICADO AO PLANEJAMENTO E CONTROLE DE
TRAJETÓRIAS DE ROBÔS MÓVEIS

Dissertação apresentada ao Departamento
de Engenharia Elétrica sob orientação do
Professor Dr. Bismark Claure Torrico,
como requisito para a obtenção do título de
Mestre em Engenharia Elétrica.

Fortaleza - Ce
Outubro - 2011

A553e

Andrade, Francisco Vanier de.

Estudo comparativo aplicado ao planejamento e controle de trajetórias de robôs móveis. / Francisco Vanier de Andrade [orientada por] Bismark Claude Torrico. – Fortaleza: 2011.

102 p.

Dissertação (Mestrado) - Universidade Federal do Ceará, 2011.

1. Engenharia Elétrica. 2. Robôs móveis.
I. Título II. Andrade, Francisco Vanier de.

CDD 621.3

ESTUDO COMPARATIVO APLICADO AO PLANEJAMENTO E CONTROLE DE
TRAJETÓRIAS DE ROBÔS MÓVEIS

Francisco Vanier de Andrade

Esta dissertação foi julgada adequada para obtenção do título de Mestre em Engenharia Elétrica, área de Eletrônica de Potência e Acionamento de Máquinas e aprovado em sua forma final pelo programa de Pós-Graduação em Engenharia Elétrica na Universidade Federal do Ceará.

Francisco Vanier de Andrade

Banca Examinadora:

Prof. Dr. Bismark Claire Torrico

Orientador – DEE UFC

Prof. Dr. Otacílio da Mota Almeida

Co-Orientador – DEE UFC

Prof. Dr. José Carlos Teles Campos

DEE UFC

Prof. Dr. Eloy de Macedo Silva

Examinador Externo - IFCE

Fortaleza, Outubro de 2011

DEDICATÓRIA

Dedico este trabalho à memória de meu pai.
À minha mãe e irmãos, e para Marcus e Miguel.

AGRADECIMENTOS

Ao bondoso Deus, fonte de toda inspiração e sabedoria.

Aos meus orientadores, Prof. Dr. Otacílio da Mota Almeida e Prof. Dr. Bismark Claire Torrico, pela compreensão e apoio dado à conclusão deste trabalho.

Ao meu irmão Vêber, por sua ajuda na finalização do trabalho e construção do robô.

À minha irmã Valquíria e meu cunhado Carlos pelo apoio dado.

Aos colegas do mestrado, ao Rafael Gomes e demais funcionários do Departamento de Engenharia Elétrica pela amizade.

“O princípio da sabedoria é o desejo autêntico de instrução, e a preocupação pela instrução é o amor...”

“...Por isso, eu supliquei e a inteligência me foi dada. Invoquei, e o espírito da sabedoria veio até mim. Eu a preferi aos cetros e tronos e, em comparação com ela considerei a riqueza como um nada. Não a comprei com a pedra mais preciosa, porque todo o ouro, ao lado dela, é como um punhado de areia. E junto dela, a prata vale o mesmo que um punhado de barro. Amei a sabedoria mais do que a saúde e a beleza, e resolvi tê-la como luz, porque o brilho dela nunca se apaga. Com ela vieram todos os bens, e em suas mãos existe riqueza incalculável.”

“Sem malícia, aprendi a sabedoria, e agora a distribuo sem inveja nenhuma”.

(Livro da Sabedoria 6:17; 7:7-11; 6:10)

Andrade, F. V. de, “Estudo comparativo aplicado ao planejamento e controle de trajetórias de robôs móveis”, Universidade Federal do Ceará – UFC, 2011.

O presente trabalho propõe o estudo comparativo de duas técnicas de controle aplicadas às trajetórias de um robô móvel: GPC (*Generalized Predictive Control*) e PID (Proporcional – Integral - Derivativo). O sistema de controle do robô utiliza a configuração em cascata, sendo constituído de uma malha interna para o controle de velocidade dos motores e uma malha externa para o controle de posição e orientação do robô que fornecerá a referência de velocidade dos motores. O sistema de controle em cascata permite utilizar-se de controladores cinemáticos mono-variável. Foi utilizado ainda um controlador dinâmico preditivo multi-variável a fim de se comparar o desempenho com os outros controladores mono-variável. Foram desenvolvidos simuladores para o robô móvel através de uso dos controladores GPC e PID. A teoria foi implementada em um pequeno robô móvel com tração diferencial construído para este propósito. Foram utilizados como atuadores motores CC e como sensores de velocidade e posição, foram utilizados *encoders* e sonares, respectivamente. Uma vez que o ajuste dos controladores usados é baseado em modelo, foram utilizados dois métodos de identificação de parâmetros: o Método dos Mínimos Quadrados Recursivos (MQR) e por Redes Neurais. Os métodos de controle empregados permitiram o seguimento de trajetórias pelo robô móvel.

Palavras-chave: Controle Preditivo, GPC, Robôs Móveis, PID, Rede Neural.

Andrade, F. V. de, "Comparative study applied to planning and controlling of mobile robots trajectories", Universidade Federal do Ceará – UFC, 2011.

The present work proposes a comparative study of two control techniques applied for mobile robots trajectories: GPC (Generalized Predictive Control) and PID (Proportional – Integral - Derivative). The control system uses a cascade configuration composed by an internal loop for controlling the speed of motors, and an external loop for position and orientation control, being used to get the reference speed motors. This cascade control system can run a mono-variable cinematic controller. It was used a multi-variable predictive controller to compare the behavior with the single-variable controllers. It was developed simulators for mobile robot using GPC and PID controllers. The theory was applied to a little differential mobile robot, designed to this proposes. It was used as actuators DC motors, and speed e positional sensors, encoders and sonar, respectively. Once the controllers adjust are referenced by model, was used two identification methods: Recursive Minimum Square and Neural Networks. The control methods employed had permitted the trajectories following by the mobile robot.

Keywords: Predictive Control, GPC, Mobile robots, PID, Neural Network.

SUMÁRIO

CAPÍTULO 1 - Introdução.....	1
1.1 Objetivos do Trabalho.....	6
1.2 Organização do Texto.....	6
CAPÍTULO 2 - Noções de Robótica Móvel.....	8
2.1 Robôs Exploradores.....	8
2.2 Arquiteturas.....	10
2.3 O problema da trajetória	13
CAPÍTULO 3 - Modelagem e Identificação	16
3.1 Modelagem matemática.....	16
3.2 Restrições Cinemáticas.....	18
3.2.1 Roda Fixa	19
3.2.2 Roda Castor	20
3.3 Grau de Mobilidade	21
3.4 Grau de Direcionabilidade.....	22
3.5 Cinemática	23
3.6 Modelagem dos Atuadores	26
3.7 Modelagem Cinemática.....	27
3.8 Identificação de Sistemas	31
3.9 Considerações sobre sistemas de controle.....	33
3.10 Modelo Discreto	35
3.11 Mínimos Quadrados Recursivo	37
3.12 Redes Neurais	38
CAPÍTULO 4 - Controladores.....	44
4.1 Controle Preditivo.....	44
4.2 Controle GPC.....	47
4.3 Controle PID.....	50
4.4 Controle em cascata GPC-PID.....	54
CAPÍTULO 5 – O Robô Móvel.....	56
5.1 O Robô Móvel.....	56
5.2 Mensagens de comunicação.....	60
5.2.1 Mensagens originadas pelo controlador	61
5.2.2 Mensagens originadas pelo robô	62
5.3 Algoritmo de execução da trajetória.....	63

CAPÍTULO 6 – Resultados e Discussões	65
6.1 Dados coletados de velocidade	65
6.2 Resultados de identificação dos motores	66
6.2.1 Mínimos Quadrados Recursivo	66
6.2.2 Redes Neurais	67
6.3 Controle dos motores com o PID	68
6.3.1 Identificação por RNA	68
6.3.2 Identificação MQR	70
6.4 Controle dos motores com o GPC	72
6.4.1 Identificação MQR	73
6.5 O simulador MATLAB do robô móvel	74
6.6 O simulador Simulink	78
6.7 O simulador dos Controladores – SIMULINK	82
6.8 Resultados experimentais de seguimento de trajetória	98
Conclusão e Sugestões de Trabalhos Futuros.....	101

LISTA DE ILUSTRAÇÕES

Figura 1.1 iRobot 510 PackBot

Figura 1.2 Robô Remora usado nas buscas de destroços de um acidente aéreo

Figura 1.3 Robô ASIMO

Figura 2.1 Sondas Sojourner (1997) e Spirit (2004)

Figura 2.2 Comparativo do tamanho dos robôs Sojourner e Spirit

Figura 2.3 Robô Curiosity no laboratório de Jato Propulsão

Figura 2.4 Arquitetura Geral de Robôs Móveis

Figura 3.1 Esboço do robô

Figura 3.2 Referência local e global

Figura 3.3 Roda Fixa

Figura 3.4 Roda Castor

Figura 3.5 Deslocamento infinitesimal

Figura 3.6 Diagrama do Motor CC

Figura 3.7 Velocidade em função do tempo

Figura 3.8 Aproximação linear da velocidade

Figura 3.9 Verificação do modelo

Figura 3.10 Soma quadrática dos erros

Figura 3.11 Correlação múltipla

Figura 3.12 Estimação recursiva

Figura 3.13 Neurônio humano

Figura 3.14 Modelo de um neurônio

Figura 3.15 Modelo de entrada /saída

Figura 3.16 Modelo de Espaço de Estados

Figura 4.1 Estrutura do MPC

Figura 4.2 Controle geral MPC

Figura 4.3 Lei de controle GPC

Figura 4.4 Controle em cascata GPC-PID

Figura 5.1 Robô Móvel GPAR

Figura 5.2 Sistema de Redução

Figura 5.3 Sub-sistemas do robô móvel

Figura 5.4 Fluxograma do Controle de Trajetória

Figura 6.1 Coleta de dados de velocidade

Figura 6.2 Ruído presente nas medidas

Figura 6.3 Identificação de parâmetros do motor esquerdo (MQR)

Figura 6.4 Identificação de parâmetros do motor esquerdo (RNA)

Figura 6.5 Desempenho do PID (Identificação RNA – 80 neurônios ocultos)

Figura 6.6 Desempenho do PID (Identificação RNA – 120 neurônios ocultos)

Figura 6.7 Desempenho do PID (Identificação MQR)

Figura 6.8 Desempenho do controlador PID (MQR – média de 3 amostras)

Figura 6.9 Desempenho do controlador PID (MQR média) referência diferente

Figura 6.10 Resultados com o controlador GPC (identificação MQR)

Figura 6.11 Interface do simulador do robô móvel

Figura 6.12 Parâmetros dos motores (simulação)

Figura 6.13 Referências geradas pelo simulador

Figura 6.14 Trajetória do simulador do robô móvel

Figura 6.15 Trajetória calculada pelo simulador

Figura 6.16 Erros de Posição

Figura 6.17 Diagrama de blocos do simulador

Figura 6.18 Trajetória obtida com a atuação de perturbações de amplitudes diferentes

Figura 6.19 Velocidades dos motores com os controladores PI

Figura 6.20 Velocidades dos motores com os controladores PI (sem sobressinal)

Figura 6.21 Identificação pelo método Ziegler-Nichols (motor direito)

Figura 6.22 Identificação pelo método Ziegler-Nichols (motor esquerdo)

Figura 6.23 Velocidades controladas pelo PI com a inclusão dos ruídos

Figura 6.24 Controladores GPC de posição e orientação

Figura 6.25 Controladores PID de posição e orientação

Figura 6.25a Transformação em ω_E e ω_D

Figura 6.25b Malha Interna

Figura 6.25c Transformação em v e ω

Figura 6.25d Cinemática do robô

Figura 6.26 Controlador PID da malha interna (motor esquerdo)

Figura 6.27 Controlador PID da malha interna (motor direito)

Figura 6.28 Trajetória com o controlador PID da malha externa

Figura 6.29 Trajetória com o controlador GPC da malha externa
Figura 6.30 Trajetória com o controlador PID da malha externa
Figura 6.31 Trajetória com o controlador GPC da malha externa
Figura 6.32 Trajetória com o controlador GPC MIMO
Figura 6.33 Trajetória com o controlador PID da malha externa
Figura 6.34 Trajetória com o controlador GPC da malha externa
Figura 6.35 Trajetória com o controlador GPC MIMO
Figura 6.36 Trajetória com o controlador PID da malha externa
Figura 6.37 Trajetória com o controlador GPC da malha externa
Figura 6.38 Trajetória com o controlador GPC MIMO
Figura 6.39 Trajetória com o controlador PID da malha externa
Figura 6.40 Trajetória com o controlador GPC da malha externa
Figura 6.41 Trajetória com o controlador GPC MIMO
Figura 6.42 Trajetória com o controlador GPC MIMO
Figura 6.43 Trajetória com o controlador GPC MIMO
Figura 6.44 Trajetória com o controlador GPC MIMO
Figura 6.45 Trajetória com o controlador GPC MIMO
Figura 6.46 Trajetória real executada com o PID
Figura 6.47 Trajetória real executada com o GPC
Figura 6.48 Trajetória real com o PID
Figura 6.49 Trajetória real com o GPC

LISTA DE TABELAS

Tabela 3.1 Simbologia utilizada

Tabela 5.1 Características do robô móvel

Tabela 5.2 Parâmetros de comunicação

Tabela 5.3 Mensagens originadas pelo controlador

LISTA DE ABREVIATURAS E SIGLAS

AGV – *Auto Guide Vehicle*
BIBO – *Bounded Input-Bounded Output*
CARIMA – *Controller Auto-Regressive Integrated Moving Average*
CARMA – *Controller Auto-Regressive Moving Average*
CPBM – *Controle Preditivo Baseado em Modelo*
CC – *Corrente Contínua*
DIF – *Detecção e Isolação de Falhas*
DMC – *Dynamic Matrix Control*
EPSAC – *Extended Self Adaptive Control*
FPGA – *Field Programmable Gate Array*
IA – *Inteligência Artificial*
GPAR – *Grupo de Pesquisas em Automação e Robótica*
GPC – *Generalized Predictive Control*
JPL – *Jet Propulsion Laboratory*
MIMO – *Multiple Input Multiple Output*
MLP – *Multi Layer Perceptron*
MPC – *Model Predictive Control*
MQR – *Minimos Quadrados Recursivo*
NARX – *Nonlinear Auto-Regressive with Exogenous Input*
NASA – *National Aeronautics and Space Administration*
NEPSAC – *Non-linear Extended Self Adaptive Control*
PID – *Proporcional- Integral – Derivativo*
PRBS – *Pseudo Random Binary Signal*
PWM – *Pulse Width Modulation*
RNA – *Redes Neurais Artificiais*
ROV – *Remote Operated Vehicle*
SISO – *Single Input Single Output*

LISTA DE SÍMBOLOS

LATINOS

a	Aceleração angular
e	Erro de modelagem
l	Largura do robô
m	Massa do robô
r	Raio da roda
s	Espaço percorrido
T_a	Período de amostragem
v	Velocidade linear
vs	Velocidade do som (341 m/s)

GREGOS

ω	Velocidade angular
θ	Ângulo de orientação
λ	Comprimento de onda
τ	Constante de tempo

CAPÍTULO 1 - Introdução

A robótica móvel está sendo utilizada nos diversos ambientes, sejam eles industrial ou doméstico. Podem-se visualizar aplicações em muitos trabalhos insalubres ou perigosos ao ser humano, como na exploração de ambientes hostis, auxílio no desarmamento de bombas, entre outros.

Desastres nucleares, como os ocorridos nas usinas de Chernobil (1986) e Fukushima (2011), necessitaram de robôs móveis para auxiliar os trabalhos, onde o nível de radioatividade era alto demais para se utilizar pessoas.

O robô iRobot 510 PackBot está sendo usado para avaliar danos e medir a radiação em Fukushima.



Figura 1.1 iRobot 510 PackBot

Fonte: <http://info.abril.com.br>

“A Nasa criou os controles de torque e a estrutura de liga leve do chassi que, hoje, são usadas para examinar o interior dos reatores nucleares japoneses em áreas ainda não seguras para humanos. Outro ancestral do PackBot é o robô Urbie, projetado para aplicações militares ou de emergência – como contaminações químicas e terremotos. A estrutura física do robô foi desenvolvida pela iRobot, empresa de Massachusetts, enquanto os sensores inteligentes e algoritmos de visão ficaram por conta do JPL. É

graças a esses mecanismos que o robô consegue passar por obstáculos e determinar rotas. Após o sucesso do Urbie, a iRobot usou sua tecnologia para criar seu sucessor, o PackBot.

Variações do robô já foram usadas pelo exército, marinha e força aérea americanas. Em julho de 2002, por exemplo, o veículo serviu como olhos e ouvidos dos soldados em áreas de risco no Afeganistão. O PackBot também ajudou nas buscas dos destroços do ataque do 11 de setembro, em 2001, em Nova York.

Agora, a iRobot forneceu dois veículos para ajudar o Japão após o terremoto e o tsunami de 11 de março. No momento, eles fazem leituras dos níveis de radioatividade na usina nuclear de Fukushima. Seus sensores indicam que a radiação ainda é muito alta para a presença de equipes de reparo.”

Fonte: <http://info.abril.com.br>

Robôs submarinos podem localizar e recolher destroços de acidentes ocorridos com navios e aviões.

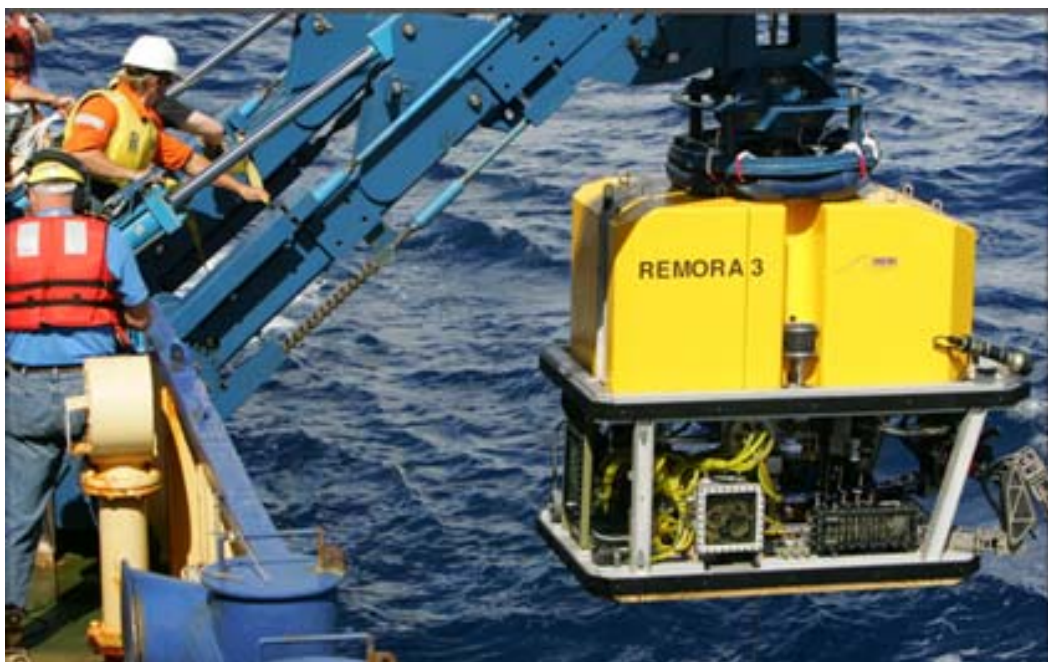


Figura 1.2 Robô Remora usado nas buscas de destroços de um acidente aéreo

Fonte: <http://terra.com.br/noticias>

“ O pequeno tamanho do robô permite que o veículo se desloque com precisão em espaços apertados. O Remora 6000 é controlado remotamente por operadores. A máquina é composta por duas câmeras: uma auxilia a navegação e a outra, a busca em meio aos destroços.

As imagens coletadas pelo robô e o fornecimento de energia são transmitidos por meio de cabos de fibra ótica que ligam o equipamento ao navio. As caixas-pretas foram recolhidas pelo Remora com auxílio dos braços mecânicos. Os equipamentos do Airbus foram colocados em uma espécie de gaiola, afixada com um cabo na parte de baixo do robô e transportada por cerca de 4 km. O Remora 6000 conta com 25 cavalos de potência, quatro propulsores laterais axiais e dois verticais, altímetro, sensores de pressão, scanner sonar e lâmpadas. Com todos os periféricos, chega a pesar 900 kg e atinge 1,7 m de comprimento, 1 m de largura e 1,2 m de altura. Tem capacidade de operar em até 6 km de profundidade.”

Fonte: <http://terra.com.br/noticias>

O sistema do Remora 6000 é complexo e inclui um veículo, cabo de fibra ótica, guincho, sistema para lançamento e recuperação, além de dispositivos de manutenção. Além da coleta debaixo da água, o Remora pode ser utilizado para pesquisa, documentação visual, instalação de equipamentos e suporte a cabos submarinos.

O Remora 6000 integra a família dos Veículos Operados Remotamente (ROV). Normalmente usado em atividades de exploração de petróleo submarinas, o robô é equipado com câmeras, "braços" articulados e é capaz de manipular equipamentos para cortar fuselagens.

Atualmente já se pode também visualizar o uso de robôs móveis no ambiente doméstico e entretenimento, como o futebol de robôs. A figura 1.3 mostra o robô ASIMO produzido pela Honda para fins domésticos.



Figura 1.3 Robô ASIMO

Fonte: <http://howstuffworks.com>

O nível de inteligência dos robôs depende da atividade a ser desenvolvida por ele. Um dos grandes problemas enfrentados pela robótica móvel está relacionado à localização, sendo esta essencial para se ter um controle da trajetória a ser efetuada pelo robô.

O controle de trajetória exige o uso de controladores bem sintonizados. Entre os mais utilizados atualmente no meio industrial estão o PID e o preditivo.

O uso de controladores com ação proporcional-integral-derivativa (PID) foi incorporado à tradição de operação de processos industriais devido à sua simplicidade e ao fato de que quando sintonizados adequadamente proporcionam, na maioria dos casos, um desempenho satisfatório do processo controlado (Almeida, 1990). Esses controladores têm limitação quando a planta possui atrasos ou dinâmica mais complexa. Aplicações onde os controladores PID podem ser utilizados com melhor desempenho podem ser encontradas em (Astrom & Witternmark, 1984).

Diversos métodos de sintonia podem ser encontrados na literatura, dentre estes o método de alocação de pólos para um controlador PID discreto, pode ser encontrado em (Batista, 1985). Outros métodos frequenciais para ajuste de controladores PID podem ser encontrados em (Almeida, 1990).

Em processos mais complexos entretanto, outras técnicas de controle, como o Controle Preditivo Baseado em Modelo (CPBM), podem apresentar melhor resposta.

Os controladores preditivos foram aplicados inicialmente em processos químicos, entretanto sua aplicação foi estendida a diversas áreas (Camacho & Bordons, 1999). Esse tipo de controlador é baseado em um modelo, sendo que este desempenha papel fundamental no sistema de controle. Se os parâmetros da planta são fixos o desempenho é satisfatório, entretanto este nem sempre é o caso.

Outro problema enfrentado pelos sistemas de controle diz respeito aos atrasos, bem como as não-linearidades presentes. Em (Torrico, 2007), estuda-se o controle preditivo robusto de sistemas com atraso e se faz um estudo de aplicação em dosagem de droga para pacientes durante cirurgias. O trabalho aborda tópicos relacionados aos erros de modelagem, além de fazer um estudo detalhado da robustez e estabilidade.

O atraso diminui as margens de ganho e de fase do sistema, dificultando o projeto do sistema de controle. Se este é pequeno (menor que duas vezes a constante de tempo dominante do sistema), e o modelo de baixa ordem, o ajuste do PID permite a obtenção de uma solução aceitável. Mas se o atraso é grande e se deseja obter respostas em malha fechada mais rápidas, então é conveniente utilizar sistemas de compensação de atraso (Camacho & Normey-Rico, 2006). A ação derivativa do controlador PID pode ser interpretada como uma

predição linear do erro, sendo o ajuste desse parâmetro importante para proporcionar um melhor desempenho desse controlador na presença do atraso.

No caso da variação de parâmetros, faz-se necessário o uso de estimadores *on-line*, que permitem a atualização paramétrica constante. Diversas técnicas de identificação de parâmetros utilizadas em sistemas lineares podem ser encontradas em (Coelho & Coelho, 2004).

O interesse por controladores que se adaptem às mudanças ocorridas na planta tem crescido nos últimos anos. Estes controladores são denominados adaptativos, podendo ser implementados através de técnicas de Controle Preditivo, Redes Neurais e Lógica Difusa, dentre outras, sendo que estas têm a vantagem de se adaptar melhor às não-linearidades presentes nos sistemas de controle. Em (Haykin, 2004) é ilustrado o uso de identificação e controle por redes neurais.

Uma das principais áreas onde os controladores adaptativos são utilizados é na robótica móvel, tendo como um dos principais desafios o controle de trajetória caracterizado pela não-holonomia de alguns tipos de robôs, como aqueles que usam rodas e tração diferencial.

O controle de trajetória de robôs móveis pode ser dividido em dois tipos: seguimento de caminho e estabilização (Vieira, 2005). Técnicas de controle preditivo para robôs móveis não-holonômicos podem ser encontradas em (Kuhne, 2005). Em (Lages, 1998) é tratado o controle de posição e orientação de robôs móveis.

Em (Siegwart & Nourbakhsh, 2004) são tratados os robôs móveis autônomos, descrevendo problemas relacionados desde à localização e percepção até o planejamento de rotas, enquanto (Hoffmann, 2004), trata o problema de controle neural de robôs móveis em ambientes não estruturados.

Em (Langer, 2007) é desenvolvido um estudo comparativo entre diversos métodos de planejamento de trajetórias de robôs móveis, enquanto (Dias, 2010) descreve um controlador adaptativo robusto desacoplado para robôs móveis.

A utilização de mapas para navegação de robôs através de um sistema de visão pode ser encontrada em (Deccó, 2004).

Outros autores fazem estudos através de simulação, podendo-se encontrar diversos simuladores em duas e três dimensões em trabalhos relacionados à área. Em (Melo, 2007) foi desenvolvido um simulador para navegação de robôs móveis.

1.1 Objetivos do trabalho

Dentre os principais objetivos deste trabalho destacam-se:

1. A aplicação de dois diferentes métodos de controle, sendo estes: o controle Proporcional-Integral-Derivativo (Astrom & Wittenmark, 1984) e o Controle Preditivo GPC (*Generalized Predictive Control*) encontrado em (Camacho & Bordons, 1999), neste último foram utilizados o controle monovariável e multivariável.
2. Aplicação à robótica móvel de dois métodos de identificação de parâmetros, sendo estes: o método dos mínimos quadrados recursivos (MQR) com busca aleatória (*random walk*), encontrada em (Coelho & Coelho, 2004) e por redes neurais (Haykin, 2004);
3. O estudo comparativo do desempenho das técnicas de controle e identificação empregados na trajetória de robôs móveis;
4. O desenvolvimento de um simulador para um robô móvel;
5. A construção de um pequeno robô móvel para implementação prática dos métodos.

1.2 Organização do texto

O texto foi organizado da seguinte forma:

Capítulo 2: Noções de Robótica Móvel

São apresentados conceitos de robótica móvel, arquiteturas de robôs móveis e problemas de trajetórias.

Capítulo 3: Modelagem e Identificação

São apresentados a modelagem matemática do robô móvel, as restrições cinemáticas dos robôs com ação diferencial, e tópicos de identificação de sistemas, incluindo o método dos mínimos quadrados recursivos e por redes neurais.

Capítulo 4: Controladores

São apresentados tópicos referentes aos controladores preditivo, incluindo o *Generalized Predictive Control* - GPC, e o controlador Proporcional-Integral-Derivativo.

Capítulo 5: O Robô Móvel

É feita uma descrição do robô móvel desenvolvido, além das mensagens de comunicação entre o robô e o controlador.

Capítulo 6: Resultados e Discussões

São mostrados os resultados obtidos e feitas algumas considerações sobre estes.

Conclusão

É apresentada uma conclusão sobre o trabalho e sugestões para trabalhos futuros.

CAPÍTULO 2 – Noções de Robótica Móvel

Este capítulo apresenta conceitos de robótica móvel, e de arquiteturas de robôs móveis, definindo os problemas encontrados no controle de trajetória de robôs móveis.

2.1 Robôs Exploradores

A aplicação para os robôs, que antes se restringia à indústria, pode ser visualizada também no ambiente doméstico. Robôs podem trabalhar em ambientes dinâmicos e executar tarefas complexas.

Pesquisas recentes têm gerado muito desenvolvimento na área da robótica móvel, sendo estes robôs responsáveis por grande parte do trabalho perigoso ou insalubre executado por seres humanos, como: exploração em águas profundas, ambientes radioativos, crateras de vulcões, detecção de minas terrestres, desativação de bombas, auxílio no combate a incêndios e até mesmo exploração de outros planetas.

Os mais recentes exploradores espaciais desenvolvidos pela *National Aeronautics and Space Administration* (NASA) foram: o *SPIRIT*, que aterrisou em Marte em 3 de janeiro de 2004, e *OPPORTUNITY*, que aterrisou em Marte em 24 de janeiro de 2004.

A figura 2.1 mostra as sondas Sojourner (esquerda) e Spirit (direita).



Figura 2.1 Sondas Sojourner (1997) e Spirit (2004)

Fonte: <http://www.nasa.gov>



Figura 2.2 Comparativo do tamanho dos robôs Sojourner e Spirit

Fonte: <http://www.nasa.gov>

Segundo o site <http://terra.com.br>, a NASA pretende lançar em novembro de 2011 um novo robô móvel para exploração do planeta Marte. Trata-se do robô *Curiosity*. O custo da missão deverá ser de R\$ 4 bilhões. Movido com energia nuclear, o robô tem o tamanho de um utilitário e analisará rochas e solo, para se determinar se o planeta já teve um ambiente capaz de sustentar vida.



Figura 2.3 Robô Curiosity no Laboratório de Jato Propulsão

Fonte: <http://www.nasa.gov>

2.2 Arquiteturas

Antes de abordar a definição de arquitetura, é necessário diferenciar os tipos de robôs. Em (Vieira, 2005) são classificados os sistemas robóticos como robôs manipuladores e robôs móveis. Os robôs móveis necessitam de uma forma de planejar e executar sua trajetória, necessitando portanto, de uma divisão das funções executadas.

Dowling apud (Junior, 2006) se refere a uma arquitetura para robôs como sendo soluções de software e hardware usadas para desenvolver o sistema de controle do robô, incluindo a forma como os componentes da arquitetura se comunicam entre si. Kortenkamp, Bonasso e Murphy apud (Junior, 2006), definem arquitetura como sendo a descrição a partir da qual um sistema é construído por seus componentes básicos e como estes componentes se encaixam formando o todo.

Além de fornecer uma estrutura, a arquitetura também impõe restrições na forma como o problema de controle pode ser resolvido. Hayes-Roth apud (Junior, 2006), se refere a uma arquitetura como sendo os componentes estruturais em que percepção, raciocínio e ação ocorrem.

Em (Junior, 2006) são definidos como componentes básicos em um sistema de controle que definem uma arquitetura:

Percepção – Interpretação e integração dos sensores, modelagem do mundo real e reconhecimento;

Planejamento – Envolve o planejamento de tarefas, sincronização e monitoramento da execução de toda a atividade do robô;

Atuação – Envolve as atividades de execução dos movimentos e ações do robô, e controle dos atuadores.

Uma das formas de se classificar arquitetura está relacionada ao uso de deliberação e reatividade dentro do sistema de controle, sendo divididas em: **deliberativas, reativas e híbridas**.

Deliberação está associada ao processo de tomada de decisão ou planejamento das ações e movimentos do robô utilizando um modelo interno do mundo para que se possa alcançar um determinado objetivo, enquanto que a reatividade está associada à execução de ações pré-definidas em resposta a uma informação sensorial obtida localmente (Junior, 2006).

Em um sistema deliberativo há o sequenciamento de etapas a serem executadas desde a percepção até a ação, sendo a percepção a extração de características ambientais de interesse

a partir das informações sensoriais. Na arquitetura reativa, existe a tentativa de simular a reação imediata e não deliberativa dos animais aos estímulos do ambiente (Santana, 2005). As arquiteturas reativas evitam o uso de um modelo interno do mundo (Junior, 2006).

O projeto de arquitetura de robôs móveis, pode ser encontrado em (Selvatici, 2006).

Uma proposta de arquitetura reativa com detecção de obstáculos usando *Field Programmable Gate Array* (FPGA) pode ser encontrada em (Simões, 2006).

Outro termo presente na área de robótica móvel é o que define o comportamento. Na maioria das vezes, um comportamento se refere a um componente específico, definido dentro da arquitetura. No contexto de arquiteturas reativas, comportamento pode ser definido como sendo uma função ou procedimento que mapeia entradas sensoriais diretamente a padrões de ações motoras usadas para cumprir uma tarefa (Murphy apud Junior, 2006).

Mataric apud (Junior, 2006) define comportamento como sendo uma lei de controle que satisfaz um conjunto de restrições para alcançar ou manter um objetivo em particular.

A saída de um comportamento é enviada para os atuadores do robô ou para outros comportamentos do sistema. Portanto, comportamento é uma função que relaciona estímulos sensoriais a ações produzidas sobre os atuadores do robô, de acordo com um plano realizado a partir de um modelo interno do ambiente (Ribeiro et al, 2001), podendo atuar de maneira isolada ou conjunta, neste último caso, necessitando de uma coordenação.

Em (Junior, 2006) é definida a coordenação de comportamentos em uma arquitetura reativa de 2 formas: competitiva ou cooperativa.

Na **coordenação competitiva** em um dado momento apenas um dos comportamentos prevalece determinando a ação que o robô deve realizar.

Na **coordenação cooperativa** todos os comportamentos contribuem para determinar a ação do robô.

A coordenação define o grau de autonomia do robô em relação ao sistema de locomoção. Em (Jung et al, 2005) definem-se robôs móveis como tendo capacidade de locomoção e de operação de modo semi-autônomo ou completamente autônomo.

Um agente autônomo é um sistema computacional que adquire e analisa dados sensoriais ou estímulos externos e executa um comportamento que produz efeitos no ambiente. Tais sistemas são capazes de lidar com problemas imprevistos, mudanças dinâmicas ou modelos pobres do ambiente (Liu & Wu, 2001).

Em (Romano, 2002) são diferenciados os robôs móveis em dois tipos:

Robô autônomo – É livre para se mover em todas as direções: sua controlabilidade exige um grau de interação com o ambiente;

Robô não-autônomo – Movem-se sobre trajetórias fixas restritas de alguma forma, por exemplo: trilhos, ou seguindo marcas contínuas no solo. Ambos são chamados de veículos auto-guiados (AGV – *Auto Guide Vehicle*).

Um AGV é um robô móvel que usualmente segue uma marcação pré-estabelecida e mesmo se este possui integrada uma capacidade de se deslocar de modo autônomo e de perceber o ambiente, seu sistema de controle depende desta informação para continuar atuando. Os níveis de robustez, autonomia e inteligência de um veículo deste tipo são reduzidos, não sendo capaz de suportar falhas ou de desviar de obstáculos e retornar para a execução de sua tarefa (Jung, 2005).

As consequências das falhas podem ser minimizadas através do uso de técnicas de detecção e isolamento de falhas (DIF) (Tinos, 2003), podendo ser estas de origem mecânica, elétrica ou de software.

Alguns tipos de robôs possuem modelos do ambiente no qual estão inseridos. Junior (2006) classifica o modelo que um robô possui de mundo de duas de formas:

Simbólico – baseado em lógica, tradicionalmente utilizado em inteligência artificial.

Geométrico – o ambiente é representado de forma espacial indicando regiões livres e regiões ocupadas por obstáculos.

Lee (2000), define sistema como uma função que mapeia sinais de seu domínio (entrada) em sinais de uma faixa (saída). O domínio e a faixa são ambos conjuntos de sinais (espaço de sinais). Matematicamente, pode-se modelar sinais e sistemas como funções. Assim, sistemas são funções que operam em funções.

Se a informação contida no modelo deixar de refletir a realidade do mundo, as ações determinadas pelo planejador podem não conseguir executar a tarefa com sucesso (Junior, 2006). Devido à sua forte dependência de um modelo interno do mundo para gerar as ações do robô, as arquiteturas deliberativas são mais adequadas para ambientes praticamente estáticos e muito bem controlados. Em ambientes dinâmicos, o uso de arquiteturas deliberativas pode se tornar proibitivo pois existe a necessidade de se fazer replanejamentos frequentes (Junior, 2006).

Uma grande quantidade de sistemas pode ser caracterizada usando o conceito de estado e a idéia que um sistema evolui através de uma sequência de mudanças no estado (transições de estado). Esta caracterização é chamada de modelo de espaço de estado.

Como o ambiente de trabalho de um robô móvel tem normalmente características variáveis, torna-se mais difícil o seu controle. As condições iniciais podendo variar com o tempo exige que estes sistemas se adaptem a tais mudanças (Simões, 2006), o que leva a um

dos principais problemas: a execução da tarefa de controle em tempo real. A execução em tempo real de sistemas embarcados aplicados a veículos autônomos inteligentes é tratado em (Jung, 2005).

Uma característica desejável de um sistema de controle é sua robustez às variações paramétricas. Um sistema é dito ser robusto se retém satisfatoriamente o desempenho na segurança de erros de modelagem, ruídos e/ou falhas e é dito ser reconfigurável quando sua estrutura ou seus controladores podem ser alterados em resposta às falhas (Tinós, 2003).

Pode-se dividir a arquitetura de robôs móveis em níveis ou camadas, onde cada uma delas desempenha uma função dedicada para a realização de uma tarefa.

A figura 2.4 mostra de forma geral a divisão da arquitetura de um robô móvel.

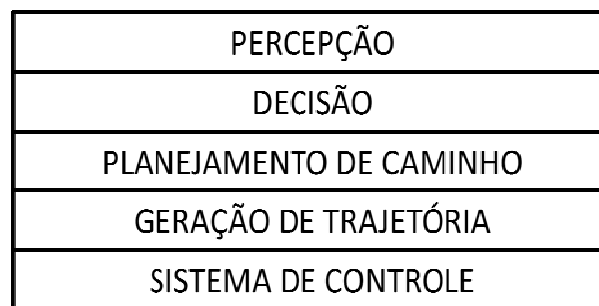


Figura 2.4 Arquitetura geral de robôs móveis

O nível de **percepção** executa a interface entre o robô e o ambiente, sendo composta pelos sensores. O nível de **decisão**, quando presente, executa a tomada de decisões para a execução do trabalho a ser executado. O **planejamento de caminho** usa técnicas para a escolha do melhor caminho que o robô deve seguir, enquanto que a **geração de trajetória** determina as restrições temporais para que o caminho seja executado, como as velocidades e acelerações. O **sistema de controle** gera os sinais para os atuadores.

2.3 O problema de trajetória

Em (Vieira, 2005) divide-se o problema de trajetória de robôs móveis em dois tipos:

Estabilização que consiste em atingir uma configuração final desejada que é definida por uma localização no espaço cartesiano e uma orientação a partir de uma configuração inicial do robô;

Seguimento de trajetória que consiste em fazer com que o robô siga uma trajetória no espaço, iniciando de uma configuração inicial dentro ou fora da trajetória.

Em (Vieira, 2005) é feito um estudo de controle dinâmico de robôs móveis com acionamento diferencial sem considerar a presença de obstáculos, buscando uma estratégia de controle que considera apenas uma parte do estado do robô.

O planejamento de trajetória pode ser abordado de três maneiras (Barreto et al, 1997):

Robótica clássica – Baseia-se na modelagem matemática cinemática (a modelagem dinâmica é menos comum) do robô e do ambiente;

Inteligência Artificial (IA) Clássica – Baseada em algoritmos de busca heurística;

Redes Neurais Artificiais – Usa o modelo cinemático inverso.

As características de movimentação de um robô dependem da disposição e funcionalidade de seus dispositivos de locomoção, que podem restringir seus movimentos ou a realização de trajetórias mais complexas (Vieira, 2005).

Diversos métodos têm sido utilizados no controle de robôs móveis. Em (Bianchi et al, 2001) são definidos três métodos de projetos de robôs inteligentes: os que se baseiam em modelos biológicos, outros que se baseiam em atividade simulada e outros ainda guiados por experimentação. Em (Hoffmann & Silva, 2004) é proposto um controlador cujo modelo usa aprendizado por reforço (não-supervisionado) a fim de evitar colisões. Outros trabalhos propõem o uso de comportamentos reativos para seguir pistas. A grande parte dos trabalhos se refere a robôs dotados de rodas movidos através de um sistema diferencial de tracionamento, resultando em limitações nos movimentos dos mesmos, característica conhecida como não-holonômica.

Em (Vieira, 2005) são definidos como não-holonômicos sistemas com dimensão finita onde algum tipo de restrição é imposta a um ou mais estados do sistema. Robôs móveis que usam acionamento diferencial tem restrições de movimento, como aqueles normais ao plano de suas rodas.

O artigo de (Pedrosa et al, 2003), mostra um método de geração de trajetória para robôs com restrições não-holonômicas, onde uma camada adicional é responsável por transformar um caminho puramente geométrico em outro que respeite as restrições existentes.

Em (Barreto, 1997) são definidas como etapas no controle de trajetória de um robô móvel:

- Planejamento de trajetória, onde se determinam os pontos no espaço de trabalho por onde o robô deve seguir;

- Determinação do modelo cinemático, necessária para se fazer o mapeamento dos pontos em ações de controle;
- Execução do plano, onde se corrige os desvios através de realimentação.

Conclusão do capítulo

O capítulo mostrou conceitos utilizados na robótica móvel que são necessários para se tratar o problema de controle de trajetória de robôs móveis embora, neste trabalho, o enfoque seja dado somente na camada mais baixa da arquitetura, que se refere ao controle. Outras camadas da arquitetura, que se referem à localização e planejamento não foram implementadas.

CAPITULO 3 – Modelagem e Identificação

Este capítulo apresenta o desenvolvimento matemático adotado para a determinação de um modelo cinemático do robô móvel além de conceitos de identificação de sistemas através de dois métodos: Mínimos Quadrados Recursivos e Redes Neurais.

O desenvolvimento matemático do robô foi dividido em duas partes: cinemática e dinâmica dos atuadores através do uso das leis da mecânica. Este modelo desenvolvido para um determinado sistema é apenas uma representação aproximada de algumas características do sistema real.

3.1 Modelagem Matemática

Modelagem matemática é a área do conhecimento que estuda maneiras de desenvolver e implementar modelos matemáticos de sistemas reais (Aguirre, 2004). Uma das formas de se obter um modelo matemático para representar a dinâmica do sistema consiste na utilização das leis da física (Trofino et al, 2003).

A figura 3.1 mostra um esboço utilizado na modelagem do robô móvel. Será utilizada a nomenclatura da tabela 3.1 na determinação do modelo do robô. As coordenadas do robô serão medidas em relação à um referencial fixo na sala onde o mesmo executará seu movimento.

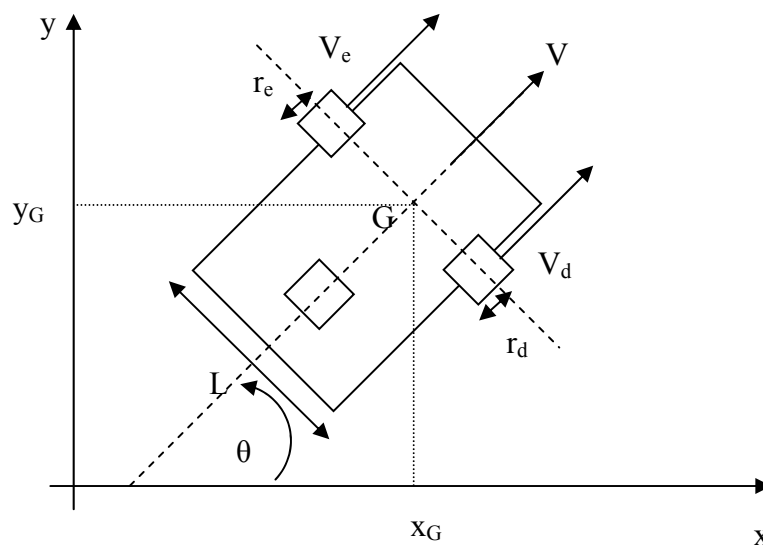


Figura 3.1 Esboço do robô

Tabela 3.1 Simbologia utilizada

Simbologia	Descrição
L	Largura do robô móvel
θ	Ângulo de orientação
v	Velocidade linear do robô
ω	Velocidade angular do robô
r_e	Raio da roda esquerda
r_d	Raio da roda direita
ω_e	Velocidade angular do pneu esquerdo
ω_d	Velocidade angular do pneu direito
f_e	Força atuando no lado esquerdo
f_d	Força atuando no lado direito
τ_e	Torque no lado esquerdo
τ_d	Torque no lado direito
E_b	Força contra-eletromotriz

Em (Chen, 2004) o problema de projeto de sistemas é dividido em: empírico e analítico. No primeiro deles, o projeto é baseado na experimentação, enquanto o segundo consiste de quatro passos: modelagem, determinação das equações matemáticas, análise e projeto.

Os dois primeiros passos estão intimamente relacionados. Dos quatro passos utilizados, a modelagem é o passo mais crítico.

O modelo depende do nível das equações matemáticas usadas. Se estas equações são sofisticadas, o modelo será mais complexo, entretanto mais realista.

Pode-se classificar os modelos de um sistema em:

- **CINEMÁTICOS** – Descrevem o robô em função da velocidade e orientação das rodas;
- **DINÂMICOS** – Descrevem o robô em função das forças generalizadas aplicadas aos atuadores;
- **POSTURA** – Considera como estado a posição e orientação do robô;
- **CONFIGURAÇÃO** – Considera além da postura outras variáveis internas, como o deslocamento angular das rodas.

3.2 Restrições Cinemáticas

Antes da dedução do modelo cinemático a ser utilizado pelo robô, faz-se necessário o conhecimento das restrições de movimento impostas pela estrutura do mesmo. Essas restrições de movimento são chamadas de não-holonômicas.

O robô móvel tem três graus de liberdade, sendo dois deles usados para a posição e um para a orientação.

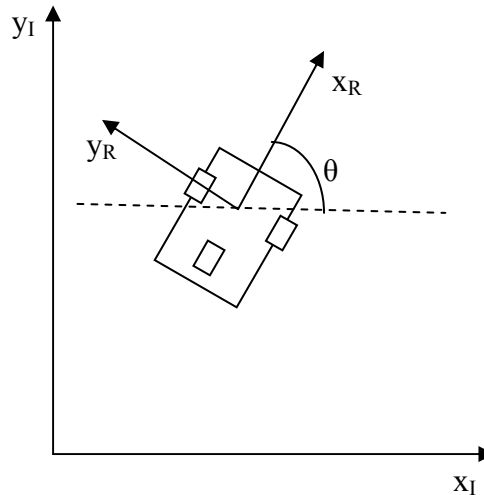


Figura 3.2 Referência local e global

A postura do robô é dada por: $\mathbf{q} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$. Executando a projeção dos eixos de referência local (x_R, y_R) na referência global, obtém-se a matriz de rotação ortogonal, que converte a postura de uma referência em outra.

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Os tipos de rodas utilizadas em robôs móveis são: fixas, orientáveis centralizadas, castor (orientáveis não-centralizadas), suecas e esféricas.

Serão mostradas as restrições de movimento para os tipos de rodas encontradas no robô móvel utilizado neste trabalho. As outras equações podem ser encontradas em (Lages, 1998).

Será considerado que o robô é um corpo rígido e que o movimento ocorre apenas no plano. Ainda será considerado que não há deslizamento.

3.2.1 Roda fixa

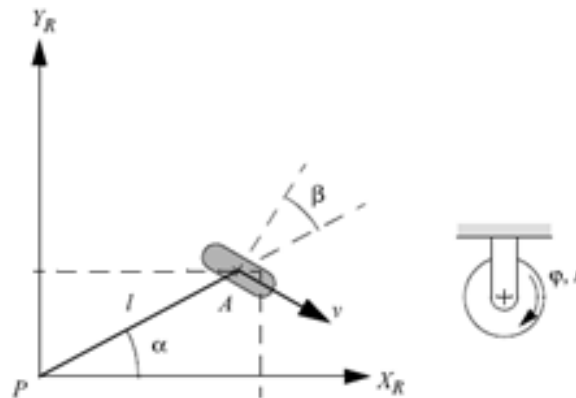


Figura 3.3 Roda Fixa

Fonte: Siegwart & Nourbakhsh (2004)

Como caso geral, supõe-se que a roda está deslocada de um ângulo α em relação ao chassi do robô e que o plano perpendicular à roda esteja deslocado de um ângulo β . Supõe-se, ainda que a distância entre o ponto médio e a roda é l .

Existem duas restrições:

- A primeira é dada pelo conceito de contato de rolamento, ou seja, a roda deve rolar na direção do movimento (sem derrapar). Há um único ponto de contato da roda com o solo e este tem velocidade nula.

$$[\sin(\alpha + \beta) \quad -\cos(\alpha + \beta) \quad -l\cos\beta]\mathbf{R}(\boldsymbol{\theta})\dot{\mathbf{q}} - r\dot{\varphi} = 0 \quad (3.2)$$

Onde o último termo da equação (3.2) é a velocidade linear da roda.

Para o robô móvel utilizado, tem-se $\alpha_D = -90^\circ$ e $\alpha_E = 90^\circ$, enquanto $\beta_D = 180^\circ$ e $\beta_E = 0^\circ$.

- A segunda é dada pelo conceito de não-deslizamento lateral, ou seja, não há movimento perpendicular ao plano da roda.

$$[\cos(\alpha + \beta) \quad \sin(\alpha + \beta) \quad l\sin\beta]\mathbf{R}(\boldsymbol{\theta})\dot{\mathbf{q}} = 0 \quad (3.3)$$

3.2.2 Roda Castor

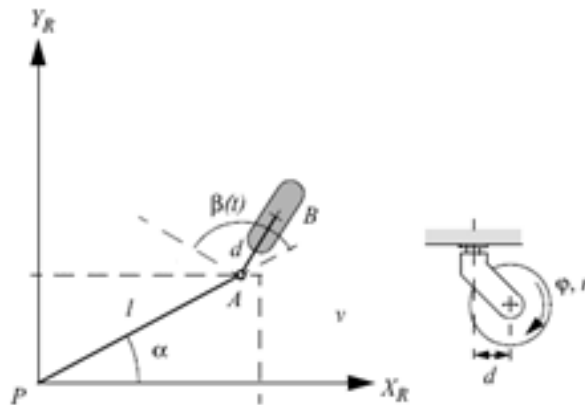


Figura 3.4 Roda Castor

Fonte: Siegwart & Nourbakhsh (2004)

A roda castor pode ser direcionada ao redor de um eixo vertical. A restrição de rolamento é a mesma da roda fixa.

$$[\text{sen}(\alpha + \beta) \quad -\cos(\alpha + \beta) \quad -l\cos\beta]\mathbf{R}(\boldsymbol{\theta})\dot{\mathbf{q}} - r\dot{\phi} = 0 \quad (3.4)$$

Devido à geometria da roda, há uma mudança na restrição de deslize. A força lateral ocorre no ponto A, portanto há uma rotação apropriada no eixo vertical.

$$[\cos(\alpha + \beta) \quad \text{sen}(\alpha + \beta) \quad d + l\text{sen}\beta]\mathbf{R}(\boldsymbol{\theta})\dot{\mathbf{q}} + d\dot{\beta} = 0 \quad (3.5)$$

As equações 3.2 a 3.5 podem ser agrupadas em forma matricial, resultando em:

$$\begin{bmatrix} \mathbf{J}_1(\boldsymbol{\beta}_s) \\ \mathbf{C}_1(\boldsymbol{\beta}_s) \end{bmatrix} \mathbf{R}(\boldsymbol{\theta})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_2\boldsymbol{\Phi} \\ 0 \end{bmatrix} \quad (3.6)$$

$$\begin{bmatrix} [1 & 0 & l] \\ [1 & 0 & -l] \\ [0 & 1 & 0] \end{bmatrix} \mathbf{R}(\boldsymbol{\theta})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_2\boldsymbol{\Phi} \\ 0 \end{bmatrix} \quad (3.7)$$

$$\dot{\mathbf{q}} = \mathbf{R}(\boldsymbol{\theta})^{-1} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{J}_2 \boldsymbol{\Phi} \\ 0 \end{bmatrix} = \mathbf{R}(\boldsymbol{\theta})^{-1} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ \frac{1}{2l} & \frac{-1}{2l} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{J}_2 \boldsymbol{\Phi} \\ 0 \end{bmatrix} \quad (3.8)$$

3.3 Grau de Mobilidade

A restrição de deslize impõe que não deve haver movimento lateral. A equação (3.6) pode ser separada para as rodas fixas e direcionáveis.

$$\begin{bmatrix} \mathbf{C}_{1f} \\ \mathbf{C}_{1s}(\boldsymbol{\beta}_s) \end{bmatrix} \mathbf{R}(\boldsymbol{\theta}) \dot{\mathbf{q}} = 0 \quad (3.9)$$

Definição: O espaço nulo de uma matriz \mathbf{A} é o espaço \mathbf{N} , tal que para qualquer vetor \mathbf{n} em \mathbf{N} , $\mathbf{A}\mathbf{n}=\mathbf{0}$.

Para que as duas restrições sejam satisfeitas, o vetor de movimento $\mathbf{R}(\boldsymbol{\theta})\dot{\mathbf{q}}$, deve pertencer ao espaço nulo da matriz de projeção $\mathbf{C}_1(\boldsymbol{\beta}_s)$. Se as restrições devem ser obedecidas, então o movimento do robô deve sempre estar neste espaço \mathbf{N} .

A cinemática do robô é função do conjunto de restrições independentes de todas as rodas. Esta independência está relacionada ao posto da matriz $\mathbf{C}_1(\boldsymbol{\beta}_s)$. Portanto, $\text{rank}[\mathbf{C}_1(\boldsymbol{\beta}_s)]$ fornece o número de restrições independentes. Quanto maior o número de restrições independentes, mais restrita é a mobilidade do robô.

Para um robô diferencial, as rodas fixas estão alinhadas, assim tem-se para as rodas 1 e 2, que: $\{(l_1 = l_2), (\beta_1 = \beta_2 = 0), (\alpha_1 + \pi = \alpha_2)\}$, de modo que:

$$\mathbf{C}_1(\boldsymbol{\beta}_s) = \mathbf{C}_{1f} = \begin{bmatrix} \cos(\alpha_1) & \text{sen}(\alpha_1) & 0 \\ \cos(\alpha_1 + \pi) & \text{sen}(\alpha_1 + \pi) & 0 \end{bmatrix} \quad (3.10)$$

A matriz da equação (3.10) tem posto unitário. A faixa de valores possíveis do posto da matriz (3.10) é: $0 \leq \text{rank}[\mathbf{C}_1(\boldsymbol{\beta}_s)] \leq 3$. Para o caso onde o posto da matriz é nulo, implica que não há restrições de movimento, ou seja não há rodas fixas no robô. Quando o posto da matriz é igual a 3, implica que o robô tem restrições de movimento em todas as direções, ou seja seu movimento no plano é impossível (robô degenerado).

Definição: O grau de mobilidade do robô é definido por:

$$\delta_m = \dim \mathbf{N}[\mathbf{C}_1(\boldsymbol{\beta}_s)] = 3 - \text{rank}[\mathbf{C}_1(\boldsymbol{\beta}_s)] \quad (3.11)$$

A dimensão do espaço nulo da matriz $\mathbf{C}_1(\boldsymbol{\beta}_s)$ mede o número de graus de liberdade do robô que podem ser manipulados na mudança de velocidade da roda.

Para um robô diferencial, há duas rodas fixas com um eixo comum. Assim uma das rodas não contribui para o aumento do número de restrições cinemáticas ao sistema. Portanto, tem-se: $rank[\mathbf{C}_1(\boldsymbol{\beta}_s)] = 1$, logo: $\delta_m = 2$. Isto implica que o robô diferencial pode manipular sua orientação e velocidade direta/reversa através de mudanças na velocidade das rodas.

3.4 Grau de Direcionabilidade

O grau de mobilidade quantifica o número de graus de liberdade controláveis baseado nas mudanças de velocidade nas rodas. O número de parâmetros direcionáveis independentemente controláveis é o grau de dirigibilidade. Assim: $\delta_s = rank[\mathbf{C}_{1s}(\boldsymbol{\beta}_s)]$.

Um aumento no posto da matriz $\mathbf{C}_{1s}(\boldsymbol{\beta}_s)$ implica maior capacidade de manobras. Pode-se especificar a faixa de valores para a dirigibilidade por: $0 \leq \delta_s \leq 2$.

Para o robô diferencial, $\delta_s = 0$, ou seja, o robô não possui rodas direcionáveis.

Definição: O número de graus de liberdade que um robô pode manipular é chamado de grau de manobrabilidade. Matematicamente, tem-se: $\delta_M = \delta_m + \delta_s$.

Para o robô diferencial, tem-se:

Grau de mobilidade: $\delta_m = 2$

Grau de dirigibilidade: $\delta_s = 0$

Grau de manobrabilidade: $\delta_M = 2$

Define-se a classe de um robô de acordo com o grau de mobilidade e de dirigibilidade (δ_m, δ_s). O robô diferencial pertence à classe (2,0).

O termo holonômico tem aplicação vasta, incluindo equações diferenciais, funções e restrições. Em robótica móvel, o termo se refere a restrições cinemáticas. Um robô holonômico é aquele que não possui restrições não-holonômicas.

Teorema 1: Uma restrição é holonômica se, para qualquer i, j, k , onde $1 \leq i \leq j \leq k \leq n$, tem-se: $A_{ijk} = 0$ (Kuhne, 2005).

$$A_{ijk} = \eta_i \left(\frac{\partial \eta_k}{\partial q_j} - \frac{\partial \eta_j}{\partial q_k} \right) + \eta_j \left(\frac{\partial \eta_i}{\partial q_k} - \frac{\partial \eta_k}{\partial q_i} \right) + \eta_k \left(\frac{\partial \eta_j}{\partial q_i} - \frac{\partial \eta_i}{\partial q_j} \right) \quad (3.12)$$

3.5 Cinemática

Cinemática é o estudo de possíveis movimentos e configurações de um sistema. A dedução de um modelo cinemático não envolve as causas dos movimentos, estudando apenas o movimento em si. Entretanto, para entender como o sistema se moverá em uma dada circunstância é necessário o conhecimento de forças, inércias, energia, etc. (Selig, 1992), caracterizando, portanto o modelo como dinâmico.

Inicialmente serão mostradas as equações usadas na dedução do modelo (ver o apêndice A para a dedução das equações).

A velocidade do móvel é dada por:

$$v = \frac{v_e + v_d}{2} = \frac{\omega_e r_e + \omega_d r_d}{2} \quad (3.13)$$

Sendo v_d e v_e as velocidades lineares dos motores direito e esquerdo.

Se as velocidades v_d e v_e forem diferentes, o robô executará um movimento curvo. Assim, pode-se deduzir que a velocidade angular do móvel será:

$$\omega = \frac{v_d - v_e}{L} = \frac{\omega_d r_d - \omega_e r_e}{L} \quad (3.14)$$

As equações (3.13) e (3.14) podem ser colocadas na forma matricial.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r_d}{2} & \frac{r_e}{2} \\ \frac{r_d}{L} & -\frac{r_e}{L} \end{bmatrix} \begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix} \quad (3.15)$$

$$\mathbf{u} = \mathbf{T}_{\omega}^u \boldsymbol{\omega} \quad (3.16)$$

A figura 3.5 mostra um deslocamento infinitesimal do robô móvel. As velocidades linear e angular se relacionam com as coordenadas, conforme mostrado na figura (3.5).

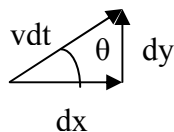


Figura 3.5 Deslocamento infinitesimal

Da figura 3.5, tem-se: $\dot{x} = v \cos \theta$, $\dot{y} = v \sin \theta$, $\dot{\theta} = \omega$ (3.17)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \Rightarrow \dot{\mathbf{q}} = \mathbf{T}_u^q \mathbf{u} \quad (3.18)$$

O vetor $\dot{\mathbf{q}}$ representa as velocidades no espaço de configuração do robô.

Substituindo (3.16) em (3.18), resulta:

$$\dot{\mathbf{q}} = \mathbf{T}_u^q \mathbf{T}_\omega^u \boldsymbol{\omega} \quad (3.19)$$

Da equação (3.17), tem-se:

$$v = \frac{\dot{x}}{\cos \theta} = \frac{\dot{y}}{\sin \theta} \Rightarrow \dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (3.20)$$

Aplicando o Teorema 1 à equação (3.20), tem-se:

$$i = 1; j = 2; k = n = 3. \eta_1 = -\sin \theta; \eta_2 = \cos \theta; \eta_3 = 0. q_1 = x; q_2 = y; q_3 = \theta.$$

$$A_{123} = \eta_1 \left(\frac{\partial \eta_3}{\partial q_2} - \frac{\partial \eta_2}{\partial q_3} \right) + \eta_2 \left(\frac{\partial \eta_1}{\partial q_3} - \frac{\partial \eta_3}{\partial q_1} \right) + \eta_3 \left(\frac{\partial \eta_2}{\partial q_1} - \frac{\partial \eta_1}{\partial q_2} \right) \quad (3.21)$$

$$A_{123} = -\sin \theta \left(0 - \frac{\partial(\cos \theta)}{\partial \theta} \right) + \cos \theta \left(\frac{\partial(-\sin \theta)}{\partial \theta} - 0 \right) + 0 \left(\frac{\partial(\cos \theta)}{\partial x} - \frac{\partial(\sin \theta)}{\partial y} \right) \quad (3.22)$$

$$A_{123} = -\sin \theta \sin \theta - \cos \theta \cos \theta = -(\sin^2 \theta + \cos^2 \theta) = -1 \quad (3.23)$$

O resultado mostra que a restrição representada pela equação (3.20) é não-holonômica.

Outra maneira de verificar a integrabilidade de sistemas pode ser dada pelo teorema de Frobenius (Victorino, 1998).

O sistema $\dot{z} = B(z)u$ é completamente integrável se:

$$Posto(B_1 B_2 \dots B_n) = Posto(B_1 B_2 \dots [B_i, B_j])$$

Onde: B_i , $i=1\dots n$, são as colunas de $B(z)$ e $[B_i, B_j] = \nabla B_j B_i - \nabla B_i B_j$ é o operador colchete de Lie.

Para o robô móvel diferencial, tem-se:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

Onde:

$$B_1 = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$B_3 = \nabla B_2 B_1 - \nabla B_1 B_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & -\sin\theta \\ 0 & 0 & \cos\theta \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$B_3 = \begin{bmatrix} \sin\theta \\ -\cos\theta \\ 0 \end{bmatrix}$$

$$Posto \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ \sin\theta & 0 & -\cos\theta \\ 0 & 1 & 0 \end{bmatrix} = 3, \text{ mas } Posto \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} = 2$$

Como o posto das matrizes são diferentes, o sistema é não-integrável, ou não-holonômico.

3.6 Modelagem dos Atuadores

Foram utilizados como atuadores dois motores de corrente contínua. O tracionamento sendo diferencial e usando caixas de redução.

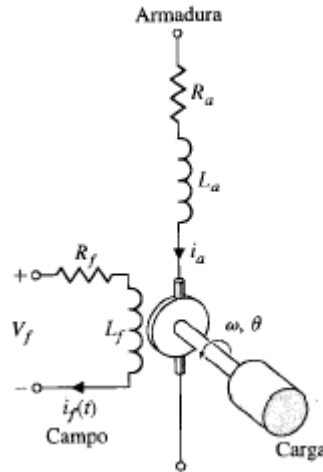


Figura 3.6 Diagrama do motor CC

Fonte: Dorf & Bishop (2004)

A figura 3.6 mostra o esquema de um motor CC. No robô o campo é constante e o controle é feito pela armadura.

Usando a lei de Kirchoff das tensões para a armadura, tem-se:

$$E_a(s) = R_a I_a(s) + sL_a I_a(s) + E_b(s) \quad (3.24)$$

A força contra-eletromotriz é dada por: $E_b(s) = K\omega(s)$

$$E_a(s) = R_a I_a(s) + sL_a I_a(s) + K\omega(s) \quad (3.25)$$

Das equações (3.5) tem-se que:

$$I_a(s) = \frac{E_a(s) - k_d \omega_d(s)}{R_a + sL_a} \quad (3.26)$$

O torque motor de cada um dos atuadores é dado por:

$$\tau_m(s) = k_m I_a(s) \quad (3.27)$$

Cada uma das rodas se comporta como uma carga para os atuadores. Considera-se que os motores são similares, portanto tem parâmetros iguais.

$$\tau_m(s) = Js\omega(s) + B\omega(s) \quad (3.28)$$

$$k_m I_a(s) = Js\omega(s) + B\omega(s) \quad (3.29)$$

$$\frac{k_m(E_a(s) - k_a\omega(s))}{R_a + sL_a} = (Js + B)\omega(s) \quad (3.30)$$

$$\frac{\omega(s)}{E_a(s)} = \frac{k_m}{[(R_a + sL_a)(Js + B) + k_m k_b]} \quad (3.31)$$

Após a discretização da equação (3.31), obtém-se:

$$G_p(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (3.32)$$

Esse modelo expressa a relação entre a velocidade e a tensão aplicada nos motores. A mesma relação é utilizada para os motores direito e esquerdo.

Cabe aos métodos de identificação determinar os valores dos parâmetros do modelo utilizado.

3.7 Modelagem Cinemática

A modelagem matemática do robô móvel envolve o uso de equações não-lineares em sua representação. Existem várias técnicas de linearização deste tipo de sistema.

O sistema pode ser expresso por: $\dot{x} = \sum_{i=1}^m f_i(x)u_i$.

Onde: $x \in R^n$; $u \in R^m$; $m = 2$, $n = 3$

Para o robô móvel, tem-se: $f_1(x) = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}$, $f_2(x) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, $u_1 = v$; $u_2 = \omega$

Teorema 2: Um sistema é localmente linearizável por realimentação de estados em torno de $x=0$, se e somente se, $rank[f_1(0) \ f_2(0)] = n$.

Para um robô com ação diferencial: $rank[f_1(0) \ f_2(0)] = rank \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} = 2$.

Como o rank é menor que 3, a linearização de todos os estados do sistema não-linear não é possível, portanto o estado deve ser modificado. Pode-se representar o estado pela posição e orientação, reduzindo o número de variáveis de estado.

O modelo cinemático linear do robô foi separado em dois sistemas: um de deslocamento linear e outro de deslocamento angular, sendo estes sistemas desacoplados. A entrada do sistema de deslocamento linear deve ter como entrada a velocidade linear e como saída a posição, que será convertida em coordenadas x e y.

O sistema de deslocamento angular tem como entrada a velocidade angular e como saída o ângulo de orientação.

Para a obtenção do modelo a ser utilizado no cálculo dos controladores será feita a dedução matemática para o deslocamento linear. O mesmo procedimento é utilizado para a dedução do modelo do sistema de orientação.

Supõe-se que a figura 3.7 mostre a velocidade de translação do robô móvel em função do tempo.

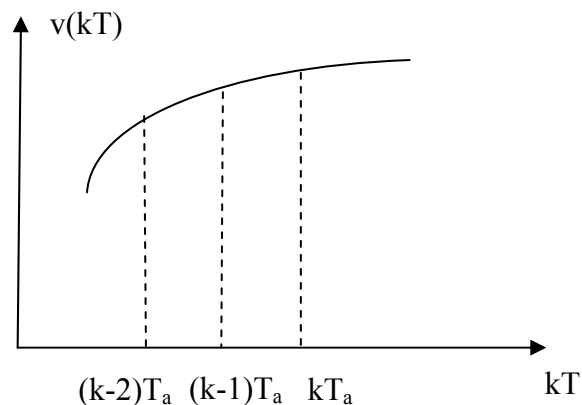


Figura 3.7 Velocidade em função do tempo

Serão feitas aproximações de modo a se obter um sistema discreto.

Supõe-se inicialmente que o robô móvel esteja na posição $S(k_0)$ em $kT_a=0$ e que o sistema seja amostrado a uma taxa de T_a , de modo que a diferença entre dois intervalos de tempo seja constante e igual a T_a .

A posição do móvel no instante $t=kT_a$ será dada por:

$$S(k) = s(k_0) + \sum_{k_0}^k v(k) T_a \quad (3.33)$$

De modo similar a posição em $k-2$ é dada por:

$$S(k-2) = s(k_0) + \sum_{k_0}^{k-2} v(k) T_a \quad (3.34)$$

A equação (3.33) pode ser escrita como:

$$S(k) = s(k_0) + \sum_{k_0}^{k-2} v(k) T_a + \sum_{k-2}^{k-1} v(k) T_a + \sum_{k-1}^k v(k) T_a \quad (3.35)$$

Onde os dois primeiros termos representam a posição em $k-2$. Logo:

$$S(k) = s(k-2) + \sum_{k-2}^{k-1} v(k) T_a + \sum_{k-1}^k v(k) T_a \quad (3.36)$$

A equação (3.36) resulta em:

$$S(k) = s(k-2) + v(k-2)T_a + v(k-1)T_a + v(k-1)T_a + v(k)T_a \quad (3.37)$$

Agrupando os termos de (3.37). resulta:

$$S(k) = s(k-2) + v(k-2)T_a + 2v(k-1)T_a + v(k)T_a \quad (3.38)$$

Pode-se associar parâmetros aos termos da equação 3.38 de modo a se corrigir o modelo de maneira constante. Assim, tem-se:

$$S(k) = a_{2s}s(k-2) + b_{2s}v(k-2) + b_{1s}v(k-1) + v(k)T_a \quad (3.39)$$

A figura 3.7 será repetida por conveniência de modo a se verificar a aproximação linear da velocidade. A aceleração linear entre os instantes $(k-2)T_a$ e kT_a pode ser considerada constante. Se o período de amostragem for suficientemente pequeno, pode-se considerar que a velocidade é constante, dentro de um intervalo entre dois períodos de amostragem. Portanto, pode-se considerar o produto $v(k)T_a$ como a deslocamento entre $(k-1)T_a$ e $(k-2)T_a$. Assim:

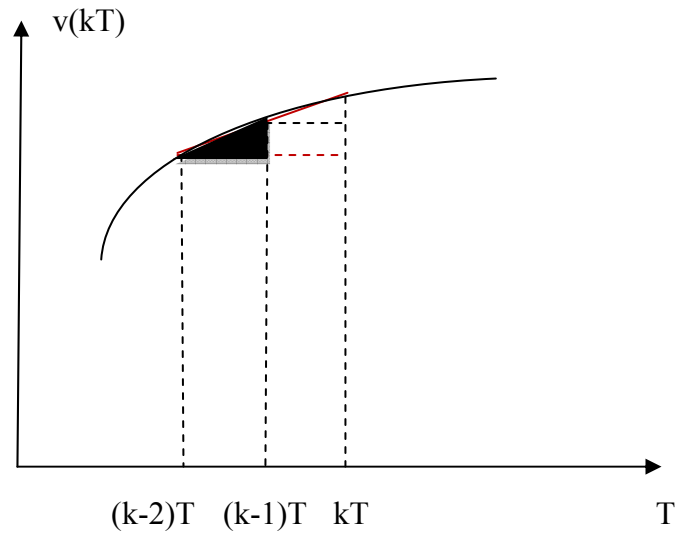


Figura 3.8 Aproximação linear da velocidade

$v(k)T_a = s(k-1) - s(k-2)$, assim o modelo é dado por:

$$s(k) = a_{2s}s(k-2) + b_{2s}v(k-2) + b_{1s}v(k-1) + s(k-1) - s(k-2) \quad (3.40)$$

Pode-se agrupar os termos de mesmo instante de modo que o modelo final é dado por:

$$S(k) = a_{1s}s(k-1) + a_{2s}s(k-2) + b_{1s}v(k-1) + b_{2s}v(k-2) \quad (3.41)$$

A função de transferência discreta entre a posição linear e a velocidade é dada por:

$$G_s(z) = \frac{b_{1s}z^{-1} + b_{2s}z^{-2}}{1 + a_{1s}z^{-1} + a_{2s}z^{-2}} \quad (3.42)$$

O mesmo procedimento feito para o deslocamento angular resulta em:

$$G_\theta(z) = \frac{b_{1\theta}z^{-1} + b_{2\theta}z^{-2}}{1 + a_{1\theta}z^{-1} + a_{2\theta}z^{-2}} \quad (3.43)$$

3.8 Identificação de sistemas

Em (Coelho & Coelho, 2004) são definidas a modelagem e identificação como a determinação do modelo matemático de um sistema representando os seus aspectos essenciais de forma adequada para uma utilização particular.

Quando pouco ou nenhum conhecimento prévio do sistema é necessário, a modelagem é chamada caixa preta ou modelagem empírica. Na modelagem do tipo caixa branca, é necessário conhecer o sistema a ser modelado, além das relações matemáticas que descrevem os fenômenos envolvidos (Aguirre, 2004).

O projeto de sistema de controle deve começar com algum conhecimento do sistema físico ou planta a ser controlada, portanto é necessário conhecer a dinâmica que rege o sistema (Ogata, 2002). Essa dinâmica que envolve o sistema e o ambiente no qual este irá operar, determinará o grau de precisão na execução da tarefa e também o grau de dificuldade na implementação do projeto.

Ao desenvolver o projeto, podem-se desprezar certas características presentes como perturbações, ruídos, etc, gerando assim modelos incompletos; entretanto quanto mais detalhado for o modelo usado para descrever o sistema, mais precisa será a representação deste.

A utilização de um modelo para a simulação do sistema constitui-se um procedimento de baixo custo e seguro para experimentá-lo. A validade dos resultados de simulação depende da qualidade do modelo matemático do sistema (Coelho & Coelho, 2004). Na determinação de alguns tipos de modelos, pode-se reduzir a complexidade do mesmo, desprezando-se as perturbações externas. A dedução de um modelo mais preciso pode exigir considerações que não foram levadas na determinação do modelo inicial, entretanto pode ser vantajoso gastar menos recursos em um projeto de controle robusto usando um modelo incompleto do que desenvolver um modelo de alta fidelidade usando técnicas tradicionais (Haykin, 2002).

Os modelos de um sistema podem ser usados para várias finalidades, dentre elas: previsão, análise e projeto do sistema de controle, supervisão e otimização.

Modelos matemáticos são usualmente empregados em análise e projeto de sistemas de controle e estes são frequentemente imprecisos e sujeitos à variação de parâmetros (Wolovich, 1999).

A representação de um sistema pode ser realizada de vários modos, dentre eles: Função de Transferência, Resposta Impulsiva ou através das equações de estados, sendo que o

modelo é válido nos instantes em que o experimento é realizado. Se a dinâmica do processo muda com o tempo, o modelo pode não ser mais válido (Astrom & Witternmark, 1984).

Modelos paramétricos de baixa ordem dão uma descrição satisfatória do comportamento do sistema real dentro de uma faixa limitada de frequência. Esses modelos baseados na resposta ao degrau são mais precisos em baixas frequências, enquanto modelos baseados na resposta de frequência são mais precisos na frequência do sinal de entrada (Astrom & Witternmark, 1984).

A qualidade da estimação é dependente da natureza do ruído, da estrutura do modelo, do tipo de aplicação e da riqueza da informação contida nas medidas (Coelho & Coelho, 2004). Outro problema de modelagem é a presença de não-linearidades, o que resulta em processos com dinâmica diferente em pontos de operação diferentes. Se a dinâmica muda com o tempo, o controlador deve ser sintonizado para o pior caso. Se a variação do processo é aleatória, pode ser usado controle adaptativo (Astrom & Witternmark, 1984). Sistemas que podem se reconfigurar para atender às variações externas, tem uma aplicabilidade em casos como o estudado nesse trabalho.

O aprendizado adaptativo permite o treinamento em tempo real com o sistema físico provendo as informações para a identificação.

Um dos modelos mais utilizados é o *Controlled Auto-Regressive Intergrated Moving Average* (CARIMA) representado pela equação (3.44).

$$A(z^{-1})\Delta y(t) = z^{-d}B(z^{-1})\Delta u(t) + C(z^{-1})e(t) \quad (3.44)$$

Este modelo assegura uma parte integral na lei de controle digital a fim de proporcionar erro nulo em regime entre a saída e a referência para qualquer *setpoint* e perturbação de carga com comportamento em degrau.

Em Identificação e Controle Adaptativo, os coeficientes dos polinômios $A(z^{-1})$, $B(z^{-1})$ e $C(z^{-1})$ são desconhecidos, podendo ser constantes ou variantes no tempo e devem ser estimados por técnicas de identificação em uma particular aplicação experimental (Coelho & Coelho, 2004).

3.9 Considerações sobre sistemas de controle

Os problemas relacionados ao controle de sistemas envolvem:

- **Análise:** São conhecidos a entrada e o modelo do sistema e se deve obter a saída;
- **Projeto:** Neste tipo de problema são conhecidos o sistema e a saída desejada e se deve obter a entrada que proporcione tal saída;
- **Identificação:** Neste último caso, são conhecidas a entrada e a saída e se deve obter o modelo do sistema;

O objetivo de um sistema de controle é melhorar o desempenho e robustez (Haykin, 2002). Em (Filho, 2004) são definidas como características mais importantes de um sistema de controle: a estabilidade e a robustez, sendo a estabilidade definida como a manutenção de uma planta sob controle, a despeito da ocorrência de perturbações, enquanto que robustez é a manutenção do sistema estável e com um bom desempenho, mesmo que apenas um modelo aproximado da planta seja conhecido, ou ainda que existam variações nos parâmetros da mesma.

O problema chave em Engenharia de Controle é a determinação das variáveis de controle que deveriam ser escolhidas para controlar o processo. Uma das técnicas de controle mais usadas é a realimentação ou *feedback*, entretanto esta técnica é reativa no sentido de que deve haver um erro, antes que as ações de controle sejam tomadas. Outra técnica conhecida como *feedforward* é pró-ativa, porque as ações de controle são tomadas antes que os distúrbios tenham gerado qualquer erro.

Uma das técnicas pró-ativa e utilizada no controle do robô é a preditiva. Neste paradigma de controle, as ações a serem executadas são calculadas em função dos valores passados da entrada e saída. Com base nestes valores e no modelo, são previstos os valores da lei de controle dentro de um horizonte finito (Camacho & Bordons, 1999).

Observa-se que se o sistema tiver uma variação de parâmetros, os valores previstos já não atenderão aos requisitos, uma vez que o modelo do sistema é levado em consideração. Portanto se faz necessário uma constante verificação das mudanças ocorridas no sistema, a fim de que o cálculo contínuo desses parâmetros seja executado.

No controle robusto, o foco é o desenvolvimento de controladores que podem manter um desempenho, mesmo se há somente um modelo pobre da planta ou se há uma variação de parâmetros da mesma.

Do ponto de vista do controle, um sistema é robusto se seu desempenho continua dentro de certos padrões em relação às variações ocorridas nos parâmetros do sistema, ou ainda, um sistema robusto é aquele que retém satisfatoriamente o desempenho na presença de erros de modelagem, ruídos e/ou falhas (Tinós, 2003).

No controle adaptativo, para reduzir os efeitos da variação de parâmetros da planta, a robustez é atingida ajustando-se (adaptando-se) o controlador on-line. Podem ser usados mecanismos adaptativos dentro das leis de controle quando certos parâmetros dentro da dinâmica da planta são desconhecidos. O desenvolvimento de um controle robusto tem como foco manter a estabilidade, mesmo na presença de dinâmica não-modelada ou de distúrbios externos (Haykin, 2002).

Grande parte dos problemas de controle consiste em manter o sistema operando com as variáveis oscilando numa estreita faixa de funcionamento chamada de regime permanente. O sistema opera em regime permanente quando todas as variáveis de estado do sistema se mantêm constantes com o passar do tempo (ponto de equilíbrio) (Trofino et al, 2003).

O controle de robôs está sujeito a incertezas paramétricas e distúrbios externos e estas podem provocar a instabilidade do sistema. O grau de dificuldade aumenta quando são considerados sistemas onde o número de atuadores é menor que o número de graus de liberdade, sendo denominados subatuados (Siqueira, 2004).

Em sistemas lineares, definem-se margem de ganho e de fase de um sistema de malha fechada, como a faixa de variação (determinado pela incerteza) que o sistema pode ter antes de se tornar instável. Em sistemas não-lineares, a estabilidade é investigada pela função de Lyapunov, sendo esta, uma medida simplificada dos objetivos de controle, permitindo que sistemas complexos não-lineares sejam analisados usando uma equação diferencial escalar. O controlador é projetado para levar a função de Lyapunov a zero para atingir os objetivos de controle.

Se uma incerteza tende a levar a função de Lyapunov para distante do zero, deve-se adicionar um termo adicional para estabilizar o algoritmo de controle, tornando o sistema de controle mais robusto (Haykin, 2002).

Pode-se definir as arquiteturas de controle em três categorias:

- **Controle Reativo** – Contém um sistema de reação sensorial-motora simples de implementar não necessitando de muitos recursos computacionais, sendo utilizado para desviar de obstáculos e seguir pistas;
- **Controle Deliberativo ou Cognitivo** – Aplicam mecanismos de planejamento de ações, podendo ser estabelecido um plano prévio de execução de uma sequência de ações baseado nos conhecimentos que o sistema possui sobre o problema a ser resolvido (mapas do ambiente, rotas disponíveis);
- **Controle Hierarquico e Controle Híbrido** – Combina múltiplos módulos de controle reativo e deliberativo.

3.10 Modelo Discreto

O modelo representando os motores CC utilizado é dado pela função de transferência discreta representada pela equação (3.32).

Essa equação foi utilizada baseada na modelagem matemática feita anteriormente para os atuadores. Foram utilizados sinais do tipo *Pseudo Random Binary Signal* (PRBS) na excitação do sistema e coletadas as respostas a essas entradas na verificação da validade do modelo. A figura 3.9 mostra um comparativo entre a resposta real e a obtida através do estimador utilizado no processo de identificação paramétrica. Neste caso foi utilizado o método dos mínimos quadrados recursivo (MQR).

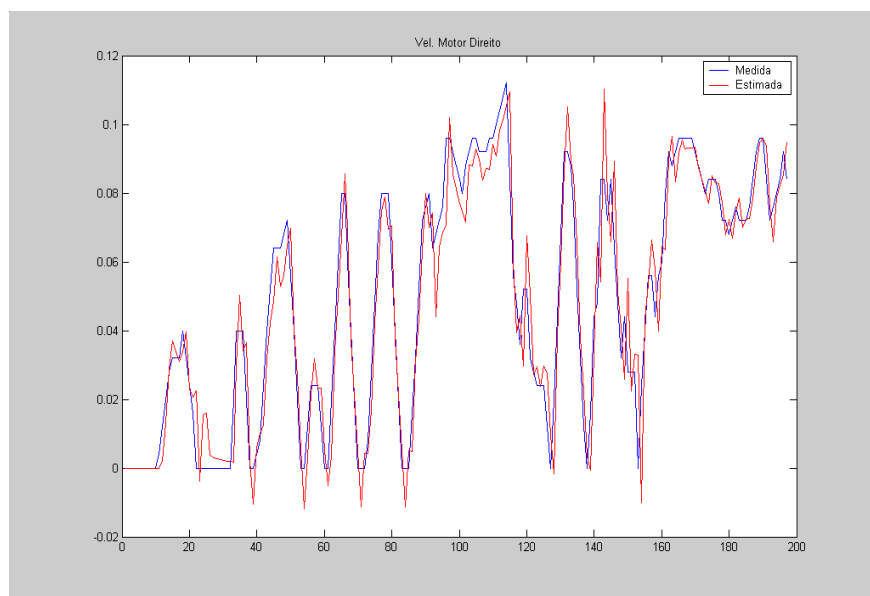


Figura 3.9 Verificação do modelo

Para a verificação da qualidade do modelo foram utilizados dois verificadores: a correlação múltipla e a soma quadrática dos erros (Coelho e Coelho, 2004). A soma quadrática dos erros deve tão pequena quanto possível para expressar uma boa qualidade do modelo, enquanto a correlação múltipla deve ter seu valor próximo da unidade, sendo considerado satisfatório, um valor acima de 0,9.

Este valor representa o valor coletado pela excitação do motor direito do robô móvel. A figura 3.10 mostra que o erro quadrático acumulado em função do número de amostras. Pode-se verificar que o valor máximo ficou abaixo 0.03 dentro de 95 amostras. O cálculo deste indicador é dado pela equação:

$$SEQ = \sum_{k=1}^N [y(k) - \hat{y}(k)]^2 \quad (3.45)$$

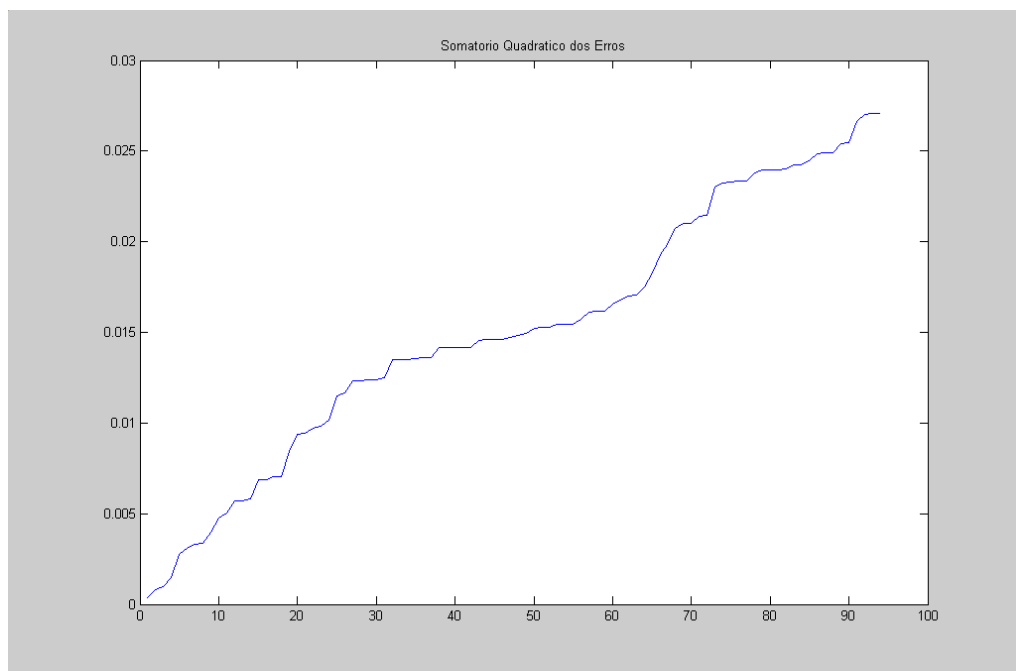


Figura 3.10 Soma quadrática dos erros

A figura 3.11 mostra os valores obtidos pela correlação múltipla em função do número de amostras. O cálculo deste indicador é dado pela equação:

$$R^2 = 1 - \frac{\sum_{k=1}^N [y(k) - \hat{y}(k)]^2}{\sum_{k=1}^N [y(k) - \bar{y}]^2} \quad (3.46)$$

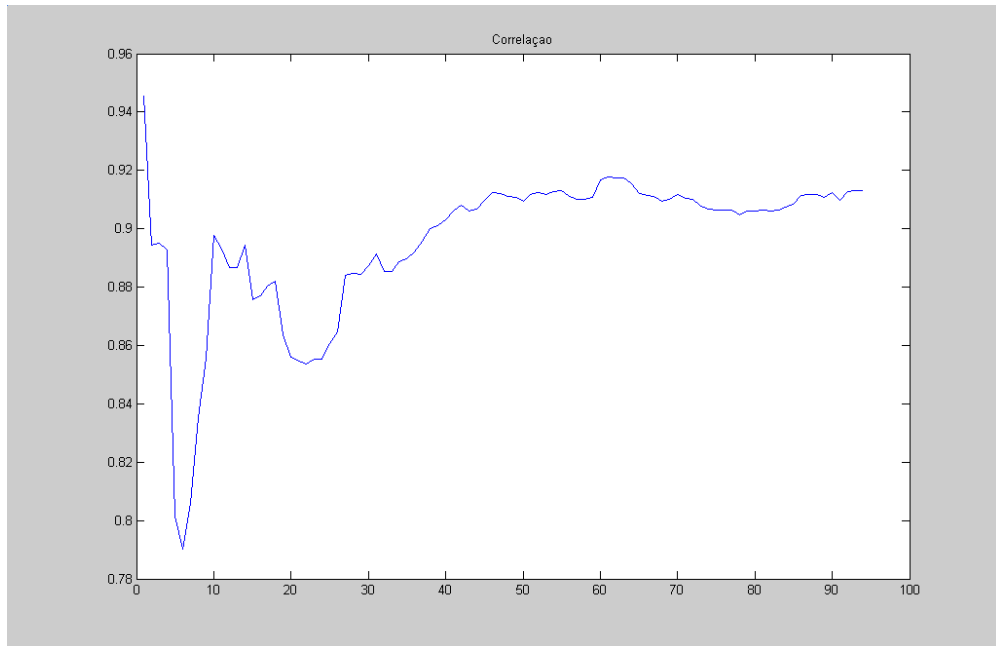


Figura 3.11 Correlação múltipla

3.11 Mínimos Quadrados Recursivo

O método de identificação dos mínimos quadrados recursivos (MQR) surgiu da necessidade de um estimador recursivo, tendo como principal vantagem o rastreamento de parâmetros *on-line*, além de não necessitar de uma grande quantidade de dados armazenados.

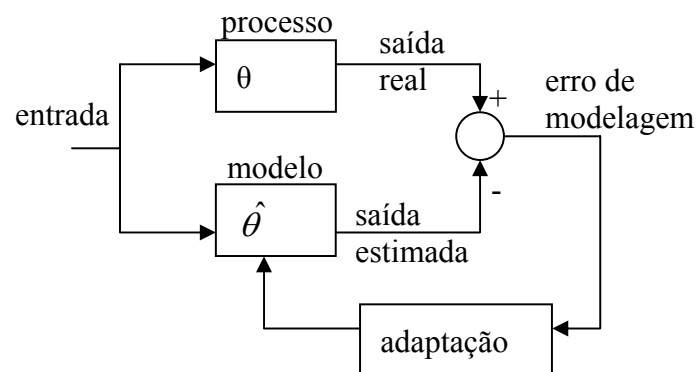


Figura 3.12 Estimação recursiva

A figura 3.12 mostra o procedimento de identificação de parâmetros de maneira recursiva.

Parte-se do modelo do processo. Os dados de entrada são aplicados ao processo e modelo. Medem-se os valores de saída real e calcula-se a saída estimada a partir do modelo, gerando-se o erro de estimação. O objetivo é minimizar o erro estimado, através da correção dos parâmetros do modelo.

A derivação do estimador MQR pode ser encontrada em (Coelho & Coelho, 2004).

Quando os parâmetros do sistema são variantes no tempo, deve-se modificar o estimador para evitar que o ganho se torne pequeno, a fim de manter a capacidade de adaptação do estimador (Coelho & Coelho, 2004).

Foi utilizado o procedimento de busca aleatória (*random walk*) a fim de manter a capacidade de adaptação do estimador. Na técnica de busca aleatória, adiciona-se uma matriz de estrutura diagonal que determina a variação de parâmetros do modelo. Uma descrição detalhada do método pode ser encontrada em (Coelho & Coelho, 2004).

Este método foi utilizado para determinar os parâmetros dos atuadores, bem como do sistema robótico. Alguns dos valores obtidos podem ser encontrados no capítulo 5, sendo que os parâmetros dos atuadores são mantidos quase constantes, devido ao fato de que não ocorre mudança de massa do robô nem variação de piso.

3.12 Redes Neurais

A utilização de redes neurais em sistemas de controle tem crescido bastante nos últimos anos. No controle de sistemas não-lineares elas desempenham papel de aproximadores de funções não-lineares limitadas, tendo a vantagem de aprendizagem que dispensa totalmente ou em parte a necessidade de modelagem do sistema (Carrara, 1999).

O modelo de um neurônio é baseado na neurobiologia. O sistema nervoso é composto de neurônios discretos que se comunicam uns com outros através de sinais elétricos (Tebelskis, 1995).

Um neurônio típico é composto de uma região de entrada, que contém ramificações (dendritos). O corpo do neurônio contém o núcleo e outros componentes. O axônio é usado para transmitir impulsos para outras células (Haykin, 2002).

O cérebro consiste de uma rede de vários neurônios através dos quais impulsos são transmitidos do axônio de um neurônio para os dendritos de outro (Haykin, 2002).

A figura 3.13 mostra a representação de um neurônio humano.

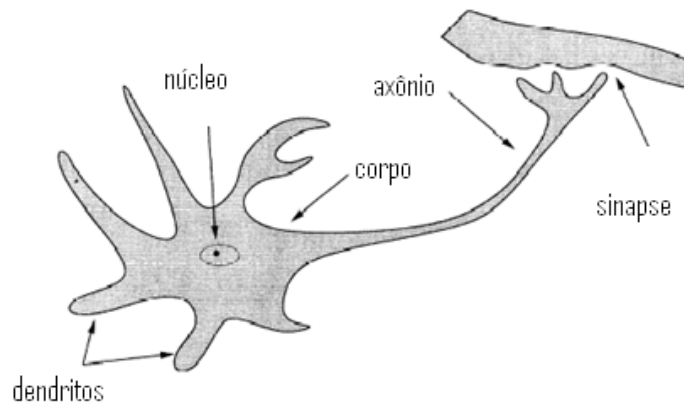


Figura 3.13 Neurônio humano

Em 1943, McCulloch & Pitts propuseram o primeiro modelo computacional para um neurônio. Rosenblatt propôs um algoritmo de aprendizagem para um tipo específico de neurônio conhecido como Perceptron de camada única. Ele demonstrou que o algoritmo sempre convergia para um conjunto de pesos sinápticos que produzia a resposta desejada, entretanto este neurônio era bastante limitado para resolver problemas mais complexos como mostrado em 1969 por Minsky & Papert. O artigo de Minsky & Papert encheu as pesquisas nesta área de certo pessimismo, paralisando-as por cerca de vinte anos.

O interesse em redes neurais foi revivido quando (Hopfield, 1982) propôs uma rede estocástica que pudesse ser treinada, resultando num comportamento desejado. O algoritmo de treinamento conhecido por *backpropagation* foi popularizado por (Rumelhart, 1986), podendo treinar redes multicamadas através da propagação do erro para frente e para trás, adequando os pesos sinápticos da rede.

A figura 3.14 representa o modelo de um neurônio.

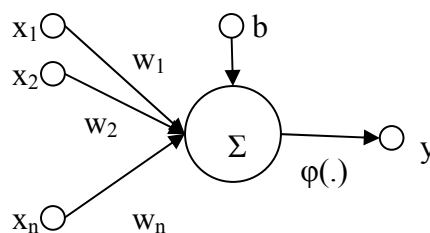


Figura 3.14 Modelo de um neurônio

Onde:

x_i são as entradas externas à rede,

w_i são os pesos sinápticos da rede,

b é o *bias*.

A saída do somador pode ser expressa pela equação matemática:

$$v = \sum_{i=1}^n x_i w_i + b \quad (3.47)$$

O valor v é chamado de campo local induzido. A saída da rede neural pode ser expressa matematicamente por:

$$y = \varphi(v)$$

Sendo φ uma função normalmente não-linear, chamada função de ativação. A mais usada delas é uma função do tipo sigmóide, normalmente uma função tangente hiperbólica, pois esta se caracteriza por ser função ímpar e continuamente diferenciável dentro do intervalo, condição desejável para a convergência do algoritmo de aprendizagem usado pela rede neural (Haykin, 2005).

O uso da função de ativação depende da aplicação. Funções do tipo limiar podem ser usadas para classificação, enquanto as funções do tipo linear podem ser usadas como aproximadores lineares. Funções do tipo sigmóide são usadas em redes *backpropagation*.

Redes que contém múltiplas camadas com funções de ativação diferentes também podem ser usadas, por exemplo: Redes com duas camadas onde a primeira contém uma função sigmóide e a segunda contém uma função linear pode ser treinada para aproximar funções.

Quanto maior o número de camadas, melhor a apresentação da rede, isto é, menor é o erro ao final do processo de treinamento, entretanto a capacidade de generalização, ou seja, a habilidade da rede responder com um erro pequeno em situações na qual não houve treinamento é em geral melhor em redes com poucas ou apenas uma camada oculta.

O número de neurônios ocultos é importante do ponto de vista do grau de aproximação efetuado pela rede (poucos neurônios levam a um erro elevado na saída; muitos neurônios provocam uma oscilação entre os pontos treinados).

Redes Neurais podem ser treinadas para realizar a transformação de uma função de um espaço de entrada para um espaço de saída, sendo duas destas funções: a previsão e a classificação (Tebelskis, 1995).

Uma das características mais importantes de uma rede neural é o poder de generalização (aprendizagem), entretanto para que este aprendizado ocorra, é necessário que sejam apresentados à rede dados que serão usados para o treinamento da mesma. O treinamento significa adaptação dos pesos sinápticos da rede de modo que esta possa exibir um comportamento desejado para os padrões de entrada (Tebelskis,1995). A modificação dos pesos sinápticos é controlada por um valor chamado de taxa de aprendizagem da rede (η).

O treinamento exige múltiplos passos na aprendizagem (iteração ou época).

Os procedimentos de aprendizagem podem ser divididos em:

Supervisionados: Neste tipo de procedimento, existe a presença de um “professor”, que fornece uma saída desejada para cada padrão de entrada. Os pesos são corrigidos através do erro entre a saída real e a desejada.

Não-supervisionados: Neste caso, não há um “professor”. A rede deve encontrar regularidades nos dados de treinamento.

O algoritmo mais utilizado no treinamento de uma rede neural utiliza a retropropagação do erro. Este algoritmo é conhecido como *backpropagation*. Neste algoritmo de aprendizagem, o erro é propagado para frente e para trás na rede a fim de se corrigirem os pesos da mesma. Embora o *backpropagation* possa ser aplicado a uma rede com um número qualquer de neurônios, apenas uma única camada oculta é suficiente para aproximar qualquer função de acordo com o teorema da aproximação universal (Krose, 1996).

Uma das desvantagens do algoritmo de aprendizagem *backpropagation* é o longo tempo de treinamento requerido (para determinados valores da taxa de aprendizagem e momentum), além disso, uma rede complexa tem uma variação grande do erro, podendo esta ficar presa em um mínimo local durante o treinamento, podendo ser resolvido através do aumento no número de camadas ocultas, havendo um limite no número de camadas após o qual o problema volta a ocorrer (Krose, 1996).

Segundo a arquitetura, uma rede neural pode ser de quatro tipos, sendo cada uma delas destinada a uma aplicação (Tebelskis, 1995):

Não-estruturada – São mais utilizadas na busca de padrões;

Redes em camadas – Utilizadas na associação de padrões, isto é: mapeamento de vetores de entrada para vetores de saída;

Redes Recorrentes – Utilizadas no sequenciamento temporal de padrões;

Redes Modulares – Utilizadas na construção de sistemas complexos;

Este último tipo de rede pode ser construído a partir de diferentes tipos de topologias.

Uma rede neural é uma estrutura de natureza estática podendo-se adicionar dinâmica a ela através do uso de uma memória (sendo assim possível guardar os estados da rede) e realimentação. Uma rede deste tipo é chamada de recorrente e a partir dos estados anteriores é possível fazer uma previsão dos estados futuros da mesma.

Enquanto a memória introduz uma variabilidade temporal, a realimentação quando aplicada indevidamente, pode produzir efeitos prejudiciais, levando à instabilidade um sistema que é originalmente estável (Haykin, 2005).

Em sistemas lineares, a estabilidade usa o critério BIBO (*Bounded Input-Bounded Output*), que afirma que a saída de um sistema não deve crescer como resultado de uma entrada, condição inicial ou distúrbios limitados, entretanto como se trata de um sistema dinâmico não-linear, pode se pensar em estabilidade no sentido de Lyapunov.

Foi utilizada neste trabalho uma rede neural do tipo recorrente para identificação dos parâmetros do sistema robótico.

Em (Haykin, 2005) são definidos dois usos funcionais para as redes recorrentes:

- Memórias associativas;
- Redes para mapeamento de entrada-saída.

Uma rede recorrente pode ter os seguintes tipos de arquitetura:

Modelo recorrente de entrada-saída – Este tipo de modelo é mostrado na figura 3.15.

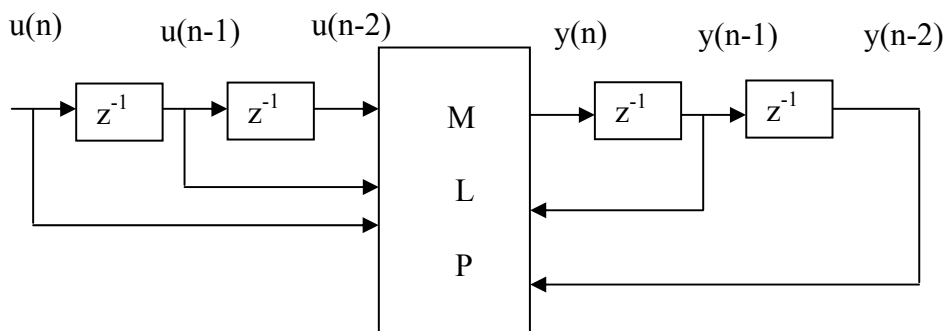


Figura 3.15 Modelo de entrada/saída

A figura 3.15 mostra uma rede multicamadas com dois atrasos unitários na entrada e saída. Este tipo de modelo é conhecido por NARX (*Nonlinear Auto-Regressive with Exogenous Inputs*).

Modelo de Espaço de Estados – Os neurônios ocultos definem o estado da rede. O número de atrasos unitários usados para realimentar a saída da camada oculta de volta para a entrada determina a ordem do modelo.

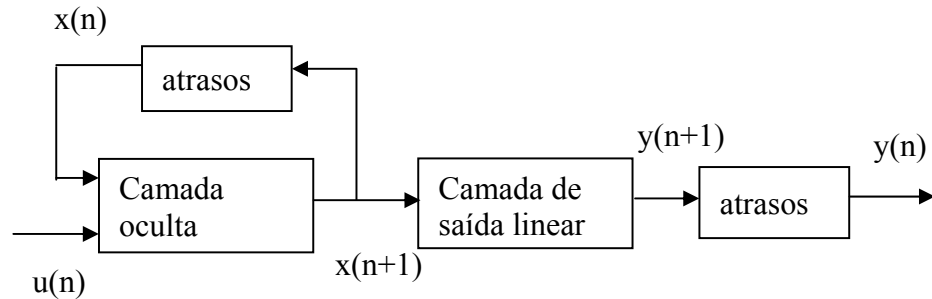


Figura 3.16 Modelo de Espaço de Estados

Conclusão do capítulo

O uso de redes neurais foi utilizado neste trabalho de modo a se obter a identificação paramétrica dos motores de tração do robô. Este método, entretanto, exige um custo computacional mais elevado. Quando o sistema tem uma variação paramétrica, necessita-se que a rede seja treinada para se adaptar às mudanças, tornando o seu uso inviável. Foi dada uma maior ênfase ao algoritmo MQR devido ao fato de o mesmo poder ser corrigido *on-line*.

O capítulo ainda trata sobre as restrições cinemáticas do robô móvel, ferramentas necessárias para se conhecer a mobilidade e dirigibilidade.

CAPÍTULO 4 - Controladores

Este capítulo aborda noções sobre controle preditivo, com foco no GPC, além do controlador PID, finalizando com a apresentação do controle em cascata.

4.1 Controle Preditivo

Nos sistemas de controle de malha fechada, o sinal atuante de erro, que é a diferença entre o sinal de entrada e o sinal de retroação, excita o controlador de modo a reduzir o erro e levar o sinal de saída para o valor desejado. A retroação reduz os efeitos das perturbações e de erros de modelagem ou variação de parâmetros sobre o desempenho de um sistema de controle (Ogata, 2002). A maneira pela qual o controlador automático produz o sinal de controle é chamada ação de controle.

Os controladores analógicos são classificados de acordo com a ação de controle. Alguns tipos de controladores são: Controladores *on-off*, Proporcional, Integral, PI, PD, PID.

Técnicas de controle diferentes podem ser usadas de acordo com o propósito de controle, sendo que algumas delas são mais complexas que outras em termos do algoritmo de controle a ser utilizado.

Técnicas de controle adaptativo podem ser usadas em sistemas que exigem tarefas de controle mais complexas, como parâmetros variáveis. Um dos problemas de aplicação de sistemas adaptativos à robótica é a capacidade dos processadores embarcados. Em (Simões & Globo, 2006) é proposta uma solução através do uso de *Field Programmable Gate Array* (FPGA).

Um requisito normalmente exigido de sistemas embarcados é a execução em tempo real, em especial quando a aplicação envolve veículos autônomos inteligentes (Jung et al, 2005).

O controle preditivo baseado em modelo teve sua origem na indústria de petróleo na década de 70. Em (Richalet et al, 1978) foi utilizado um controlador denominado *Model Predictive Heuristic Control* (MPHC). Em (Cluter & Ramaker, 1979) foram reportados bons resultados obtidos com a aplicação do controlador *Dynamic Matrix Control* (DMC) em uma coluna de pré-flash. Os controladores possuíam um modelo explícito do processo para prever a ação futura da variável manipulada (Gentilin, 2004).

O Controle Preditivo baseado em modelo *Model Predictive Control* (MPC) não designa uma estratégia de controle específica, mas uma faixa ampla de métodos de controle que faz uso de um modelo do processo para obter o sinal de controle minimizando uma função objetivo (Camacho & Bordons, 1999).

Uma das vantagens do controle preditivo é que se a evolução futura da sequência é conhecida a priori, o sistema pode reagir antes que mudanças tenham ocorrido evitando os efeitos dos atrasos na resposta do sistema, entretanto este método exige um esforço computacional maior que o PID.

Uma das áreas onde esta técnica de controle pode ser utilizada é na robótica, entretanto pode ser encontrada na indústria química, térmica e onde os efeitos de atrasos são maiores, além da possibilidade de inclusão de restrições.

Em todas as técnicas de controle preditivo, é necessário o uso do modelo do processo a ser controlado que é usado para prever a saída dentro de um horizonte de controle, calculando a sequência de controle através da minimização de uma função objetivo, além do uso de uma estratégia de horizonte deslizante (*Receding Horizon*), na qual a cada instante o horizonte é deslocado em direção ao futuro, e se aplica o primeiro sinal de controle da sequência calculada a cada passo.

A figura 4.1 mostra a estrutura do MPC:

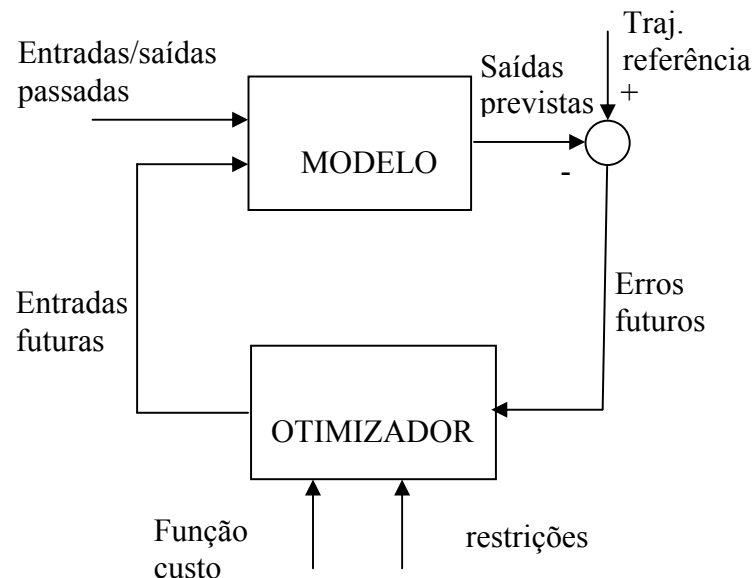


Figura 4.1 Estrutura do MPC

Os vários algoritmos diferem entre si pelo modelo que representa o ruído e o processo, além da função a ser minimizada, sendo utilizados para controlar desde processos com dinâmica simples, até os mais complexos, além de sistemas com atrasos longos ou mesmo instáveis e multivariáveis.

Dentre as técnicas de controle tem-se: *Dynamic Matrix Control* (DMC), *Generalized Predictive Control* (GPC), *Extended Self Adaptive Control* (EPSAC), *Non-linear Extended Self Adaptive Control* (NEPSAC).

Em todos os algoritmos de controle preditivo, as saídas são previstas com base nos valores passados e presentes das entradas e saídas e dos sinais de controle. Os sinais de controle futuros são calculados otimizando um critério de modo a manter a saída do processo tão próxima quanto possível da trajetória de referência.

O modelo desempenha um papel importante no controle preditivo, e este deve ser capaz de capturar a dinâmica do processo, enquanto que o otimizador fornece as ações de controle.

Em processos nos quais as não-linearidades são tão severas e cruciais para a estabilidade do sistema em malha fechada, um modelo linear não é suficiente (Camacho & Bordons, 1999). Em (Torrico & Normey-Rico, 2004) é abordado o controle preditivo de sistemas não-lineares com atraso. O controle de sistemas não-lineares e análise de robustez também podem ser encontrados em outros artigos publicados pelos mesmos autores.

A figura 4.2 mostra um esboço geral do controle preditivo levando-se em consideração perturbações presentes no sistema.

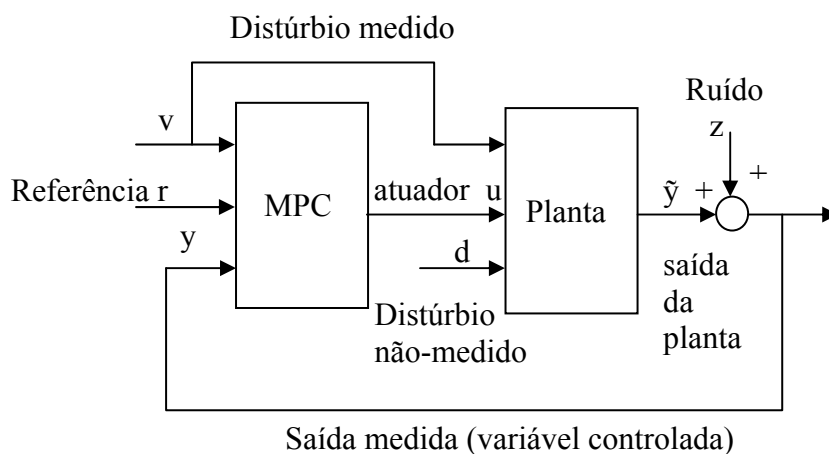


Figura 4.2 Controle geral MPC

O distúrbio não medido \mathbf{d} está sempre presente, representando todos os eventos desconhecidos e não-previstos que afetam a operação da planta, podendo ainda representar a dinâmica não-modelada. O controlador recebe o distúrbio medido \mathbf{v} diretamente, permitindo a compensação do impacto de \mathbf{v} imediatamente ao invés de esperar que o efeito apareça na saída (Controle *feedforward*).

Se restrições forem inseridas na estrutura de controle, procedimentos de busca interativos deverão ser aplicados para que tal função seja minimizada. Por outro lado, se não forem incluídas restrições, o problema de controle admite uma solução mais simples.

O horizonte de controle tem um papel importante no modelo preditivo.

A determinação dos horizontes pode utilizar as regras abaixo:

- 1) Usar um valor relativamente pequeno para o horizonte de controle (de 3 a 5);
- 2) Escolher o intervalo de controle tal que o tempo de acomodação da planta em malha aberta seja de 20 a 30 períodos de amostragem (o período de amostragem deve ser aproximadamente 1/5 da constante de tempo dominante).

4.2 Controle GPC

O controle preditivo *Generalized Predictive Control* (GPC) foi proposto por Clarke sendo um dos métodos mais usados, pois pode lidar com diferentes problemas de controle e um grande número de processos, apresentando bom desempenho e robustez, além de poder ser utilizado com processos instáveis (Camacho & Bordons, 1999).

Devido à estratégia de controle deslizante, o GPC não tem estabilidade garantida em malha fechada. Para solucionar este problema, (Clarke et al, 1991) propõem uma estratégia de controle preditivo com estabilidade em malha fechada garantida. Em (Torrico, 2007) são propostas mudanças no algoritmo para resolver ainda o problema do atraso de transporte.

A estratégia de controle preditivo se baseia em um modelo do sistema a ser controlado. Esse modelo deve relacionar as entradas e saídas do sistema, podendo-se também levar em consideração o erro.

Um dos modelos utilizados para se representar o sistema é o *Controller Auto-Regressive Moving Average* (CARMA) é representado pela equação (4.1).

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + C(z^{-1})e(t) \quad (4.1)$$

Onde: d é o atraso de transporte.

$e(t)$ é um ruído branco de média nula.

E os polinômios são dados por:

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{na} z^{-na} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{nb} z^{-nb} \\ C(z^{-1}) &= 1 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_{nc} z^{-nc} \end{aligned}$$

A idéia do controle MPC é minimizar uma função custo (normalmente quadrática), penalizando o esforço de controle. No GPC esta função é dada por:

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2 \quad (4.2)$$

sendo N_1 e N_2 os horizontes mínimo e máximo e N_u o horizonte de controle.

Em muitas aplicações, onde os distúrbios não são estacionários, deve-se modificar o modelo CARMA dado por (4.1) por outro que inclua um integrador (CARIMA).

Este modelo é representado pela equação (4.2).

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + C(z^{-1})\frac{e(t)}{\Delta} \quad (4.3)$$

Onde: $\Delta = 1 - z^{-1}$.

O objetivo é o cálculo da sequência de controle de modo que a saída esteja o mais próximo possível da trajetória de referência $w(t+j)$.

A otimização da função custo parte da equação Diophantina.

$$1 = E_j(z^{-1})\tilde{A}(z^{-1}) + z^{-j}F_j(z^{-1}) \quad (4.4)$$

Onde:

$$\tilde{A}(z^{-1}) = \Delta A(z^{-1})$$

Após manipulações com as equações (4.2) e (4.4), obtém-se:

$$\hat{y}(t+k|t) = G_j(z^{-1})\Delta u(t+j-d-1) + F_j(z^{-1})y(t) \quad (4.5)$$

A equação (4.5) pode ser escrita, como:

$$y = \mathbf{G}\mathbf{u} + F(z^{-1})y(t) + G'(z^{-1})\Delta u(t-1) \quad (4.6)$$

Os últimos termos da equação (4.6) representam termos em instantes passados, podendo ser agrupados em um termo que representa a resposta livre. Assim a resposta pode ser escrita como:

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{f} \quad (4.7)$$

Considerando-se todas as condições iniciais nulas, a resposta livre também é nula. Se um degrau unitário é aplicado no instante t , então:

$$\Delta u(t)=1, \Delta u(t+1)=0, \dots, \Delta u(t+N-1)=0$$

A sequência prevista de saída é dada pela primeira coluna da matriz \mathbf{G} .

A equação matricial do custo é dada por:

$$\mathbf{J} = (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w})^T (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w}) + \lambda \mathbf{u}^T \mathbf{u} \quad (4.8)$$

Derivando-se a equação (4.8) e igualando-se a zero, obtém-se o valor da ação de controle que minimiza a função custo, obtendo-se:

$$\mathbf{u} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T (\mathbf{w} - \mathbf{f}) \quad (4.9)$$

Como em todas as estratégias de controle preditivo, somente o primeiro elemento do vetor \mathbf{u} é enviado para a planta, enquanto todos os outros são descartados e o processo de cálculo é repetido.

A equação (4.9) pode ser escrita como:

$$\mathbf{u} = \mathbf{K}(\mathbf{w} - \mathbf{f}) \quad (4.10)$$

Sendo que : \mathbf{K} é a primeira coluna da matriz $(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T$;

\mathbf{u} é a variável de controle;

\mathbf{f} é a resposta livre.

A representação da lei de controle do GPC pode ser vista na figura 4.3.

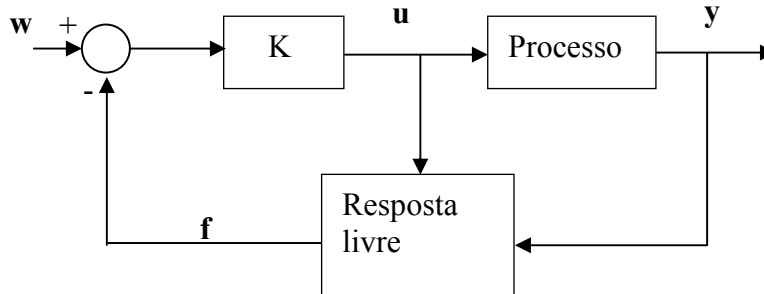


Figura 4.3 Lei de controle GPC

Onde o controlador é representado pelo bloco que contém uma ação de controle proporcional (K).

Considerando a disposição dos atuadores (motores CC) do robô simétrica e supondo que estes têm características idênticas, pode-se desprezar o acoplamento do sistema, de modo a se usar controladores SISO, tornando mais simples o procedimento de controle.

Um dos controladores utilizados neste trabalho foi o GPC, que é baseado na resposta ao degrau. Foram testados controladores GPC no controle de velocidade dos motores CC e no controle cinemático do robô.

No controle de velocidade dos motores, foi utilizado um horizonte de controle fixo no valor $N_u=5$. A dedução da lei de controle MIMO pode ser encontrada em (Camacho & Bordons, 1998).

4.3 Controle PID

O controlador PID é um dos mais utilizados na indústria. É composto das ações de controle Proporcional, Integral e Derivativa.

A lei de controle pode ser expressa por:

$$u(t) = K_p e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \quad (4.11)$$

sendo: $u(t)$ a variável de controle

$e(t)$ a diferença entre a referência e a saída a ser controlada.

A lei de controle discretizada pode ser expressa pela equação:

$$u(k) = u(k-1) + g_0 e(k) + g_1 e(k-1) + g_2 e(k-2) \quad (4.12)$$

A função de transferência correspondente é:

$$G_c(z) = \frac{g_0 + g_1 z^{-1} + g_2 z^{-2}}{1 - z^{-1}} \quad (4.13)$$

O projeto dos parâmetros do controlador é feito por meio do método de alocação de pólos.

A função de transferência dos motores foi obtida no capítulo 3, sendo dada por:

$$G_p(z) = \frac{b_0 z^{-1} + b_1 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (4.14)$$

A equação característica do sistema de controle a malha fechada é dada por:

$$1 + G_c(z)G_p(z) = 0 \quad (4.15)$$

Após a substituição de $G_c(z)$ e $G_p(z)$, tem-se:

$$1 + (a_1 - 1 + g_0 b_0) z^{-1} + (g_0 b_1 + g_1 b_0 + a_2 - a_1) z^{-2} + (b_0 g_2 + g_1 b_1 - a_2) z^{-3} + g_2 b_1 z^{-4} \quad (4.16)$$

A equação característica possui quatro raízes e os parâmetros do controlador são desconhecidos podendo ser determinados através da escolha adequada dos pólos, permitindo que o sistema em malha fechada tenha o comportamento desejado.

Uma descrição detalhada sobre o método de alocação de pólos pode ser encontrada em (Batista, 1985). A utilização de métodos de ajuste de controladores PID através da resposta de frequência pode ser encontrada em (Almeida, 1990).

Como a equação característica tem quatro pólos, estes podem ser alocados considerando dois deles como dominantes. Assim tem-se a equação característica composta por dois polinômios de segunda ordem:

$$D(z^{-1}) = 1 + d_1 z^{-1} + d_2 z^{-2} \quad \text{E} \quad \gamma(z^{-1}) = 1 + \gamma_1 z^{-1} + \gamma_2 z^{-2} \quad (4.17)$$

Sendo $D(z^{-1})$ responsável pelos pólos dominantes e $\gamma(z^{-1})$ pelos pólos não-dominantes.

Como se deseja um comportamento de segunda ordem em malha fechada, dos quatro pólos existentes, dois poderão ser definidos pelas especificações do usuário, enquanto os outros dois deverão ter pouca influência (próximo da origem) ou serem cancelados, assim a resposta do sistema fica definida pelos pólos distantes da origem (Batista, 1985).

Os pólos dominantes podem ser determinados por:

$$d_1 = -2e^{-\zeta\omega T_a} \cos(\omega T_a \sqrt{1 - \zeta^2}) \quad \text{e} \quad d_2 = e^{-2\zeta\omega T_a} \quad (4.18)$$

Os valores de ζ e ω são determinados pela característica da resposta desejada, sendo utilizado o tempo de acomodação e a ultrapassagem percentual, dadas por:

$$T_s = \frac{4}{\zeta\omega} \quad \text{e} \quad P_m = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (4.19)$$

Foram obtidos para os parâmetros: $\zeta=0.8$, $\omega=6.2$ rad/s, sendo que $T_a=0.25$ seg.

Uma descrição das características de resposta em sistema de controle pode ser encontrada em (Dorf & Bishop, 2004).

O produto $D(z^{-1})\gamma(z^{-1})$ resulta em:

$$(1 + d_1 z^{-1} + d_2 z^{-2})(1 + \gamma_1 z^{-1} + \gamma_2 z^{-2}) \quad (4.20)$$

$$1 + (\gamma_1 + d_1)z^{-1} + (\gamma_2 + d_1\gamma_1 + d_2)z^{-2} + (d_1\gamma_2 + d_2\gamma_2)z^{-3} + d_2\gamma_2 z^{-4} \quad (4.21)$$

Igualando-se as equações 4.16 e 4.21, resulta em:

$$\gamma_1 + d_1 = a_1 + b_0 g_0 - 1 \quad (4.22)$$

$$\gamma_2 + d_1 \gamma_1 + d_2 = a_2 - a_1 + b_0 g_1 + b_1 g_0 \quad (4.23)$$

$$d_1 \gamma_2 + d_2 \gamma_1 = -a_2 + b_0 g_2 + b_1 g_1 \quad (4.24)$$

$$d_2 \gamma_2 = b_1 g_2 \quad (4.25)$$

Os parâmetros do controlador podem ser obtidos.

$$g_2 = \frac{d_2 \gamma_2}{b_1} \quad (4.26)$$

$$g_0 = \frac{1 + \gamma_1 + d_1 - a_1}{b_0} \quad (4.27)$$

$$g_1 = \frac{d_1 \gamma_2 + d_2 \gamma_1 + a_2 - b_0 g_2}{b_1} \quad (4.28)$$

Partindo-se da equação (4.23) e usando-se as equações (4.27) e (4.28), pode-se obter valores de γ_2 em função de γ_1 . Obtendo-se:

$$\gamma_2 = K_1 \gamma_1 + K_2 \quad (4.29)$$

Onde:

$$K_1 = \frac{b_1}{b_0} \text{ e } K_2 = \frac{-[a_2 + \frac{b_0 a_2}{b_1} + \frac{b_1}{b_0} (d_1 - a_1 + 1) - a_1 - d_2]}{\frac{b_0 d_1}{b_1} - \frac{b_0^2 d_2}{b_1^2} - 1} \quad (4.30)$$

A equação (4.29) representa o mapeamento do plano z (círculo de raio unitário), para outro plano representado por retas, onde a região de estabilidade está contida na área de intersecção entre as retas representadas por (4.29). O desenvolvimento do método pode ser encontrado em (Batista, 1985).

Ficando assim determinados os parâmetros do controlador em função dos parâmetros da planta.

4.4 Controle em Cascata GPC-PID

Os controladores em cascata podem ser utilizados em sistemas onde se tem dinâmicas diferentes entre partes do processo a ser controlado. No caso dos robôs móveis, os motores CC têm uma dinâmica mais rápida comparada à mecânica do robô (devido a parâmetros como a massa e inércia).

Com o intuito de suavizar a mudança de set-point necessária para a execução da trajetória, foi utilizado um esquema de controle em cascata, onde um controlador GPC fornece a saída prevista que será utilizada como referência pelo controlador PID. As entradas e saídas aplicadas aos motores serão utilizadas para a correção do modelo necessária para manter a adaptatividade do sistema. A figura 4.4 mostra a proposta de controle em cascata GPC-PID.

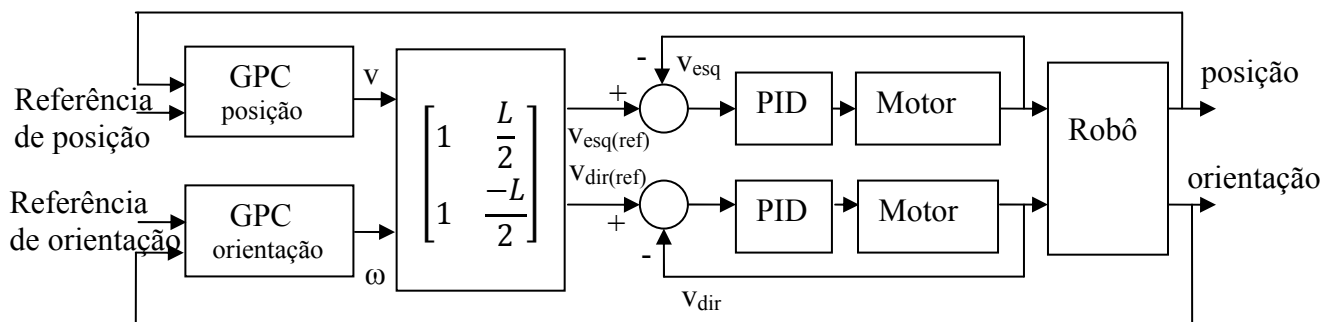


Figura 4.4 Controle em cascata GPC-PID

Os controladores GPC do robô móvel foram desenvolvidos para sistemas *Single Input Single Output* (SISO) devido às características do robô, têm-se um acoplamento fraco. Foi também desenvolvido um controlador *Multiple Input Multiple Output* (MIMO), de modo a ser verificado o efeito do acoplamento desprezado no uso dos controladores SISO.

O modelo do robô é atualizado através das entradas e saídas passadas. Os parâmetros de atualização são corrigidos pelo algoritmo MQR. Com o modelo corrigido, a função custo é minimizada, gerando as entradas de controle (velocidade linear e angular). Esses valores de entrada de controle calculados servirão como referência a serem aplicadas aos motores. A vantagem no uso da combinação dos

controles é que mudanças tipo degrau na referência podem causar mal funcionamento do controlador PID, sendo estas mudanças suavizadas pela presença do GPC.

Foram utilizados como sensores de velocidade *encoders*, sendo estes instalados diretamente nas rodas do robô móvel, portanto o controlador GPC fornecerá as referências de velocidade que serão utilizadas pelos controladores PID.

Em (Pereira, 2000) são descritas as vantagens do uso deste tipo de controle em cascata e sua aplicação a micro-robôs móveis.

Conclusão do capítulo

Foi abordado o sistema de controle em cascata e em ambas as malhas (interna e externa) foram feitos testes com os controladores PID e GPC. Foi ainda utilizado um controlador GPC multivariável de modo a ser verificar o efeito do acoplamento entre as tensões aplicadas nos motores do robô. Os resultados obtidos podem ser vistos no capítulo 6.

CAPÍTULO 5 – O Robô Móvel

Este capítulo mostra algumas características do robô móvel e apresenta as mensagens trocadas entre o robô e o sistema de controle.

5.1 O Robô Móvel

A figura 5.1 mostra uma fotografia do robô móvel desenvolvido pelo Grupo de Pesquisas em Automação e Robótica (GPAR) do Departamento de Engenharia Elétrica.

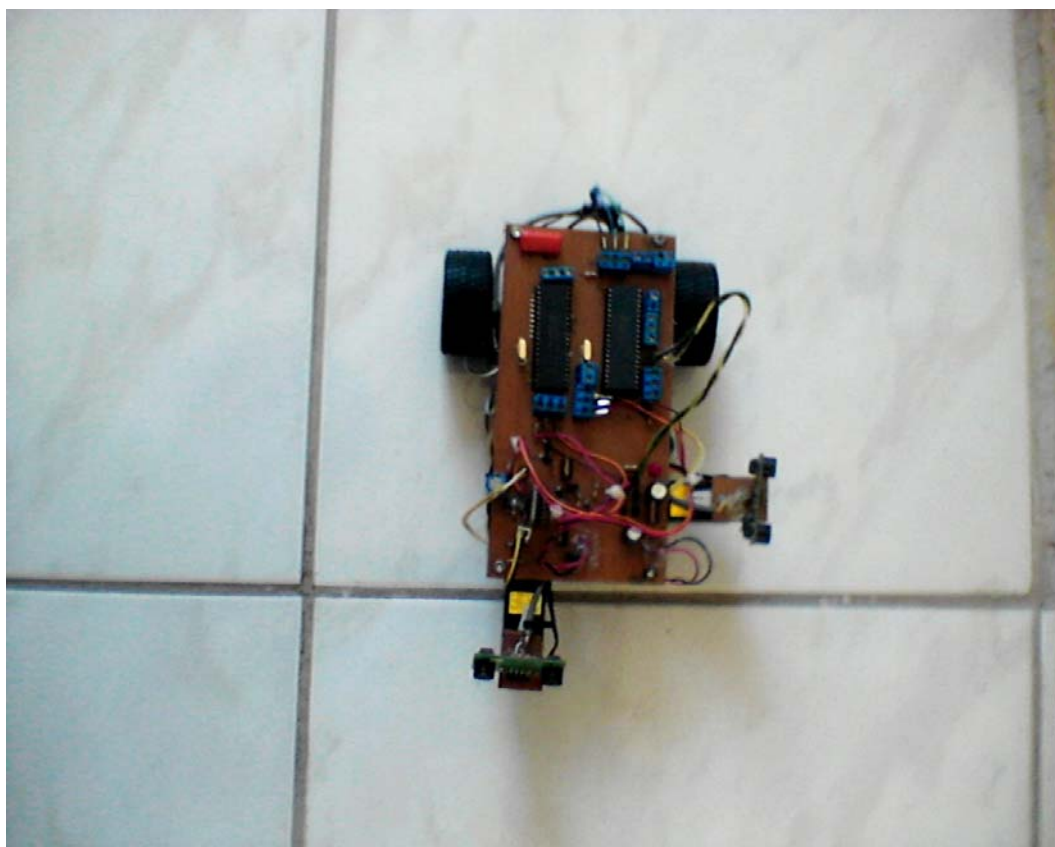


Figura 5.1 Robô móvel GPAR

O apêndice B mostra os diagramas esquemáticos dos circuitos do robô móvel GPAR. Os circuitos foram divididos em três partes principais para facilidade de desenvolvimento e testes.

As principais partes do robô são:

- **Fonte** – Fornece as tensões para o funcionamento do robô;
- **Controle** – Composto pelos micro-controladores que coletam dados dos sensores e enviam sinais de controle para os atuadores.
- **Driver de potência** – Contém o circuito de potência para fornecer a corrente de acionamento dos motores de tração.

O robô móvel GPAR, tem as seguintes características:

Tabela 5.1 Características do robô móvel

Largura	12 cm
Comprimento	20 cm
Tração	Diferencial
Atuadores	Motores CC (uso de caixas de redução)
Microcontrolador I (PIC 16F877A)	Funções de acionamento e comunicação
Microcontrolador II (PIC 16F877A)	Coleta de dados de posição
Comunicação entre microcontroladores	Rede I2C
Sensores de posição	SONARES
Sensores de velocidade	ENCODERS
Comunicação controlador-robô	Serial (Assíncrona)
Trajectoria	Programável

Os motores de tração são de corrente contínua. O robô executa as ações de controle, e coleta os dados de velocidade obtidos após cada valor de tensão colocado na entrada dos motores. Esse conjunto de dados é enviado ao programa central, onde são calculados os parâmetros de controle do robô móvel. Estes cálculos são repetidos até que a trajetória seja finalizada.

A velocidade de rotação é determinada através dos *ENCODERS* (sensores proprioceptivos), montados nos eixos dos motores de tração. Estes sensores enviam pulsos aos pinos do microcontrolador gerando interrupções por mudança de estado, fazendo com que o microcontrolador incremente a contagem dos mesmos. A cada intervalo de amostragem (0,25 seg) é verificado o valor da contagem, determinando a velocidade dos motores.

O sentido de rotação dos motores de tração do robô móvel pode ser invertido para execução de manobras. O robô não está preparado para lidar com obstáculos, pois isto implicaria mudanças na trajetória programada. O sentido de rotação dos motores é determinado por pontes H e o controle de velocidade utiliza Modulação por Largura de Pulso (*Pulse Width Modulation* – PWM).

Foi necessário a utilização de caixas de redução nas duas rodas do robô móvel de modo a permitir que o mesmo tivesse torque necessário para executar o movimento.

A figura 5.2 mostra o sistema de redução utilizado no robô GPAR. Podem-se ver os sensores (*encoders*) utilizados para medição de velocidades das rodas. A caixa de redução utilizada permite que se escolha uma dentre quatro relações de redução. O robô foi montado com a relação de 114,7:1, com torque máximo de 809 gf.cm e frequência de rotação máxima de 115 RPM.



Figura 5.2 Sistema de redução

O robô está sujeito a diversos erros. Podem-se classificar os erros em dois tipos:

Sistemáticos - Imprecisões no modelo cinemático devido a diferenças entre os raios dos pneus ou mesmo desalinhamento entre estes, bem como imprecisão dos *encoders*.

Não-sistemáticos - Também têm sua contribuição na imprecisão do sistema. Exemplos deste tipo de erro são as irregularidades no piso, ou deslizamento das rodas.

A localização do robô é obtida a cada período de amostragem pelos sonares instalados. A utilização dos sonares é necessária de modo a se evitar problemas de

dead-reckoning, onde a posição é determinada através dos deslocamentos obtidos pelos *encoders* a cada período de amostragem, gerando um erro crescente.

O envio de dados a um computador, é necessário, pois a estrutura do robô foi concebida a fim de que os cálculos de velocidade e trajetória sejam realizados externamente a este, uma vez que alguns métodos de controle exigem processamento elevado, sendo difícil sua implementação em microcontroladores, deixando para estes as funções de execução e tarefas de coleta de dados.

Devido à arquitetura escolhida para o projeto, é necessária a troca de dados entre o controlador central e o móvel, portanto foi especificado um protocolo simples para definir a execução de tarefas pelo robô. Os dados são enviados e recebidos como uma cadeia de bits de modo assíncrono, sendo necessária uma coordenação no envio e recepção destes dados.

A comunicação entre o robô e o controlador é feita através de uma interface serial, sendo utilizado um rádio a fim de se evitar a interligação através de cabo.

A figura 5.3 mostra o esboço dos sub-sistemas do robô móvel.

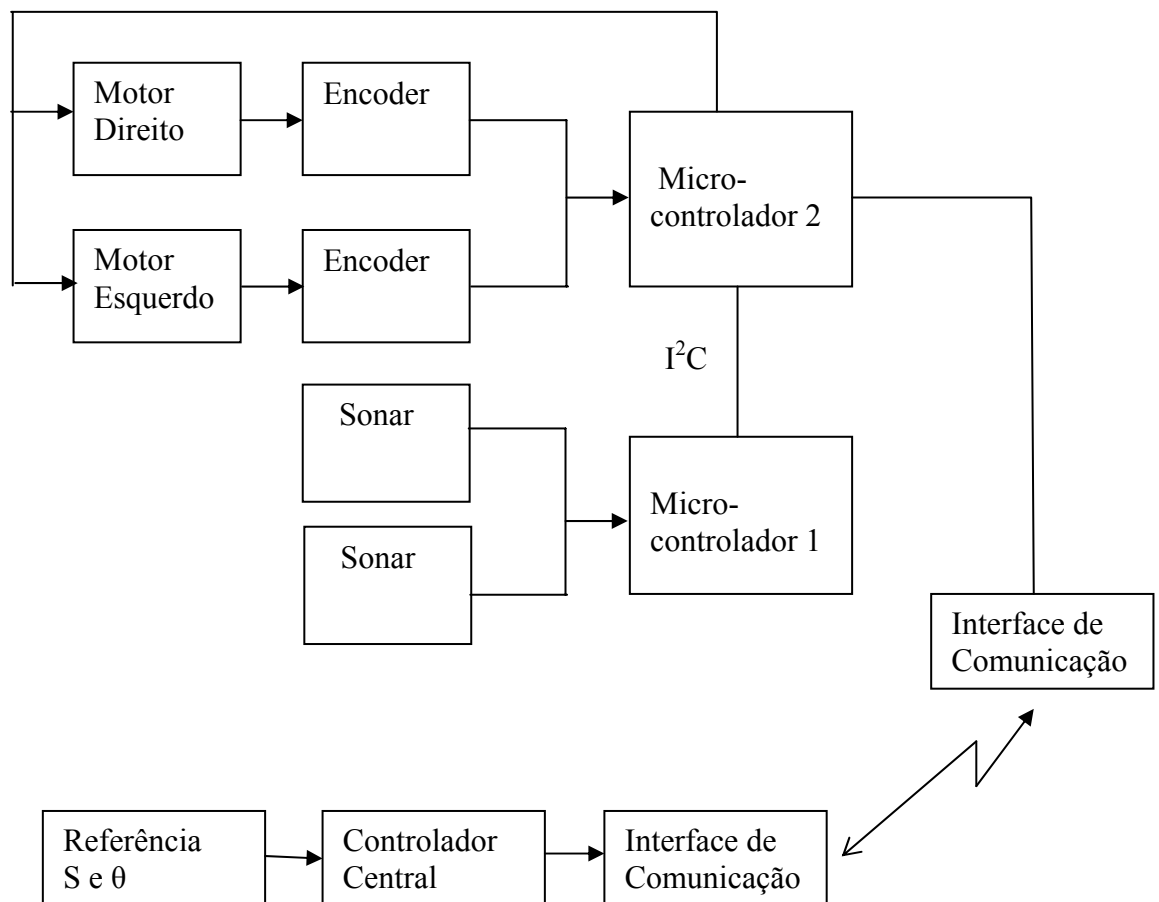


Figura 5.3 Sub-sistemas do robô móvel

5.2 Mensagens de comunicação

É necessário que haja um conjunto de regras bem-definido entre duas partes a fim de se estabelecer uma comunicação entre elas. Esse conjunto de regras é definido como um protocolo de comunicação.

Foi criado um protocolo simples para ativar a comunicação entre o robô móvel e o controlador central. Este protocolo irá determinar as regras de comunicação entre as duas partes, facilitando o envio e recebimento de dados por ambos.

O protocolo é *half-duplex* (semi-duplex), ou seja, a transmissão de informação é bidirecional, entretanto não ocorre em ambas as direções simultaneamente.

A Tabela 5.2 define os parâmetros usados na comunicação entre as partes:

Tabela 5.2 Parâmetros de comunicação

Tipo de Comunicação	Assíncrona
Taxa de transmissão	9600 bps
Bit de dados	8
Paridade	Sem paridade
Start bit	1
Stop Bit	1

É permitido o envio de mensagens de teste para o robô, além da execução de funções de controle. Não há checagem de erro, pois o sistema é executado em tempo-real, não havendo a possibilidade de correção destes erros dentro do período de amostragem.

Embora a comunicação entre o controlador e o robô seja assíncrona, a interação entre eles é síncrona, pois o atraso é desprezível (Junior, 2006). Em robôs sondas que exploram outros planetas, por exemplo, essa interação é assíncrona, pois o controlador e o robô não compartilham o mesmo ambiente, havendo um atraso considerável entre o envio de comandos e o recebimento de respostas.

É necessário ainda definir mensagens trocadas entre o controlador e o robô móvel. Essas mensagens devem ser entendidas pelas entidades que compõem o sistema, tendo sido divididas em duas categorias: as originadas pelo sistema de controle e as originadas pelo sistema móvel, conforme especificadas nas seções seguintes.

5.2.1 Mensagens originadas pelo controlador

As mensagens enviadas pelo controlador ao robô móvel pode ser uma dentre as descritas na tabela 5.3. Todas as mensagens fazem com que o robô execute uma tarefa específica. Se for desejado que o robô siga para a frente em linha reta com velocidade máxima, o sistema de controle deve executar os seguintes passos:

1. Acionar o motor esquerdo no sentido horário (envia a mensagem 0000);
2. Acionar o motor direito no sentido horário (envia a mensagem 1111);
3. Aplicar ao motor esquerdo a tensão máxima (envia a mensagem 2556), pois 255 é o valor máximo a ser aplicado ao PWM do motor;
4. Aplicar ao motor direito a tensão máxima (envia a mensagem 2557).

Deve-se observar que o valor de tensão aplicado aos PWM dos motores varia de 0 (mínimo) a 255 (máximo).

Nas mensagens XXX6 e YYY7, os valores XXX e YYY correspondem aos valores de *duty cycle* que devem ser aplicados aos PWM para controle de velocidade.

Na mensagem ZZZ8, ZZZ indica o valor do *duty cycle* a ser aplicado aos servos a fim de posicionar os sonares perpendiculares à referência para medição da posição.

Tabela 5.3 Mensagens originadas pelo controlador

Mensagem	Ação executada
0000	Aciona o motor esquerdo no sentido horário
1111	Aciona o motor direito no sentido horário
2222	Aciona o motor esquerdo no sentido anti-horário
3333	Aciona o motor direito no sentido anti-horário
4444	Pára o motor esquerdo
5555	Pára o motor direito
XXX6	Valor da entrada de controle a ser aplicada no motor esquerdo
YYY7	Valor da entrada de controle a ser aplicada no motor direito
ZZZ8	Posiciona o servomotor dos sonares
9999	Ativa robô

5.2.2 Mensagens originadas pelo robô

Essas mensagens são originadas pelo robô a cada período de amostragem a partir do instante em que o mesmo for ativado pelo controlador (através do comando 9999).

VVV;EEE;DDD;XXX;YYY; – Os valores indicam as velocidades dos motores e as coordenadas do robô.

EEE e DDD correspondem ao número de interrupções geradas pelos ENCODERS acoplados aos eixos dos motores esquerdo e direito, respectivamente.

XXX e YYY correspondem aos valores dos contadores que indicam o tempo de ida e retorno dos pulsos gerados pelos SONARES, utilizados para medir as coordenadas X e Y.

Os *encoders* montados nas rodas do robô geram pulsos (interrupções) para o microcontrolador determinar a velocidade de cada motor. O valor correspondente a uma volta é 44 pulsos.

Os sonares utilizados podem medir distâncias de até 3 metros. O microcontrolador mede a distância ao referencial ativando o sonar e um temporizador que mede o tempo que o pulso de ultra-som leva para ir e voltar até o referencial. A velocidade do som no ar é de aproximadamente 341 m/s.

O tempo máximo de ida e volta do pulso é, portanto:

$$t_{max} = \frac{6}{341} = 0,018 \text{ segundos}$$

Como exemplo, uma mensagem do tipo VVV;025;050;080;090; indica para o controlador:

$$\text{Motor esquerdo: } V_e = \frac{\text{contagem} * 2\pi r_e}{T_a * 44} = \frac{25 * 6,28 * 0,024}{0,25 * 44} = 0,34 \text{ m/s,}$$

$$\text{Motor direito: } V_d = \frac{\text{contagem} * 2\pi r_d}{T_a * 44} = \frac{50 * 6,28 * 0,024}{0,25 * 44} = 0,69 \text{ m/s,}$$

$$\text{Abscissa: } X = \frac{\text{contagem} * V_{som} * t_{max}}{255 * 2} = \frac{80 * 341 * 0,018}{2 * 255} = 0,96 \text{ m,}$$

$$\text{Ordenada: } Y = \frac{\text{contagem} * V_{som} * t_{max}}{255 * 2} = \frac{90 * 341 * 0,018}{2 * 255} = 1,08 \text{ m.}$$

5.3 Algoritmo de execução da trajetória

O fluxograma na figura 5.4 mostra a estrutura do programa de controle.

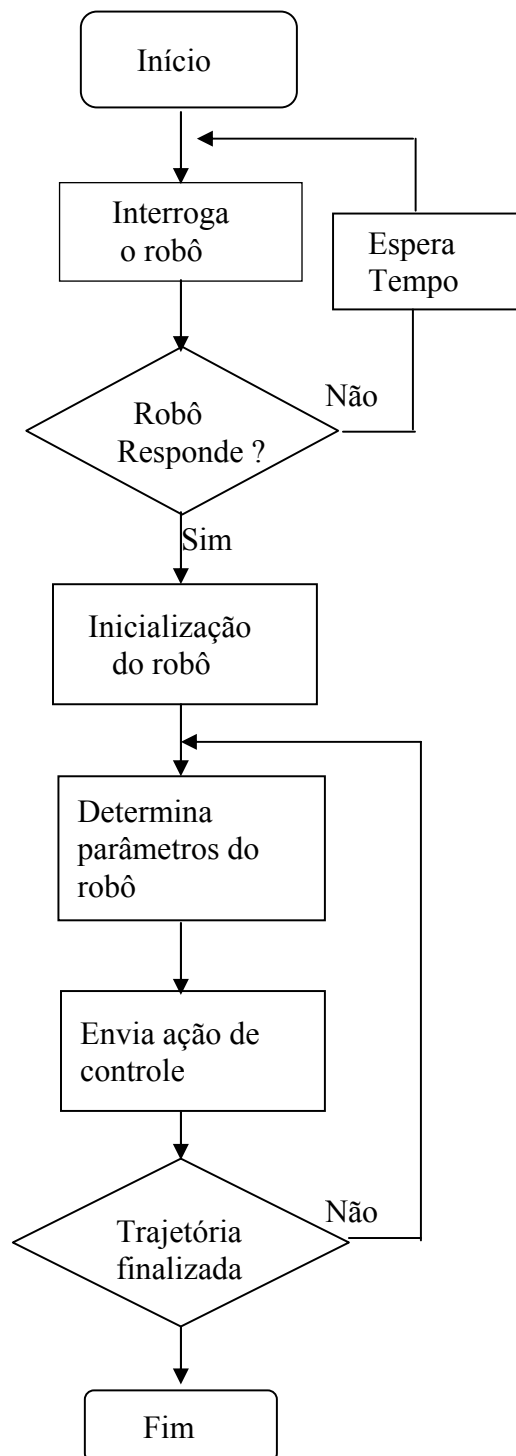


Figura 5.4 Fluxograma do controle de trajetória

Conclusão do capítulo

Este capítulo mostra alguns detalhes construtivos do robô móvel GPAR, dos sistemas componentes do robô bem como das mensagens trocadas entre o robô e o controlador, além do uso de um protocolo simples para a execução da trajetória.

O apêndice B mostra os esquemas das placas dos circuitos componentes do robô móvel.

CAPÍTULO 6 – Resultados e Discussões

Este capítulo mostra alguns dos resultados obtidos com o robô móvel. O software dos controladores foi implementado em MATLAB ® e simulink ® e os métodos de identificação utilizados foram: o MQR (Mínimos Quadrados Recursivos) com busca aleatória (*random walk*) e através de redes neurais. O software embarcado foi desenvolvido em C.

6.1 Dados coletados de velocidade

A figura 6.1 mostra a coleta de dados reais de velocidades dos motores utilizada para testar a comunicação e a coleta dos *encoders*. As medidas estão representadas em cm/s em função do tempo em segundos. Estão representados os valores obtidos dos motores direito e esquerdo do robô.

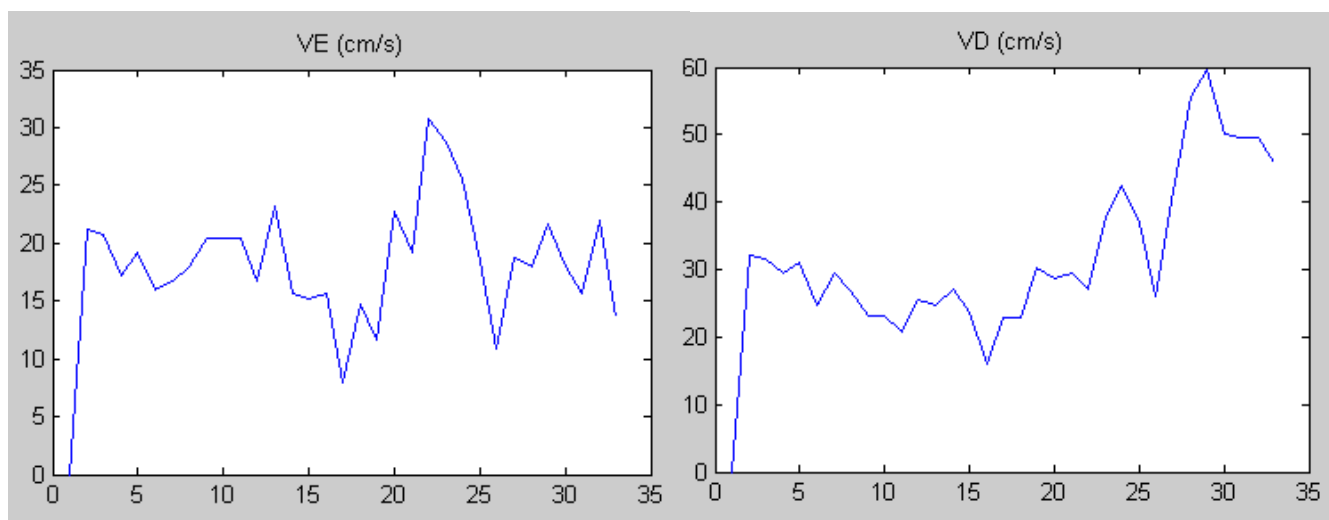


Figura 6.1 Coleta de dados de velocidade

A figura 6.2 mostra o ruído inserido pelos sensores no motor direito, para um tempo de coleta de 200 segundos, bem como a média das 3 últimas amostras e a mediana. O ruído de coleta está sempre presente, entretanto não prejudicou o sistema devido à alta relação sinal-ruído.

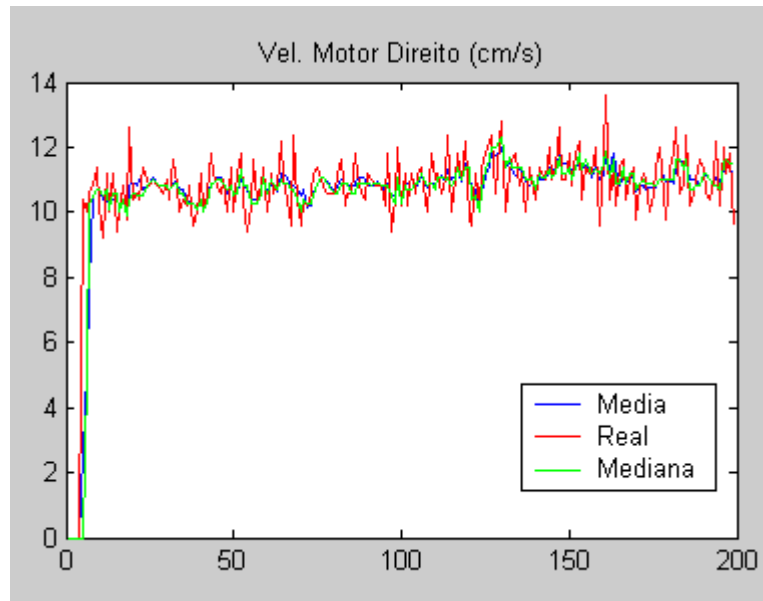


Figura 6.2 Ruído presente nas medidas

6.2 Resultados da identificação dos motores

Foram utilizados para a identificação dos parâmetros dos motores sinais do tipo *Pseudo Random Binary Signal* (PRBS). O modelo utilizado foi de segunda ordem conforme obtido no capítulo 3.

6.2.1 Mínimos Quadrados Recursivo

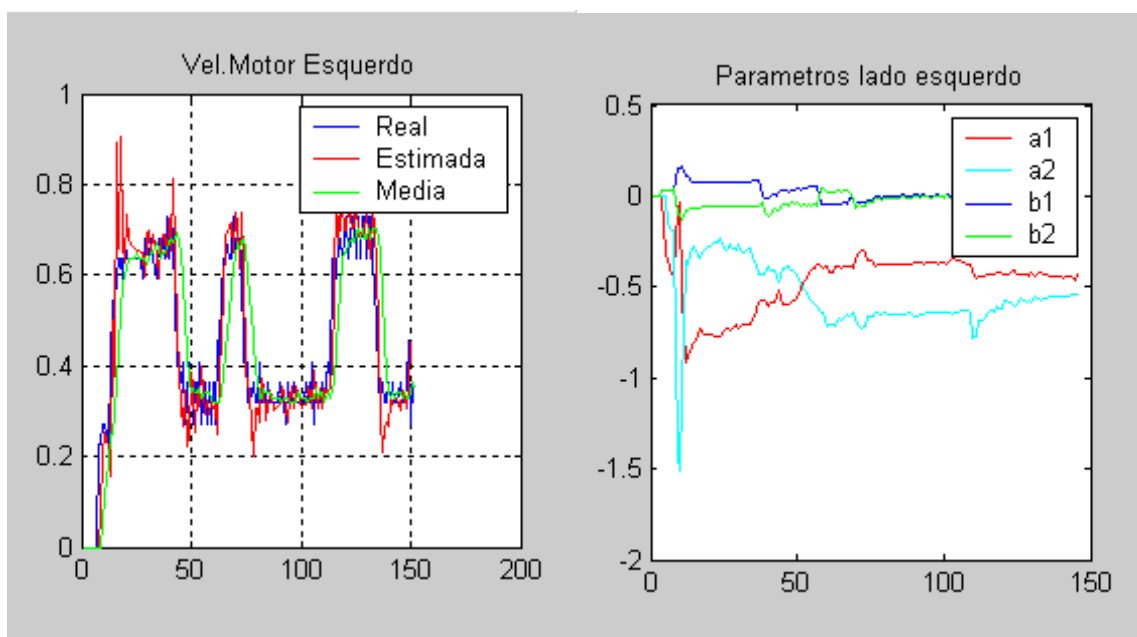


Figura 6.3 Identificação de parâmetros do motor esquerdo (MQR)

6.2.2 Redes Neurais

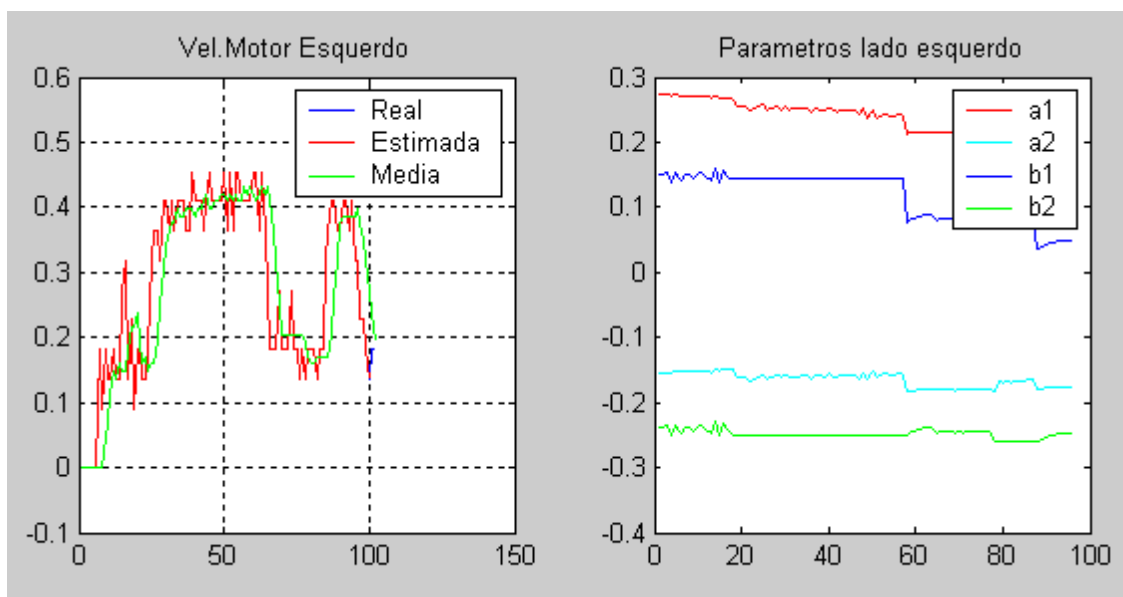


Figura 6.4 Identificação de parâmetros do motor esquerdo (RNA)

As figuras 6.3 e 6.4 mostram os valores obtidos para os parâmetros e os valores estimados por cada um dos métodos. Os resultados estão em rotações por segundo.

A rede utilizada na obtenção dos resultados tinha uma camada oculta com 80 neurônios. O modelo utilizado foi o mesmo na identificação pelo método dos mínimos quadrados recursivos.

O método utilizado por redes neurais exige uma carga de processamento maior devido ao treinamento da rede. No caso do robô móvel, onde pode haver variação de parâmetros, o método de identificação por redes neurais, não pode ser empregado com o intuito de obter bons resultados, pois não permite a correção paramétrica dentro do período de amostragem utilizado de 0.25 segundos, por isso na implementação em tempo real, optou-se pelo MQR.

6.3 Controle dos motores com o PID

As figuras nesta secção mostram os resultados experimentais obtidos com o PID controlando a velocidade dos motores para os dois métodos de identificação: MQR e RNA. O método de identificação por RNA, teve o número de neurônios ajustado de forma empírica, tentando minimizar este valor para que o treinamento da rede consumisse menos recursos computacionais.

6.3.1 Identificação por RNA

A figura 6.5 mostra o desempenho do controlador PID com uma rede neural cuja camada oculta possuía 80 neurônios.

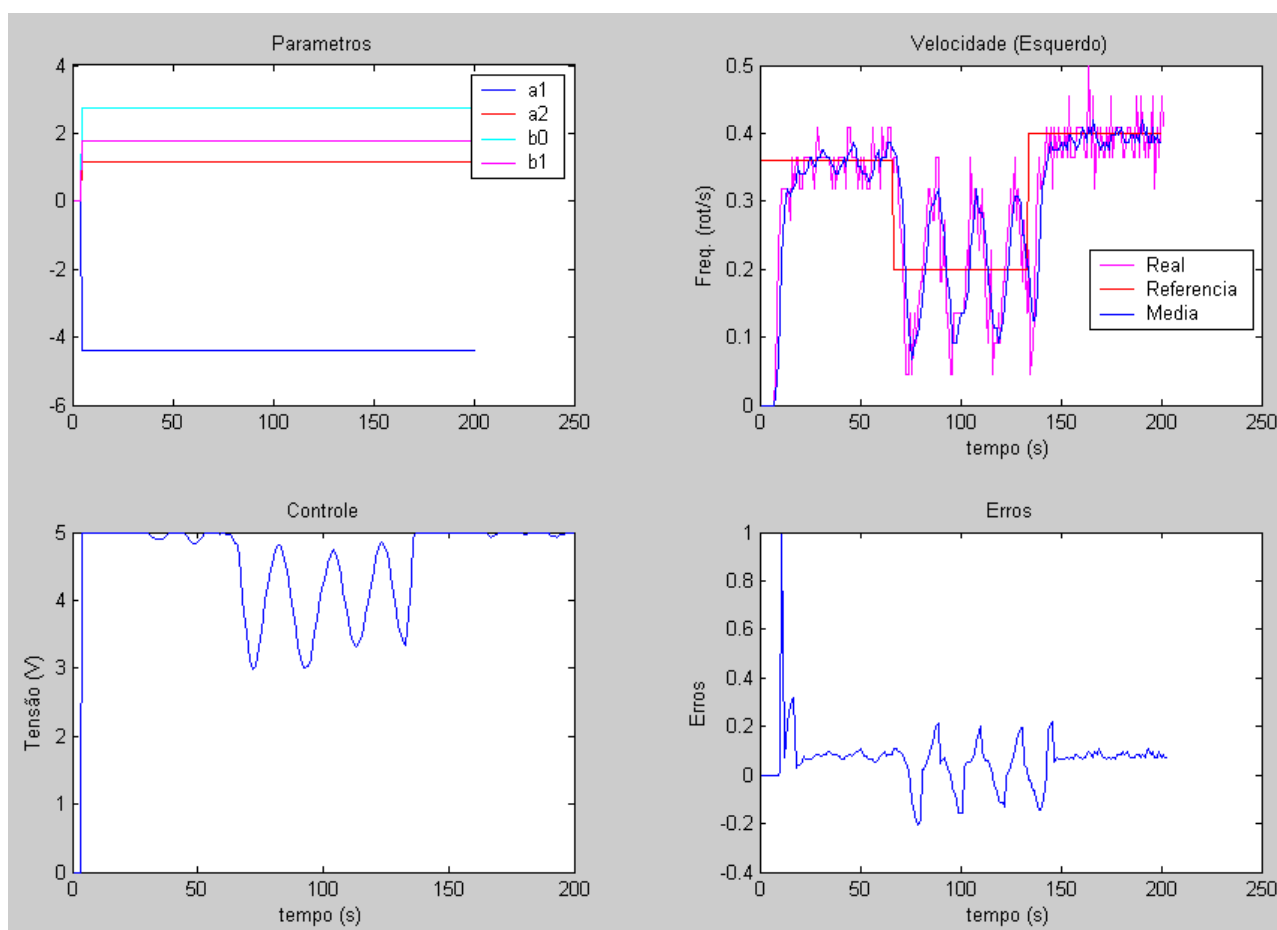


Figura 6.5 Desempenho do PID (Identificação RNA – 80 neurônios ocultos)

Embora os parâmetros se mantenham constantes, o controlador apresentou um erro alto na mudança de referência. A figura 6.6 mostra o desempenho do controlador quando o número de neurônios na camada oculta foi aumentado para 120.

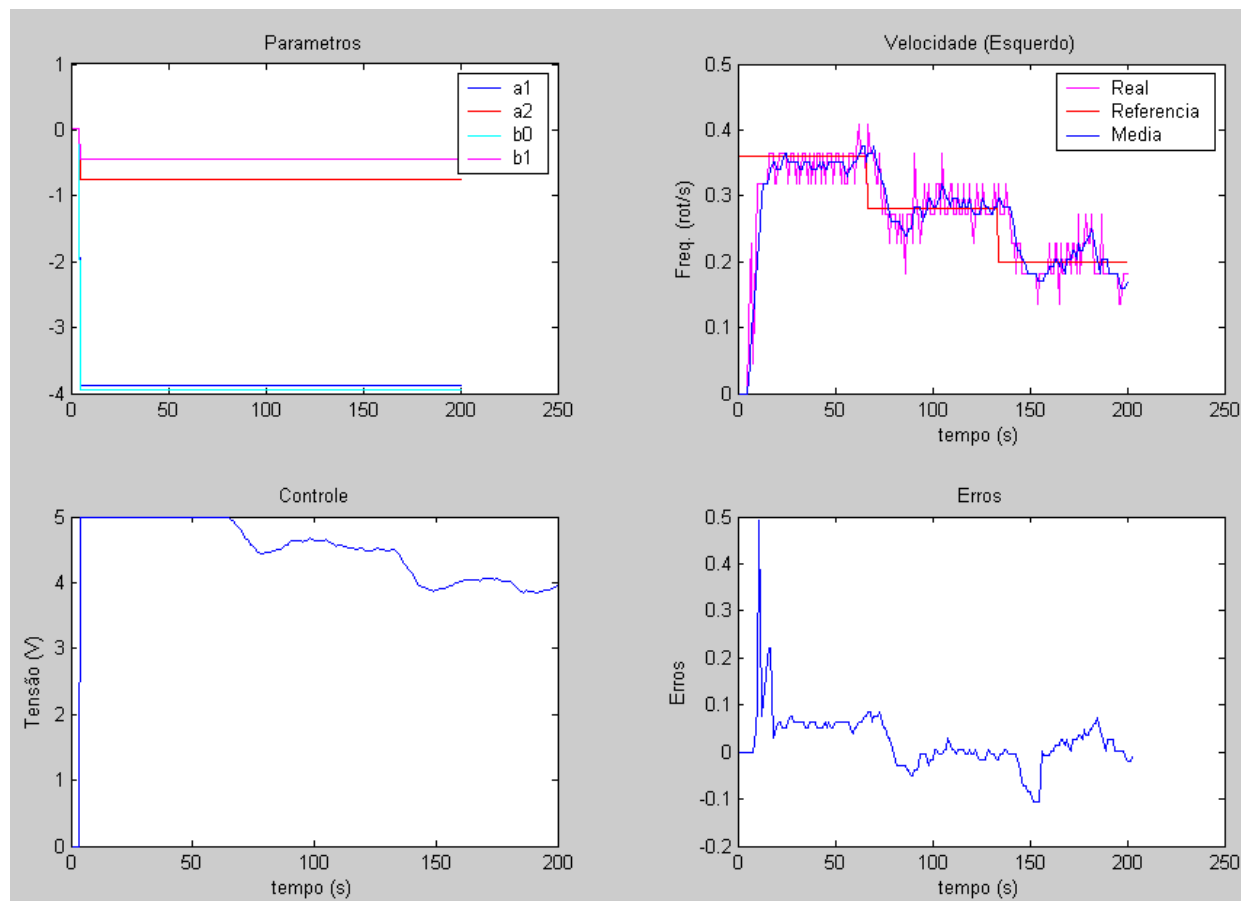


Figura 6.6 Desempenho do PID (Identificação RNA – 120 neurônios ocultos)

O aumento no número de neurônio ocultos melhorou os resultados do processo de identificação, tornando o controlador mais satisfatório.

Os valores foram medidos em rotações por segundo. Apesar de apresentarem uma quantidade considerável de ruído, o controlador atuou de maneira correta.

A correção paramétrica foi realizada a cada três períodos de amostragem com a finalidade de diminuir a carga no processamento.

Esses resultados são referentes ao controle de velocidade da malha interna. Essa malha recebe como referência a velocidade calculada pelos controladores da malha externa.

Os parâmetros do controlador PID da malha interna foram calculados com base nos parâmetros dos motores encontrados pela rede neural.

6.3.2 Identificação MQR

A figura 6.7 mostra os resultados obtidos com o controlador PID com identificação MQR para referências diferentes.

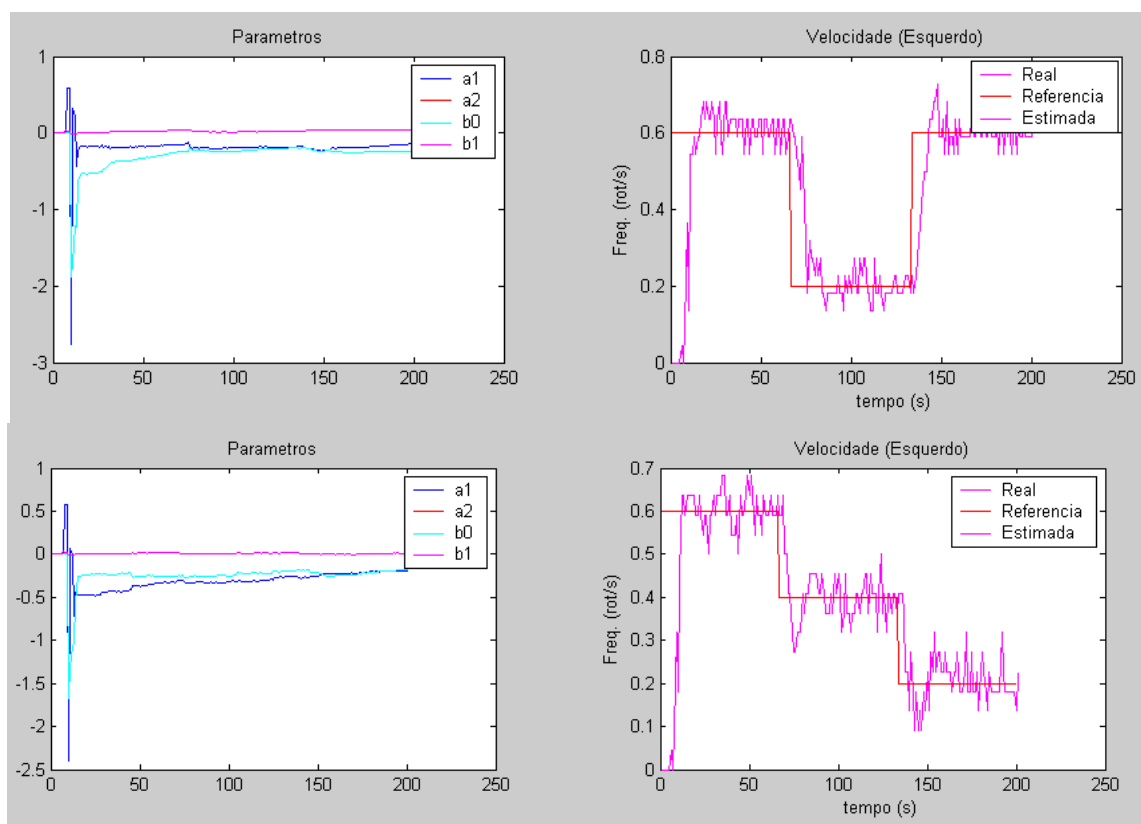


Figura 6.7 Desempenho do PID (Identificação MQR)

Foram feitas modificações dos valores de referência de modo a se verificar o desempenho do controlador.

Esses resultados foram obtidos pelo controlador de velocidade dos motores da malha interna.

Esses controladores normalmente são implementados embutidos, entretanto no caso do robô, o controle foi centralizado de modo a liberar o processamento dos microcontroladores do robô.

A figura 6.8 mostra os resultados obtidos também com o controlador PID. Nestes casos foi utilizado o valor das últimas três amostras. Este procedimento foi adotado de modo a se reduzir as variações na coleta de velocidade dos sensores, reduzindo a carga de processamento devido à menor variação de parâmetros.

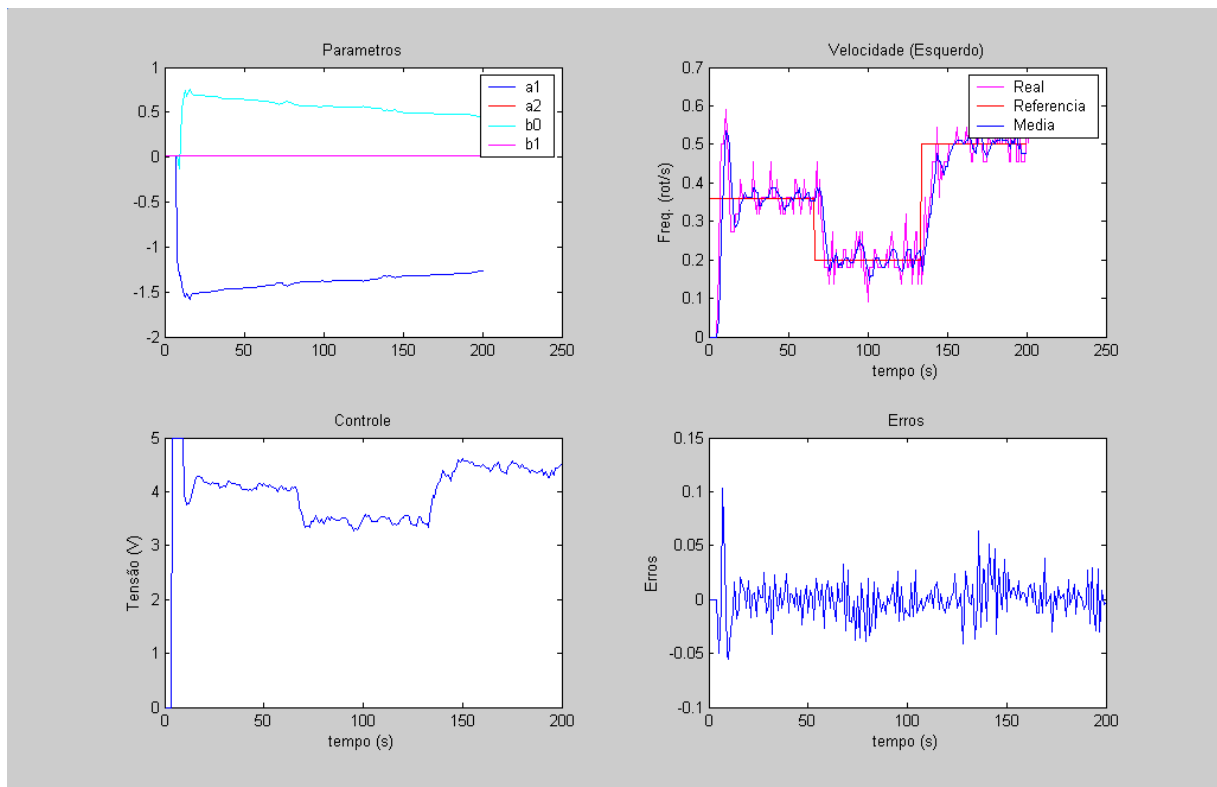


Figura 6.8 Desempenho do controlador PID (MQR – média de 3 amostras)

A figura 6.9 mostra o resultado obtido pelo mesmo procedimento mas com a mudança de referência. As medidas obtidas estão mostradas em função do tempo (em segundos).

O desempenho do controlador se mostrou satisfatório na maioria dos casos, assegurando seu uso como controlador da malha interna de velocidade.

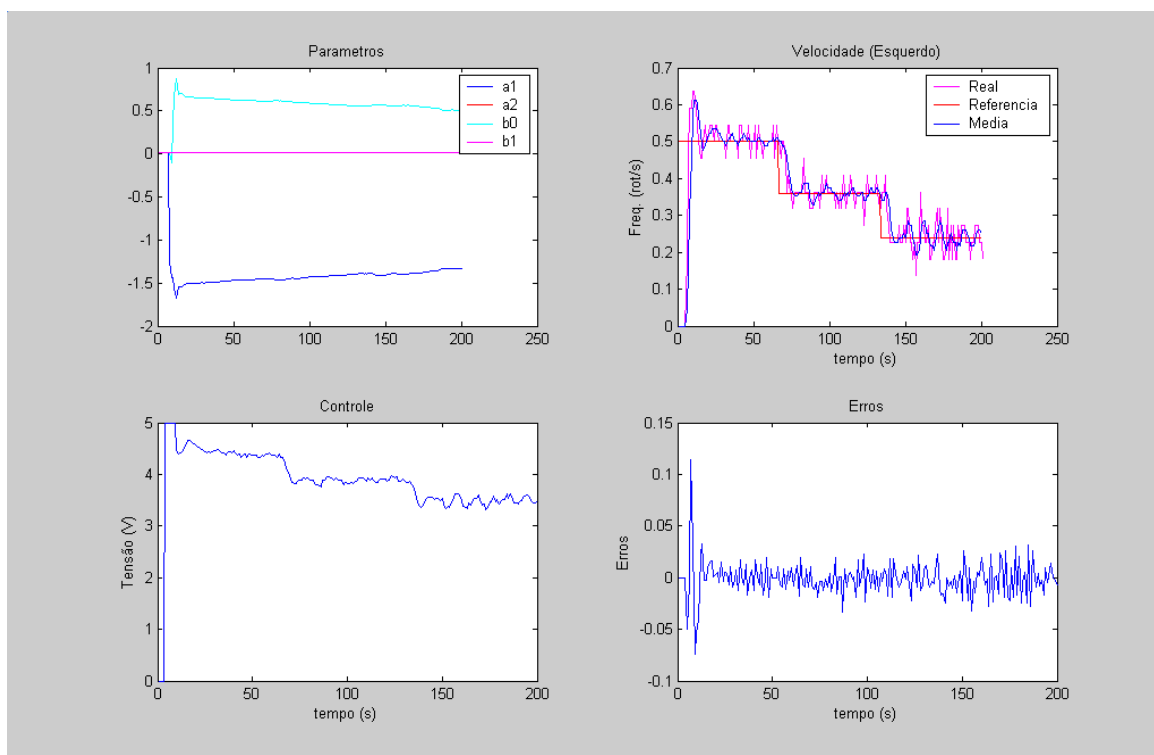


Figura 6.9 Desempenho do controlador PID (MQR média), referência diferente

6.4 Controle dos motores com o GPC

Os gráficos mostrados nesta secção apresentam os resultados experimentais obtidos com o GPC controlando a velocidade dos motores para os dois métodos de identificação: MQR e RNA.

Esses resultados foram obtidos através do controle de velocidade da malha interna. Os resultados permitem comparar a atuação do controlador preditivo com o PID.

Foi utilizado um horizonte de previsão e de controle fixo no valor de 5. Os resultados também foram satisfatórios, pois o controlador atuou buscando a referência de maneira correta.

O período de amostragem utilizado foi de 0,25 segundos, o mesmo valor utilizado anteriormente.

6.4.1 Identificação MQR

A figura 6.10 mostra o desempenho do controlador GPC, onde os parâmetros do motor foram calculados pelo método de identificação MQR.

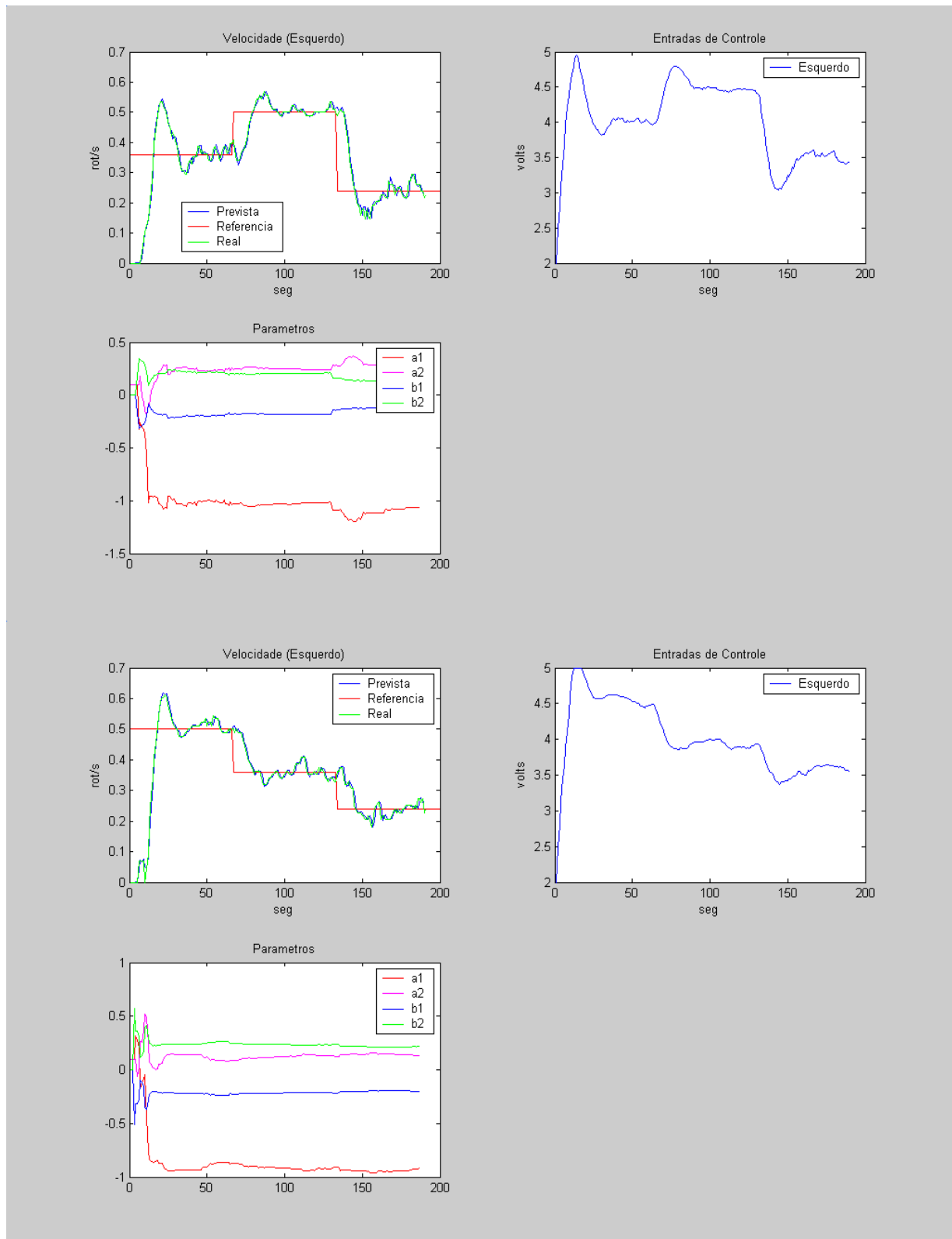


Figura 6.10 Resultados com o controlador GPC (identificação MQR)

A figura mostra resultados obtidos com valores de referência diferentes. O desempenho do controlador mostrou que o mesmo pode desempenhar, assim como o PID, a função de controlador de velocidade da malha interna, entretanto o PID foi escolhido pela simplicidade de implementação, podendo mesmo ser desenvolvido para o microcontrolador embarcado, liberando o processamento do controlador principal.

6.5 O Simulador MATLAB do robô móvel

Foi desenvolvido um simulador para o robô para uso no MATLAB, de modo que se possa fazer várias modificações e verificar o comportamento do robô diante de situações diversas, sendo portanto uma ferramenta de suporte no estudo de robótica.

A figura 6.11 mostra a interface do simulador do robô móvel. O simulador está em desenvolvimento com a finalidade de adicionar novas facilidades.

Pode-se mudar o horizonte, bem como o atraso de modo que se possa verificar o erro de modelagem. Os parâmetros dos motores direito e esquerdo também podem ser modificados de modo a se verificar a consequência da diferença de parâmetros entre os dois lados do robô móvel.

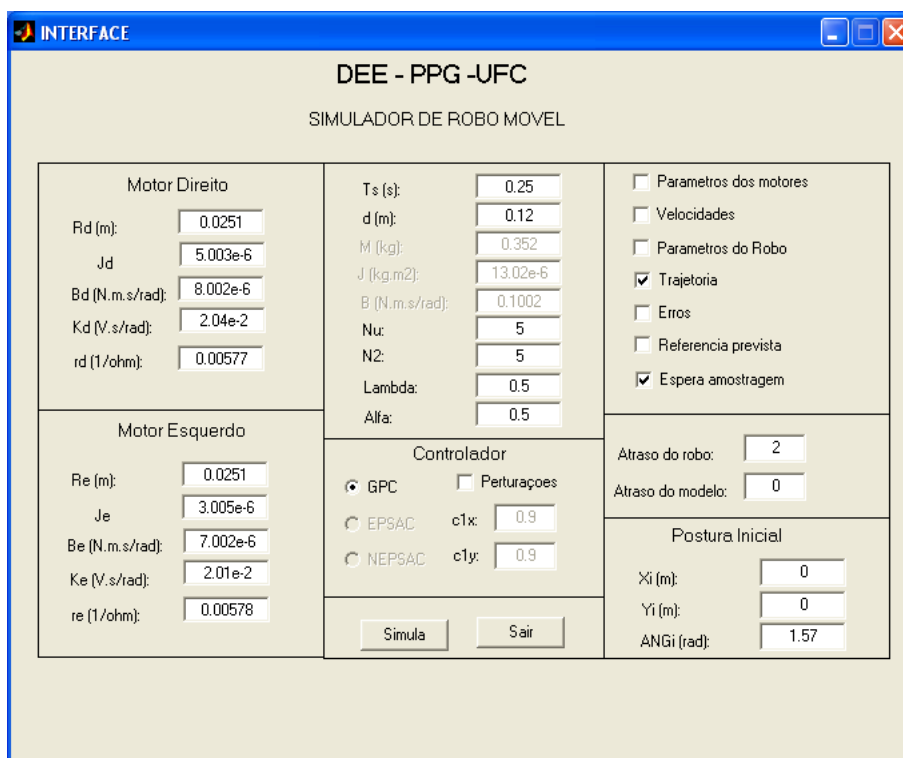


Figura 6.11 Interface do simulador do robô móvel

A figura 6.12 mostra os parâmetros obtidos pelo simulador para o modelo dos motores CC do robô móvel. O simulador permite a escolha da verificação gráfica de variáveis do sistema, inclusive simulando o percurso do robô durante a execução de sua trajetória.

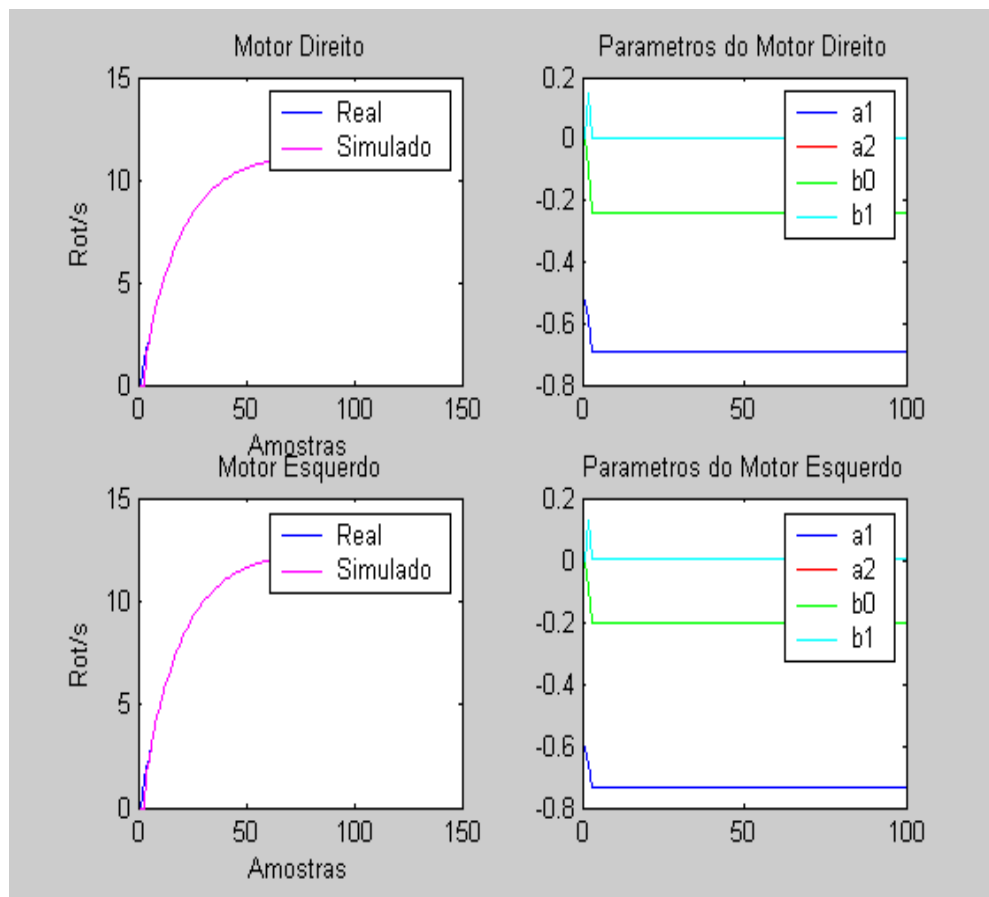


Figura 6.12 Parâmetros dos motores (simulação)

A figura 6.13 mostra o gráfico das velocidades linear e angular que o robô deveria ter para seguir a trajetória.

Pode-se visualizar ainda a trajetória de referência a ser seguida pelo robô.

A figura 6.14 mostra o desempenho do simulador do controlador GPC. Pode-se verificar que o mesmo tem um bom desempenho.

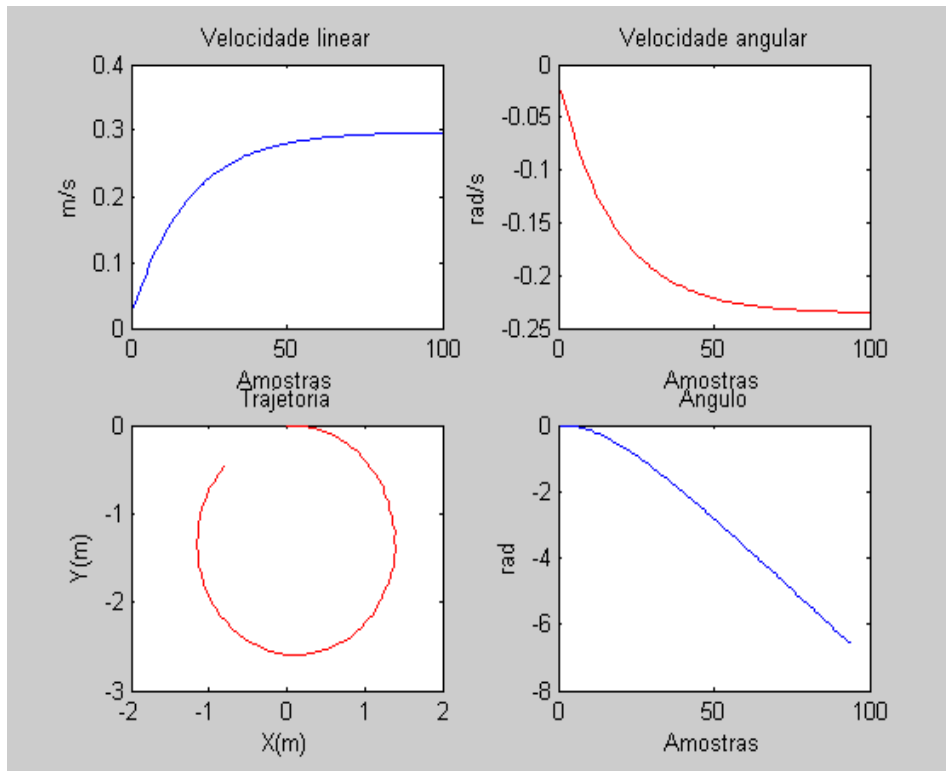


Figura 6.13 Referência geradas pelo simulador

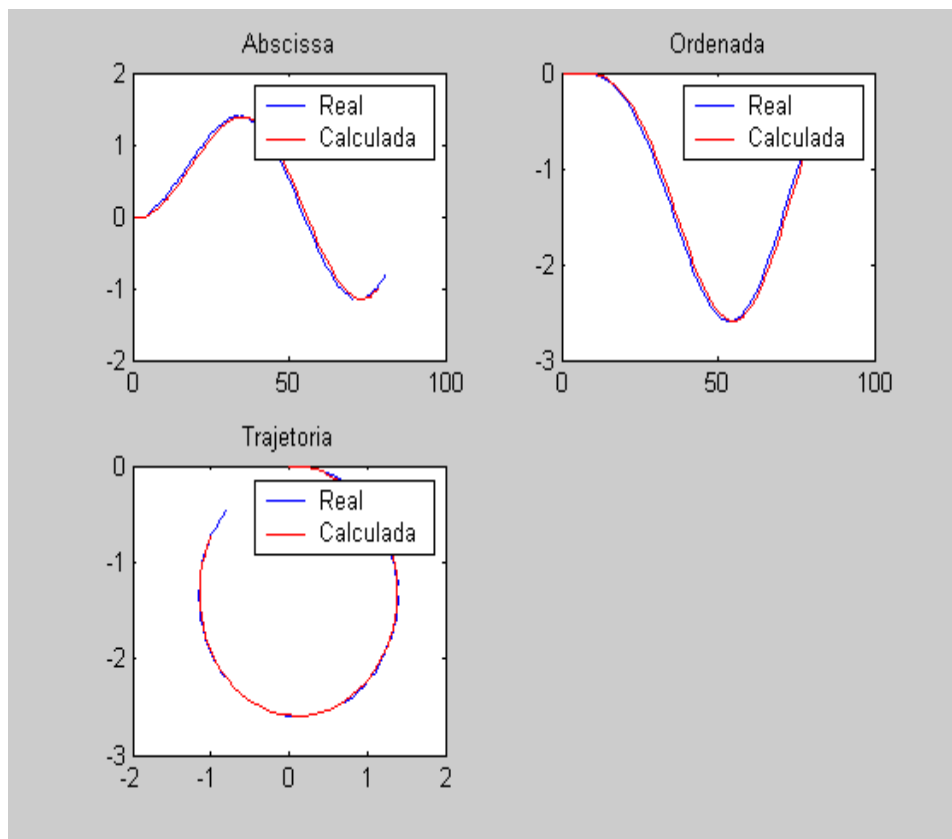


Figura 6.14 Trajetória do simulador do robô móvel

Para a geração da trajetória da figura 6.14, foi considerado que o robô possui um atraso de 2 (dois) períodos de amostragem. O valor do horizonte de controle utilizado pelo GPC foi $N_u=2$. A figura 6.15 mostra a mesma trajetória calculada pelo controlador preditivo para um valor de horizonte de controle $N_u=5$.

Pode-se verificar que com horizontes de controle menores a trajetória calculada pelo controlador preditivo fica mais próxima da referência. Os outros gráficos mostram as coordenadas em função do número de amostras obtidas na geração da trajetória.

A figura 6.16 mostra o erro calculado para as coordenadas do robô móvel. Devido ao atraso de transporte e às diferenças entre os parâmetros do lado direito e esquerdo, o robô executa movimentos irregulares, o que pode ser visto no erro gerado.

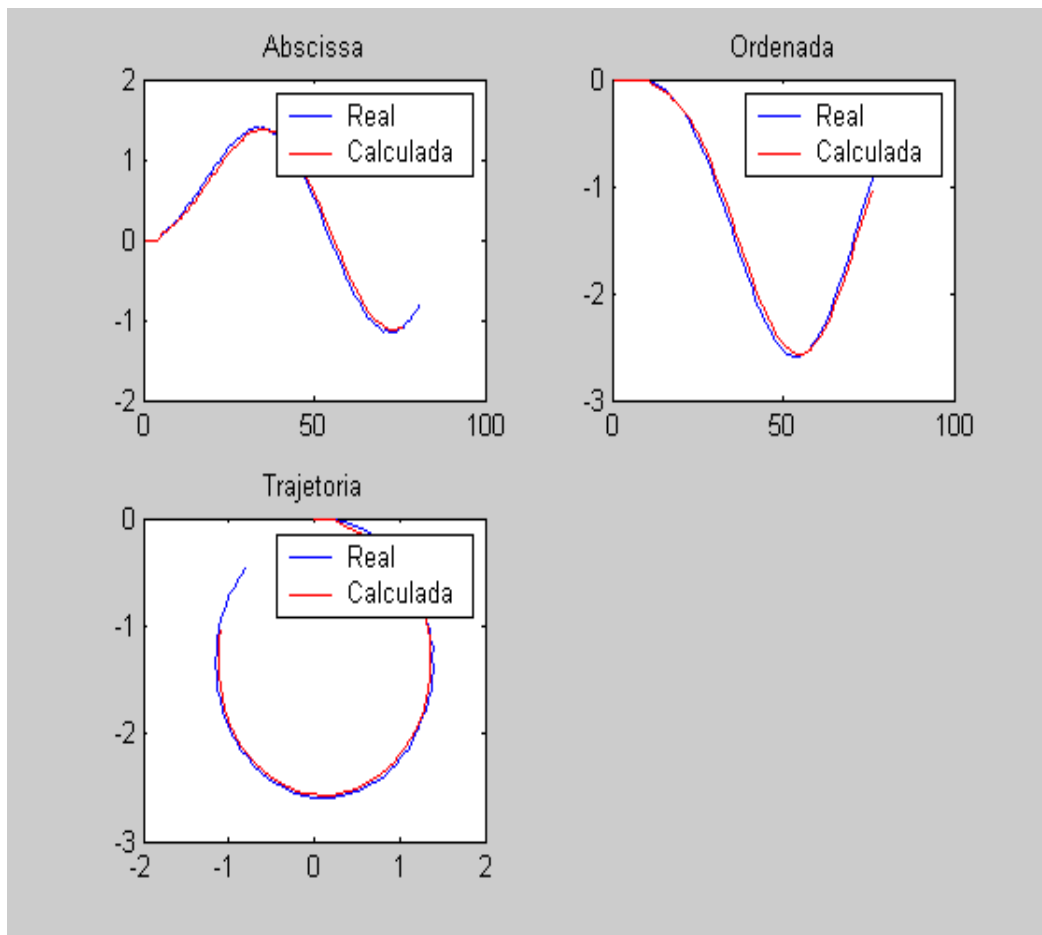


Figura 6.15 Trajetória calculada pelo simulador

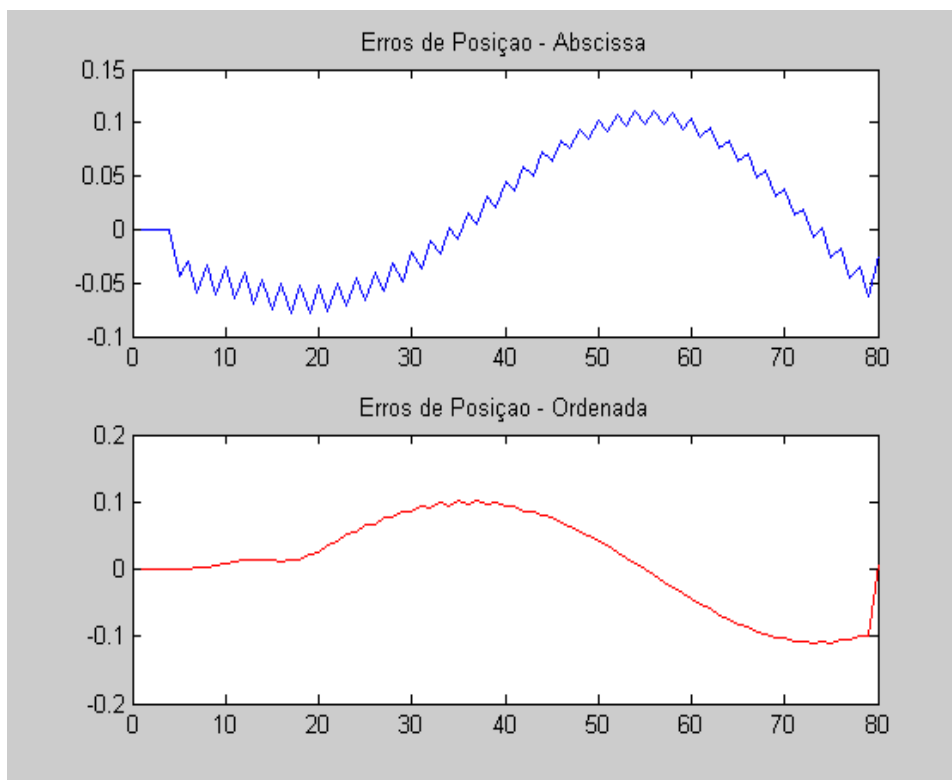


Figura 6.16 Erros de posição

6.6 O Simulador Simulink

Foi desenvolvido um simulador para o robô móvel em simulink®, de modo a se verificar o efeito na mudança de parâmetros. Este simulador permite a inclusão de perturbações ao longo da trajetória de modo a se verificar o desempenho do controlador GPC no cálculo da trajetória. O valor da perturbação, bem como o instante de tempo onde a perturbação será aplicada podem ser escolhidos pelo usuário.

Estão sendo implementadas novas funcionalidades no simulador, como variação de parâmetros do ambiente, que acarretam mudanças nos parâmetros do robô, variação no tipo de perturbação, de modo que possam ser simuladas situações diferentes que o robô possa encontrar ao longo da trajetória, adaptatividade dos controladores e simulações de defeitos sistemáticos, como diferença no raio das rodas do robô e ruído do sensor, tornando a análise de resultados mais próxima do ambiente que será encontrado em uma situação prática. A figura 6.17 mostra o diagrama de blocos do simulador sem o controlador cinemático.

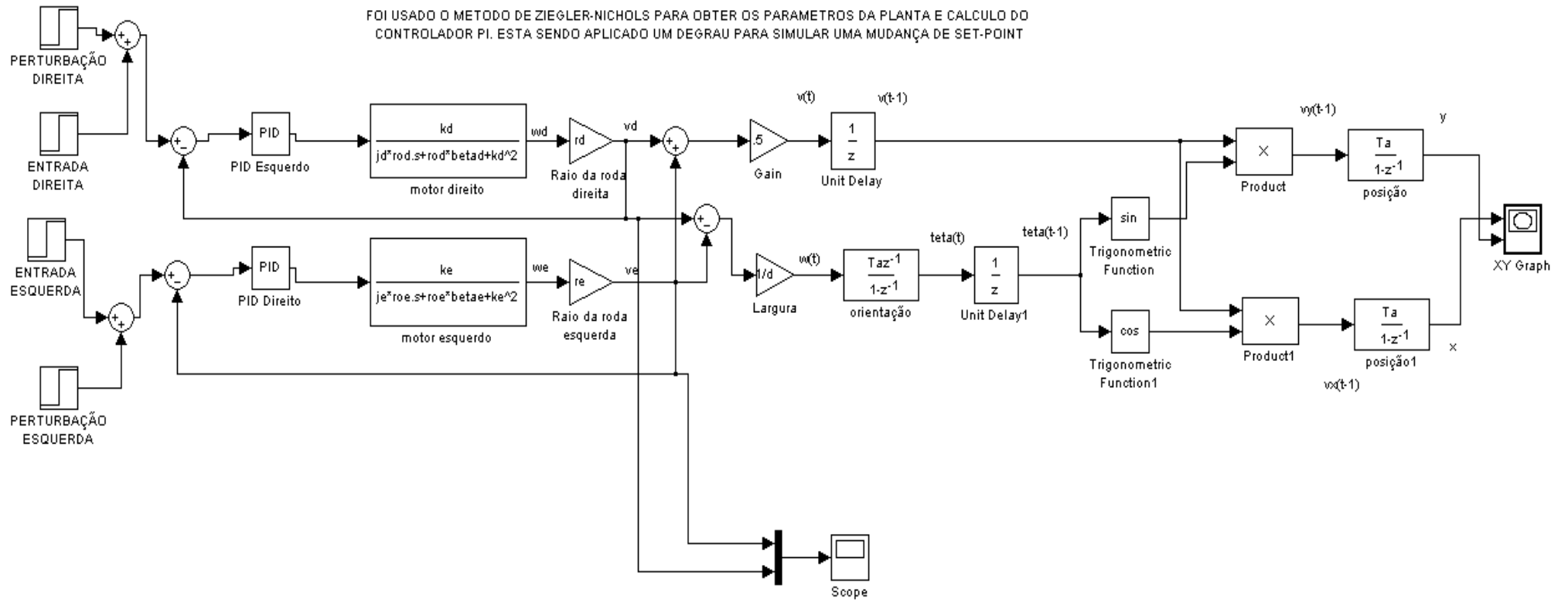


Figura 6.17 Diagrama de blocos do simulador

A figura 6.18 mostra um dos resultados obtidos com o simulador com a inclusão de uma perturbação do tipo degrau de amplitude diferente nas rodas direita e esquerda, de modo que foi obtida uma mudança na curvatura do robô móvel, fazendo com que o mesmo executasse uma trajetória sinuosa.

A figura 6.19 mostra as velocidades dos motores obtidas com o uso de controladores PI (Proporcional-Integral). Pode-se notar que o tempo de estabilização está em torno de 1 segundo. Foram feitas modificações nos parâmetros do controlador PI e obteve-se tempos de estabilização mais curtos e com um sobre-sinal menor, possibilitando seu uso em situações práticas de modo que a trajetória seja executada de maneira mais eficiente.

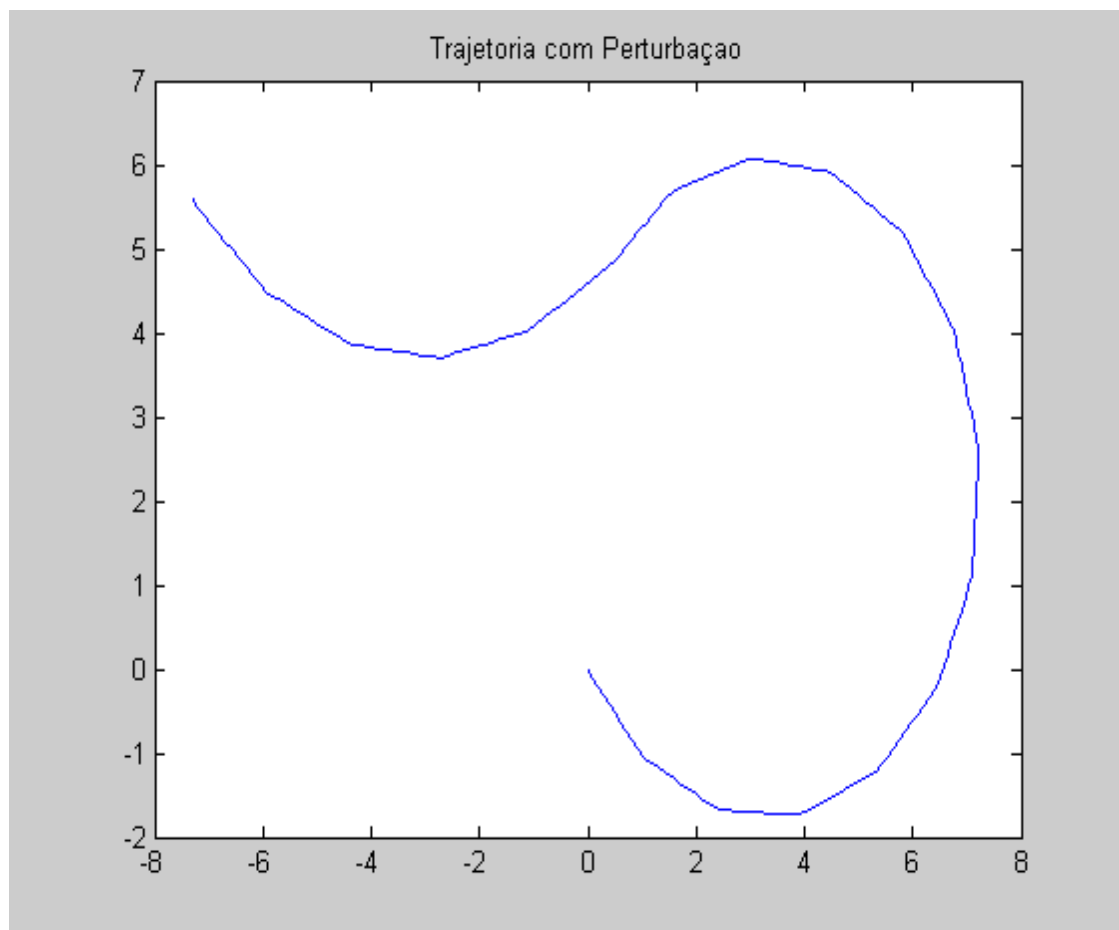


Figura 6.18 Trajetória obtida com a atuação de perturbações de amplitudes diferentes

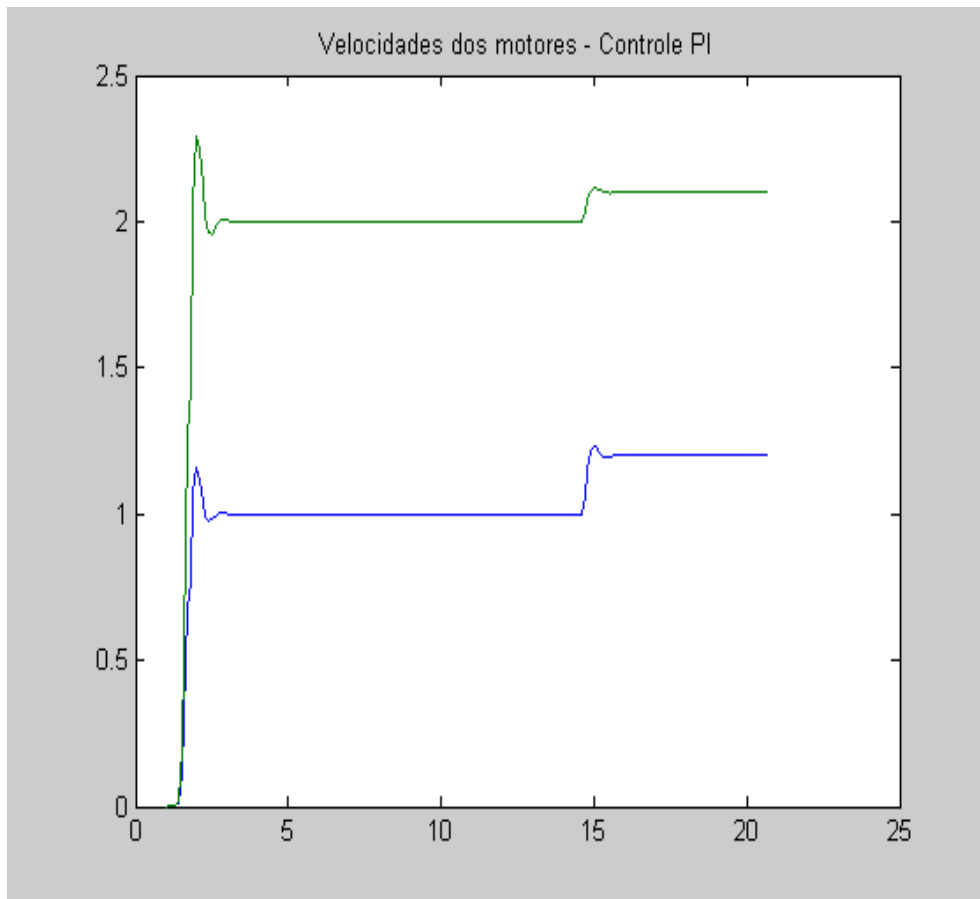


Figura 6.19 Velocidades dos motores com os controladores PI

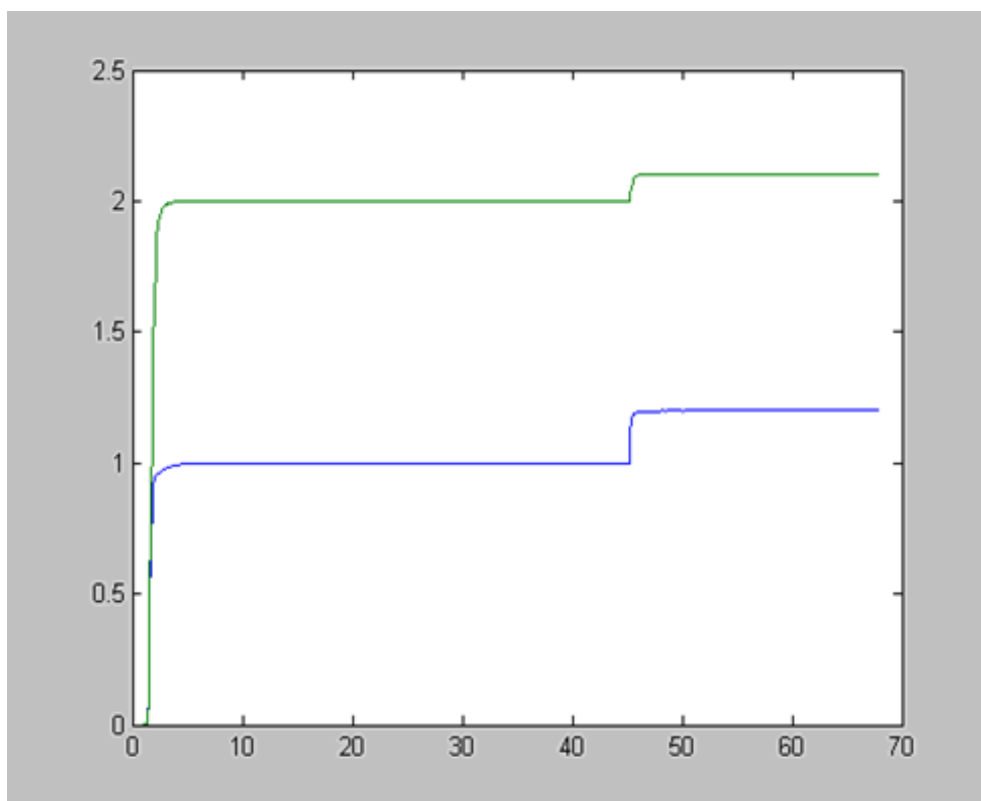


Figura 6.20 Velocidades dos motores com os controladores PI (sem sobressinal)

A figura 6.20 mostra as velocidades obtidas após a sintonia dos parâmetros do controlador PI. O tempo de estabilização foi reduzido para 0,5 segundo e o sobre-sinal foi eliminado, permitindo assim um controle desejável da trajetória. A desvantagem desse método é que os parâmetros devem ser corrigidos e o desempenho testado com os novos valores. O simulador está sendo modificado para executar a sintonia automática do controlador.

As figuras 6.21 e 6.22 mostram os valores da identificação dos motores direito e esquerdo através do método de Ziegler-Nichols. O valor do período de amostragem utilizado foi 0.25 segundos. Para a identificação foi aplicado um degrau unitário aos motores.

O modelo de cada motor foi aproximado por uma função do tipo:

$$G_p(s) = \frac{Ke^{-ds}}{Ts + 1}$$

Os valores obtidos pelo método de Ziegler-Nichols foram ajustados de modo que a saída estimada de cada motor se aproximasse da saída real, permitindo um melhor ajuste nos valores calculados para os controladores PI de velocidade.

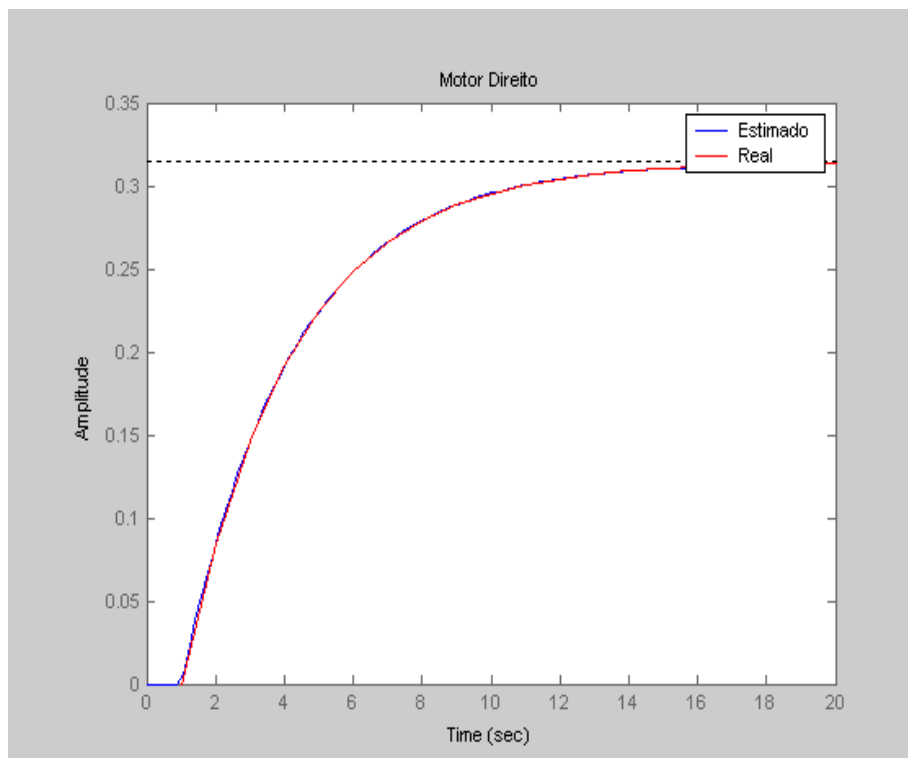


Figura 6.21 Identificação pelo método Ziegler-Nichols (motor direito)

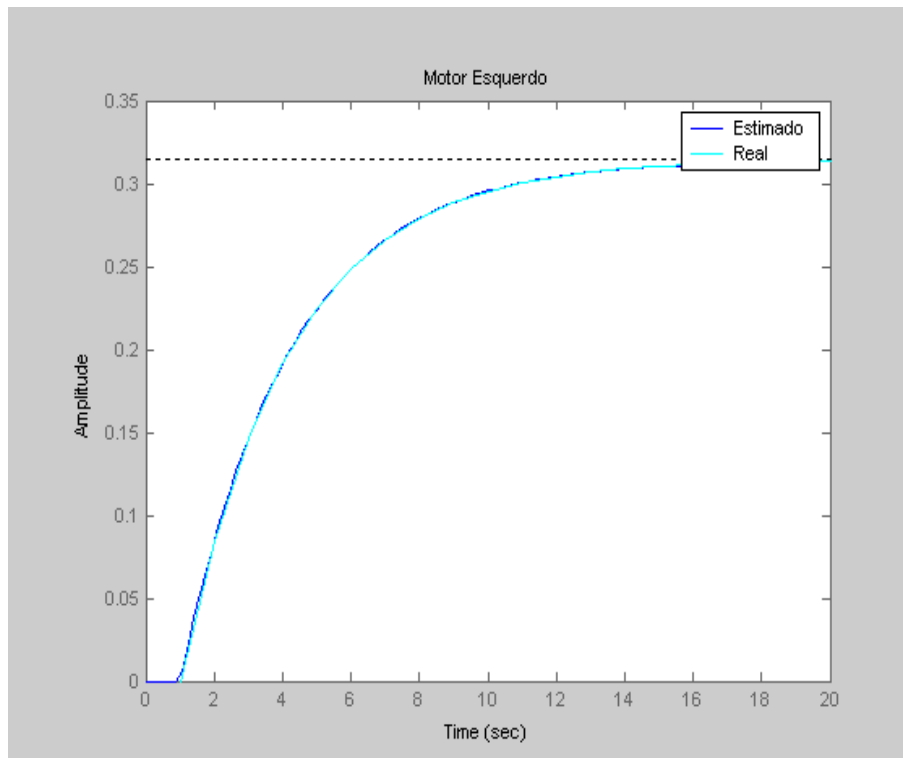


Figura 6.22 Identificação pelo método Ziegler-Nichols (motor esquerdo)

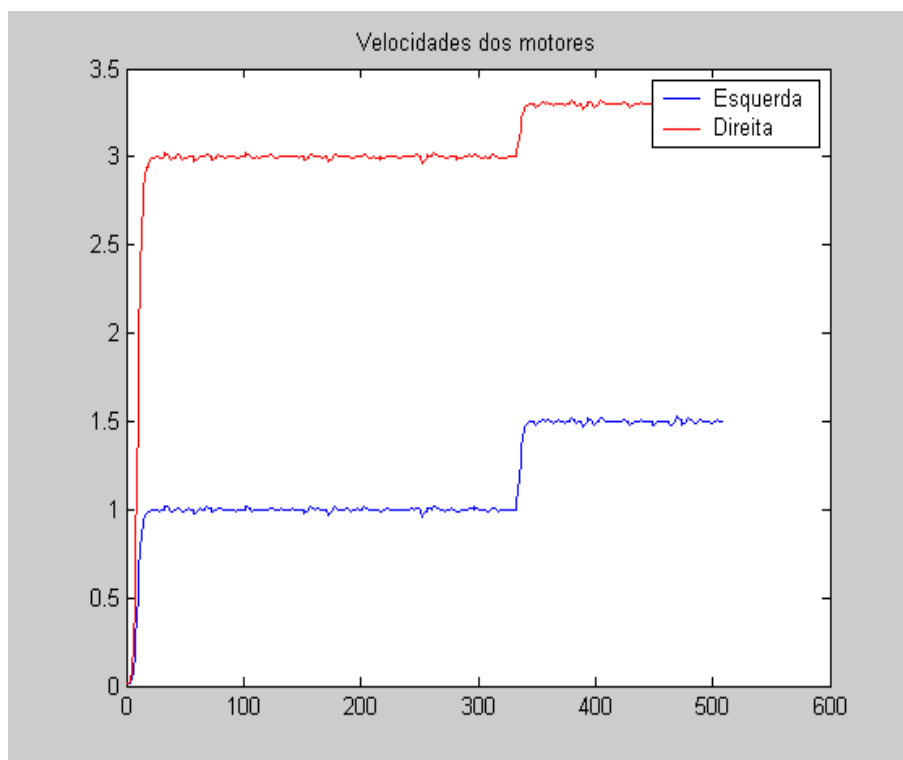


Figura 6.23 Velocidades controladas pelo PI com a inclusão dos ruídos

Foram incluídos no modelo do robô, ruídos na saída dos motores de modo a simular os valores obtidos nas medidas dos sensores.

A figura 6.23 mostra os valores obtidos com os controladores PI, após a inclusão dos ruídos dos sensores. A simulação com a inclusão do ruído nos sensores de velocidade, aproximam o modelo do robô real, podendo extrair informações mais precisas necessárias ao estudo mais completo do sistema, sem a necessidade que esse robô possa ser construído, para tal finalidade.

6.7 O Simulador dos Controladores - SIMULINK

Foi desenvolvido um simulador em simulink para o robô móvel com os controladores PID e GPC. As figuras 6.24 e 6.25 mostram os simuladores em simulink para ambos os controladores.

Foram desenvolvidas *S-functions* em matlab para as funções dos controladores. O cálculo dos controladores utilizou o modelo cinemático do robô móvel.

No controlador GPC pode-se escolher horizontes de controle de valor máximo $N_u=20$.

O simulador foi dividido em blocos menores de modo a facilitar a implementação, bem como possíveis mudanças no sistema.

O bloco malha interna é composto dos controladores PID de velocidade e dos motores CC. Estão também representados no diagrama dos simuladores blocos de transformação. A primeira das transformações faz a conversão das velocidades linear e angular (v e ω) em velocidades de referência para os motores direito e esquerdo.

O outro bloco de transformação faz a conversão inversa das velocidades dos motores direito e esquerdo, necessária para gerar v e ω , que gera os deslocamentos linear e angular.

São fornecidas ainda as referências de deslocamento angular e linear. O controlador GPC necessita do modelo de referência, assim devem ser fornecidos os parâmetros para o cálculo dos valores das variáveis de controle.

Os parâmetros são calculados pelo método MQR.

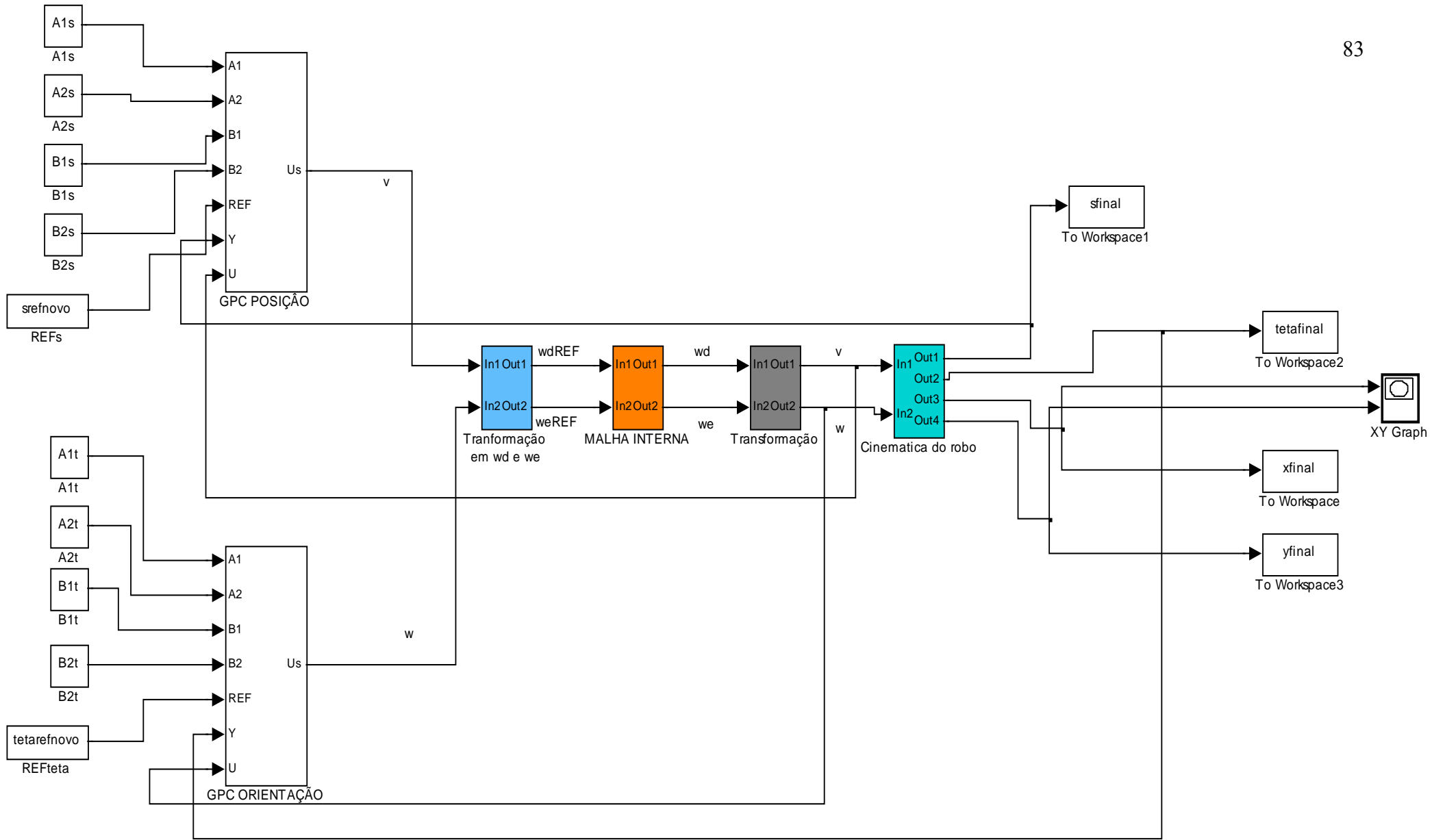


Figura 6.24 Controladores GPC de posição e orientação

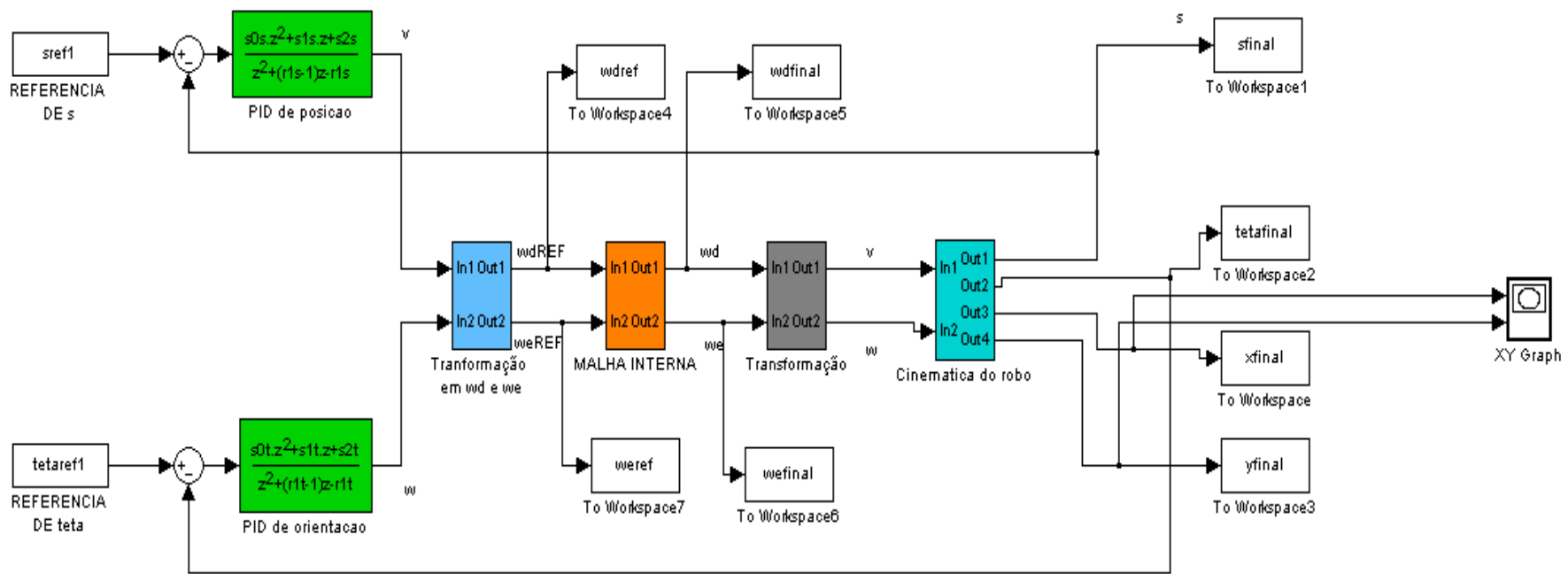


Figura 6.25 Controladores PID de posição e orientação

As figuras 6.25a, 6.25b, 6.25c e 6.25d mostram a composição dos blocos componentes do controle de posição e orientação dos robôs móveis.

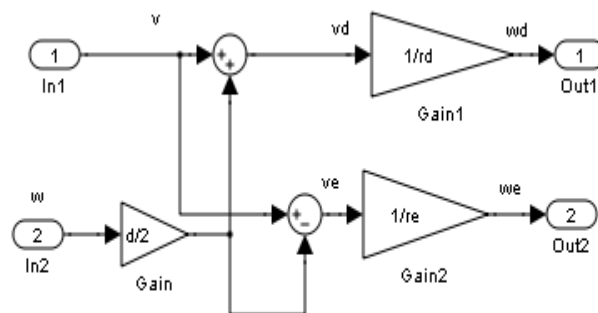


Figura 6.25a Transformação em ω_E e ω_D

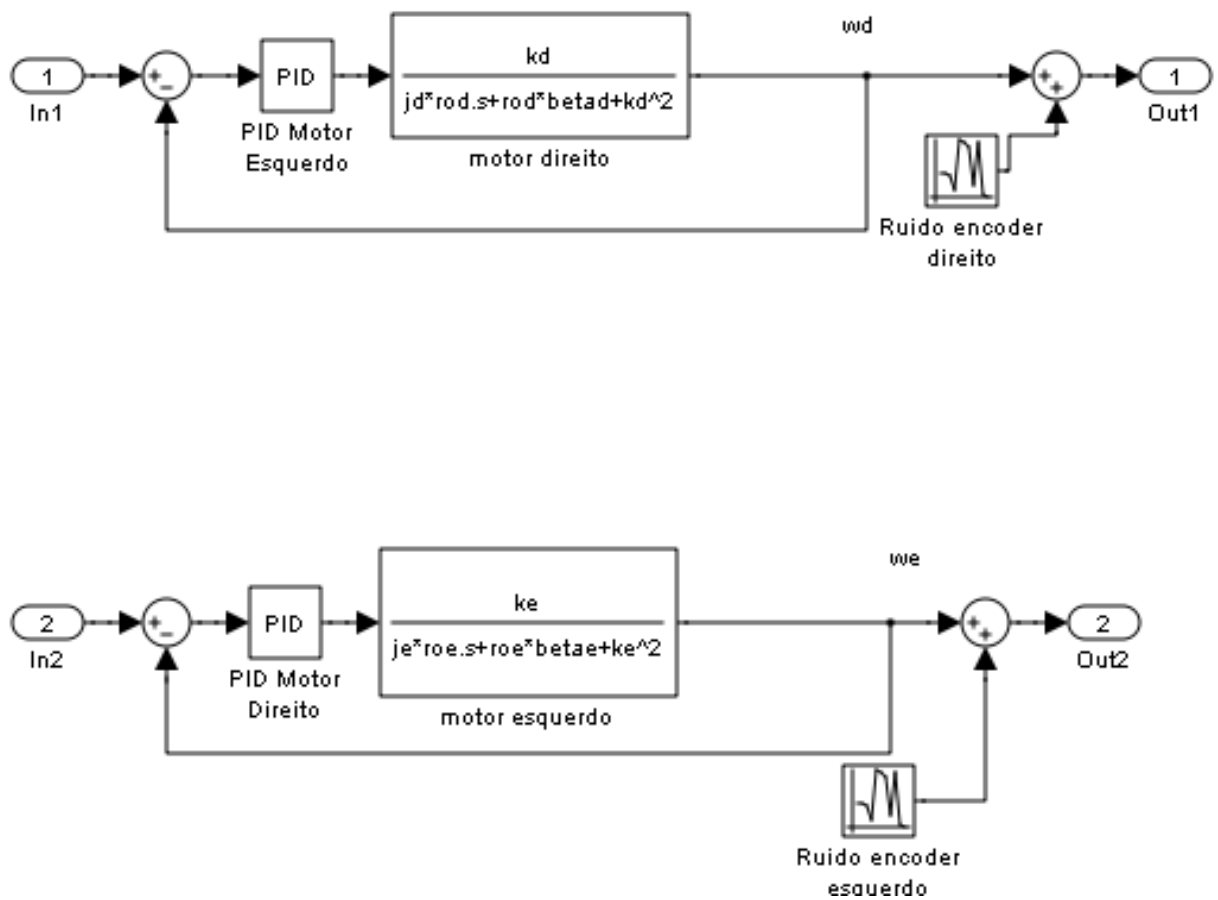


Figura 6.25b Malha Interna

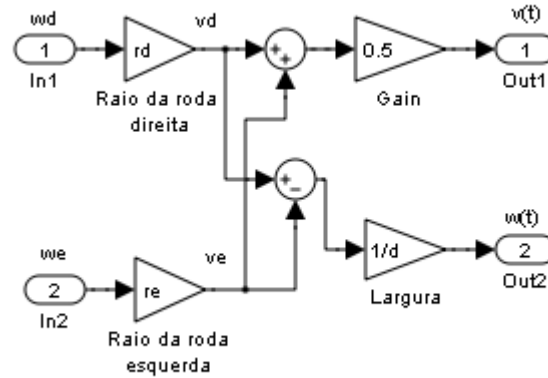


Figura 6.25c Transformação em v e ω

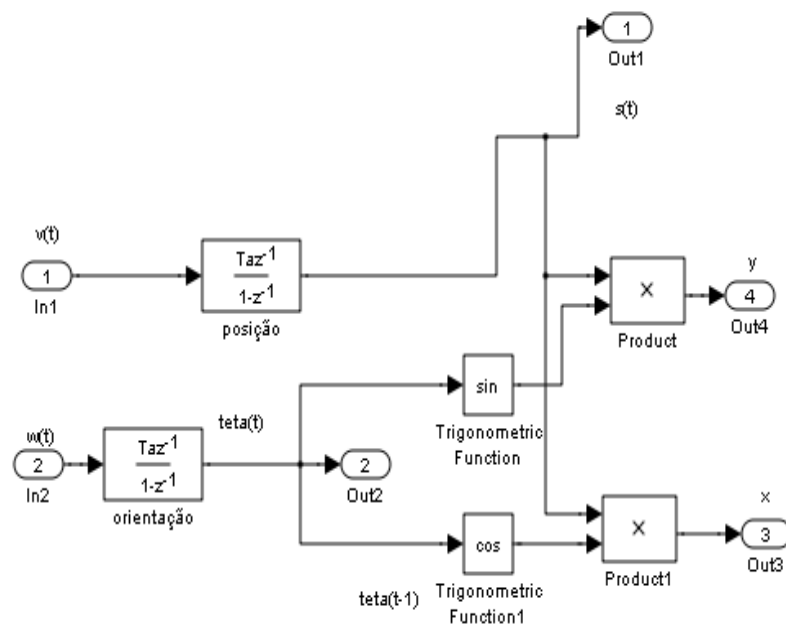


Figura 6.25d Cinemática do robô

As figuras 6.26 a 6.27 mostram os resultados obtidos com os simuladores dos controladores PID para a execução da trajetória mostrada na figura 6.28.

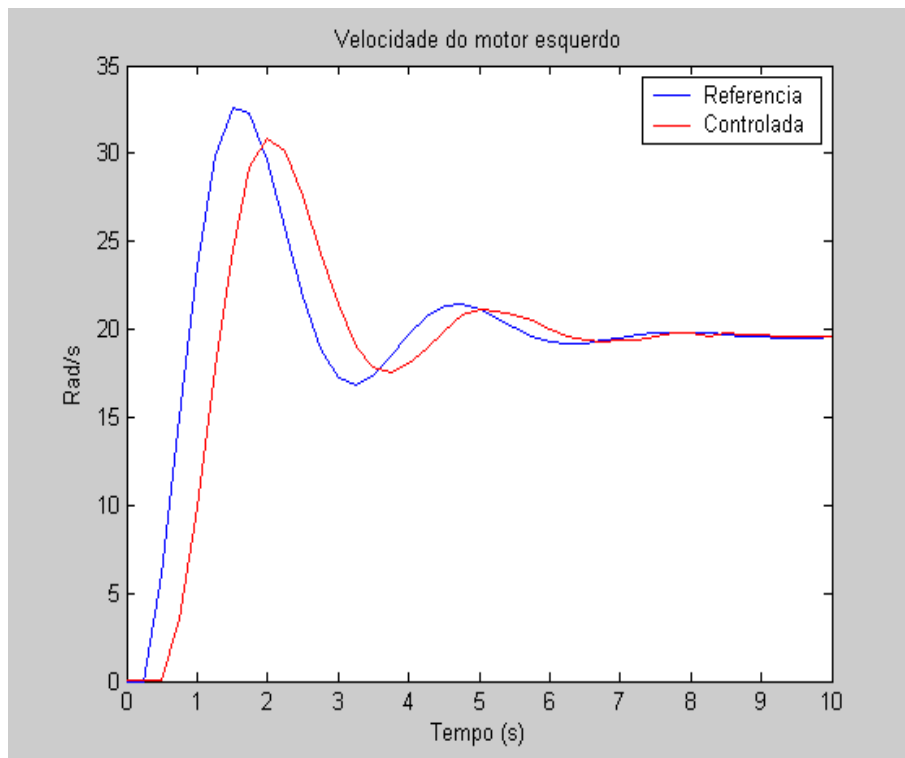


Figura 6.26 Controlador PID da malha interna (motor esquerdo)

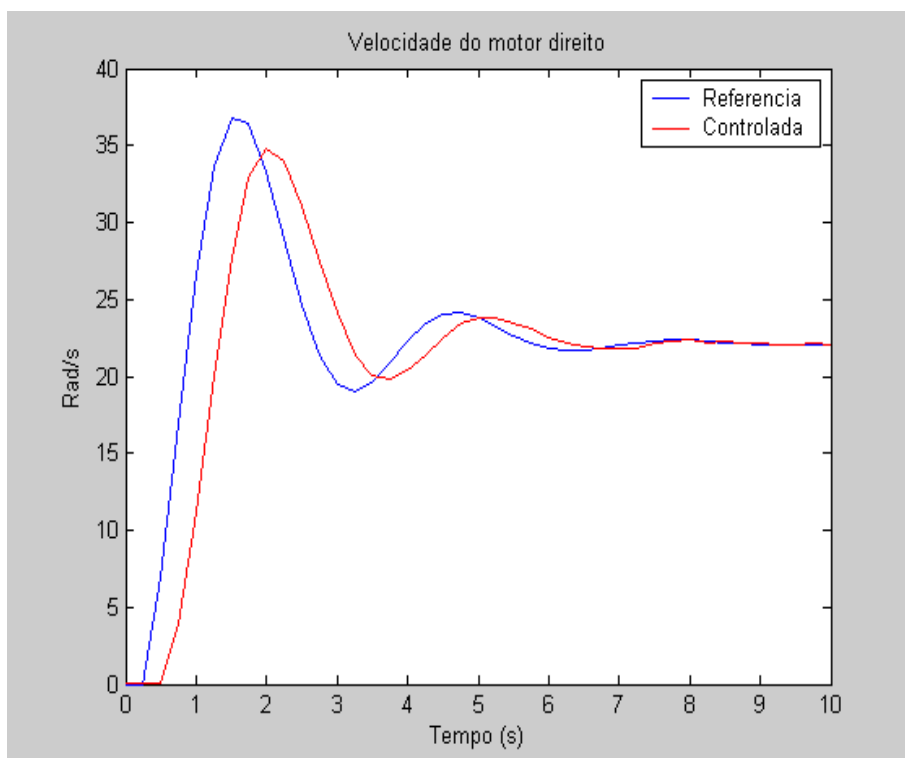


Figura 6.27 Controlador PID da malha interna (motor direito)



Figura 6.28 Trajetória com o controlador PID da malha externa

A figura 6.29 mostra a mesma trajetória do simulador do robô móvel com o controlador preditivo GPC.

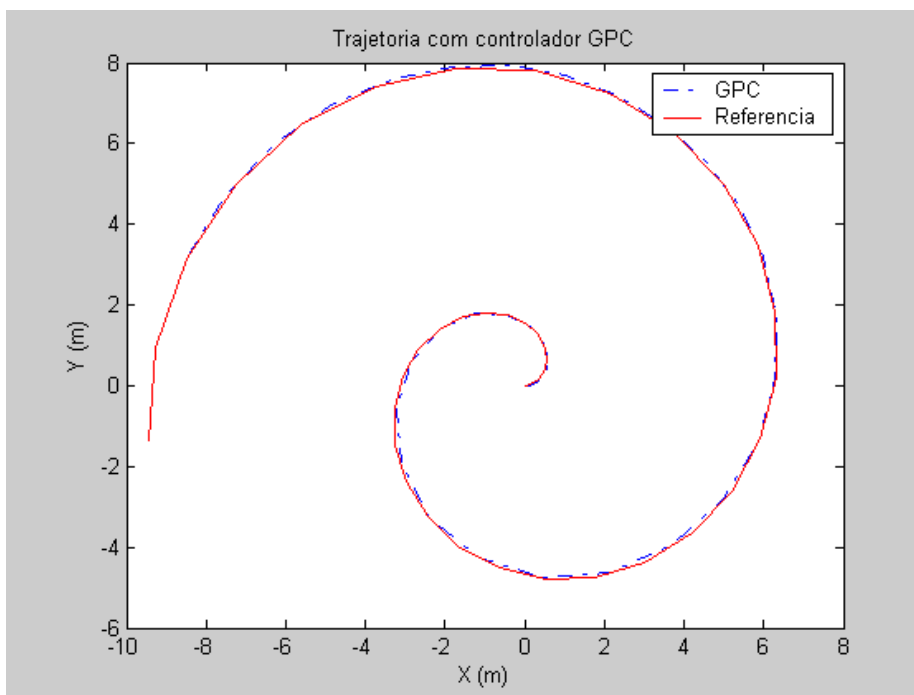


Figura 6.29 Trajetória com o controlador GPC da malha externa

Comparando-se as figuras 6.28 e 6.29, pode-se ver que os resultados obtidos foram próximos para ambos os controladores e seguiram bem a referência, sugerindo que os dois controladores executam essa tarefa de controle de maneira satisfatória.

As figuras 6.30 e 6.31 mostram o desempenho dos dois controladores em outra trajetória.

Neste caso o controlador PID mostrou um desempenho um pouco melhor do que o GPC. Foi utilizado para o controlador GPC um horizonte de controle de valor 8 e um valor de $\lambda=0.7$ e $\alpha=0$. A figura 6.32 mostra a mesma trajetória para o controlador GPC MIMO N=3.

As figuras 6.33, 6.34 e 6.35 mostram para a mesma trajetória o desempenho dos controlador PID, GPC (SISO) e GPC MIMO.

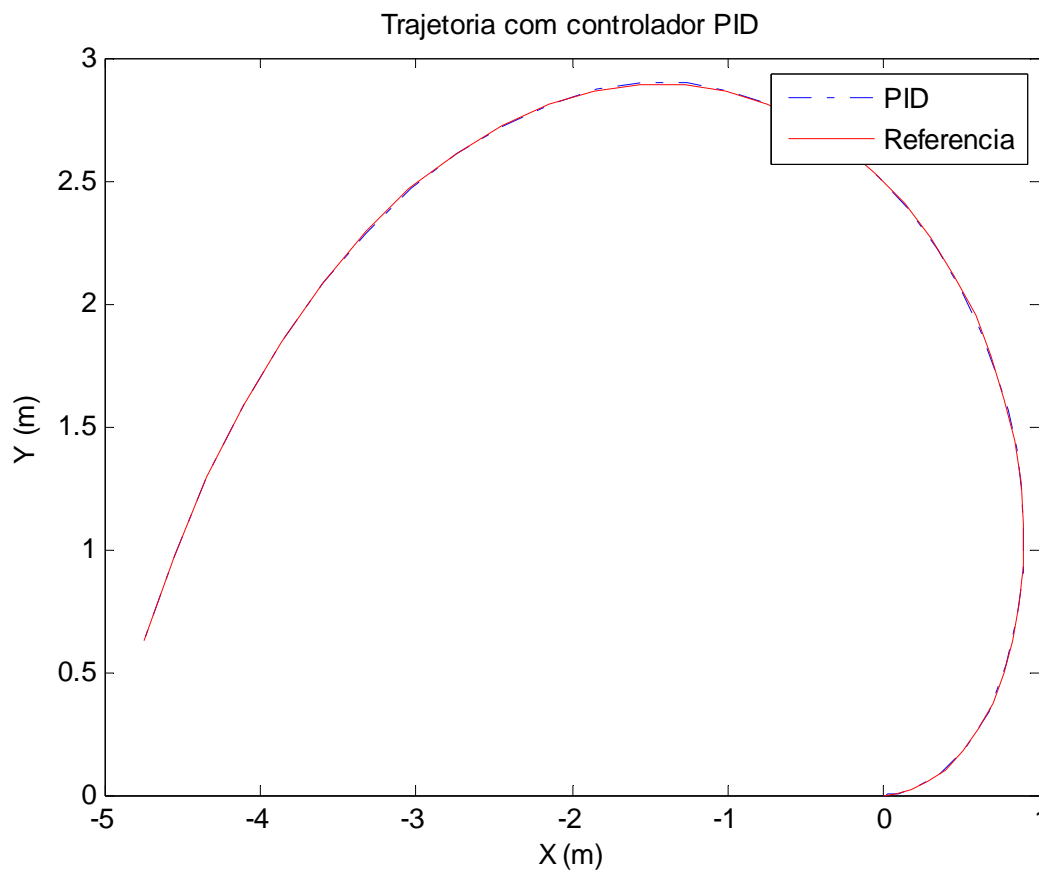


Figura 6.30 Trajetória com o controlador PID da malha externa

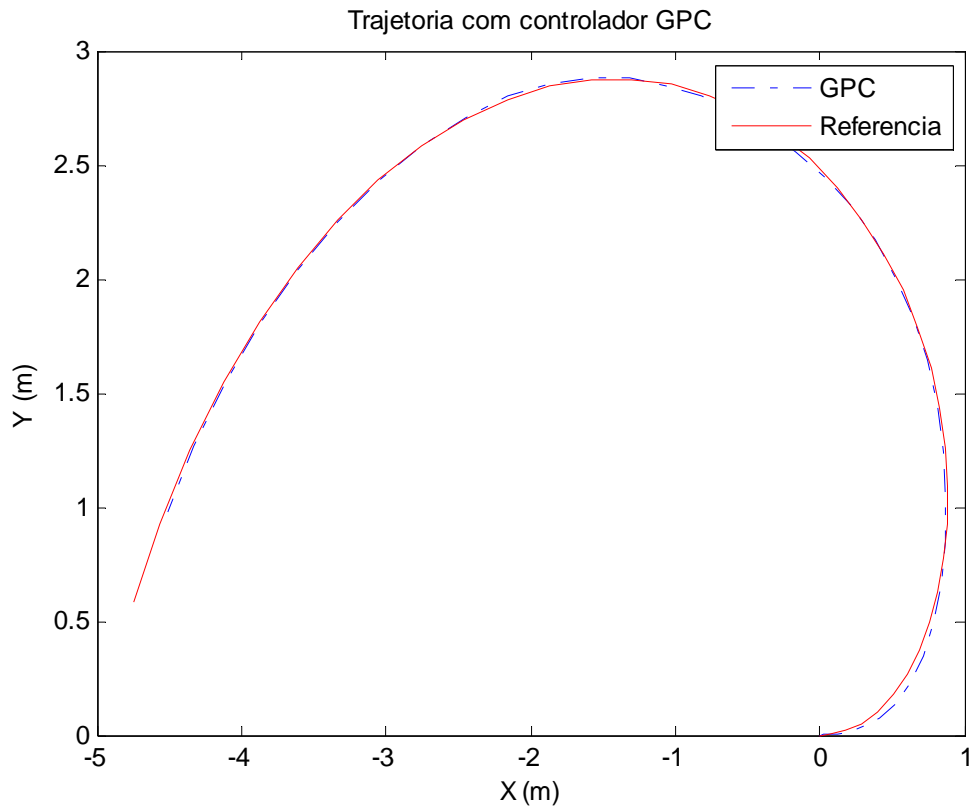


Figura 6.31 Trajetória com o controlador GPC da malha externa

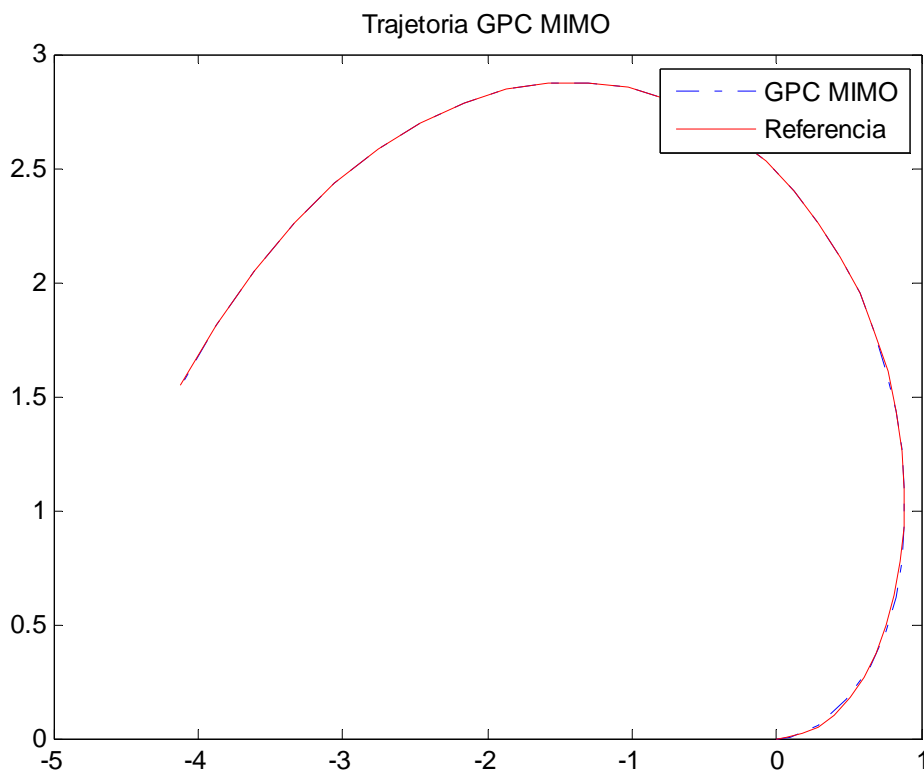


Figura 6.32 Trajetória com o controlador GPC MIMO

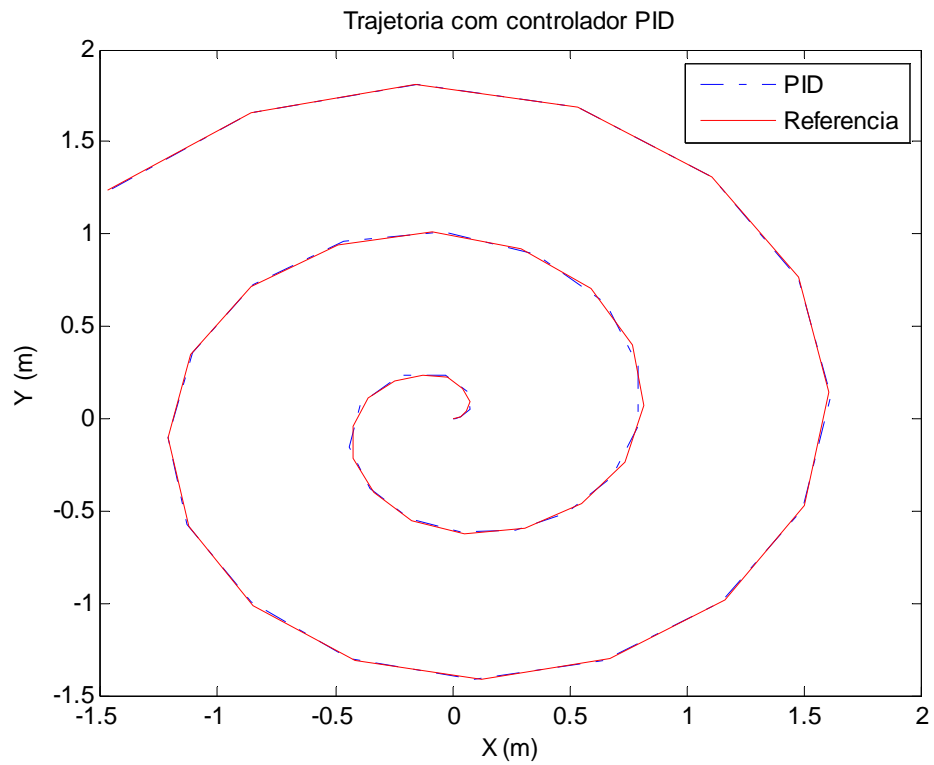


Figura 6.33 Trajetória com o controlador PID da malha externa

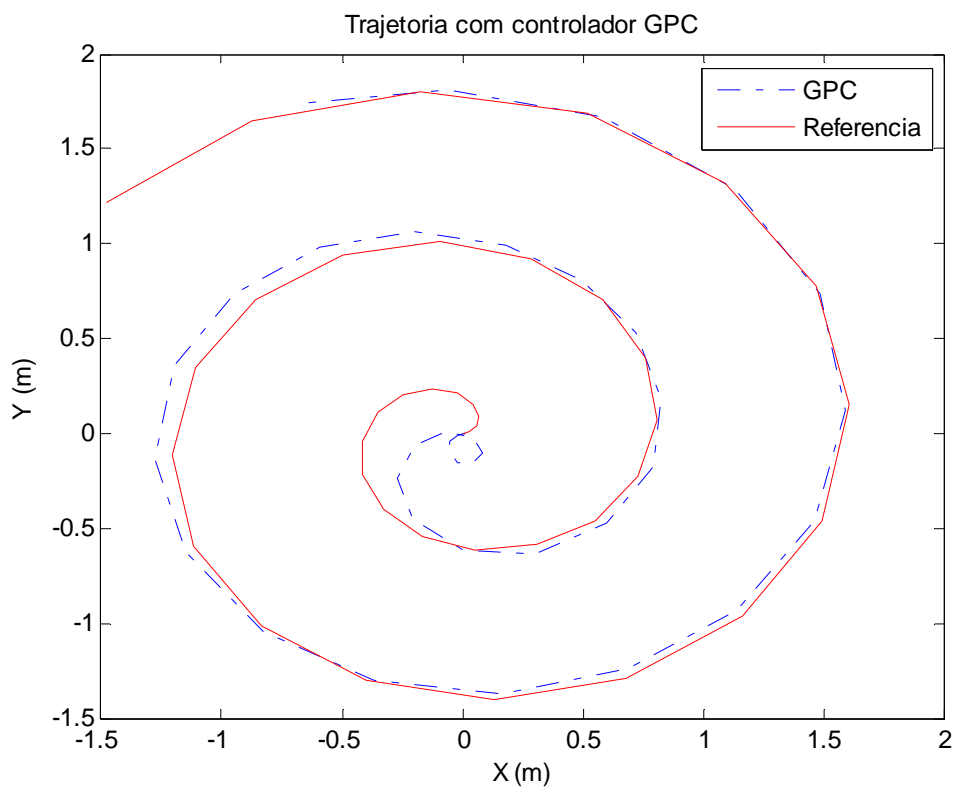


Figura 6.34 Trajetória com o controlador GPC da malha externa

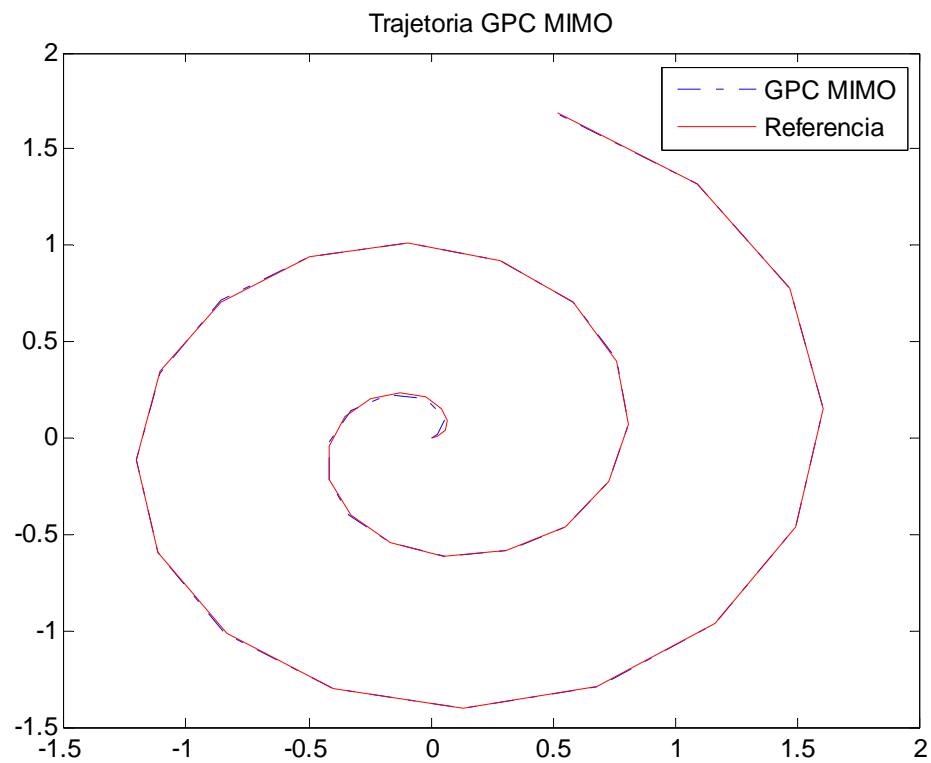


Figura 6.35 Trajetória com o controlador GPC MIMO

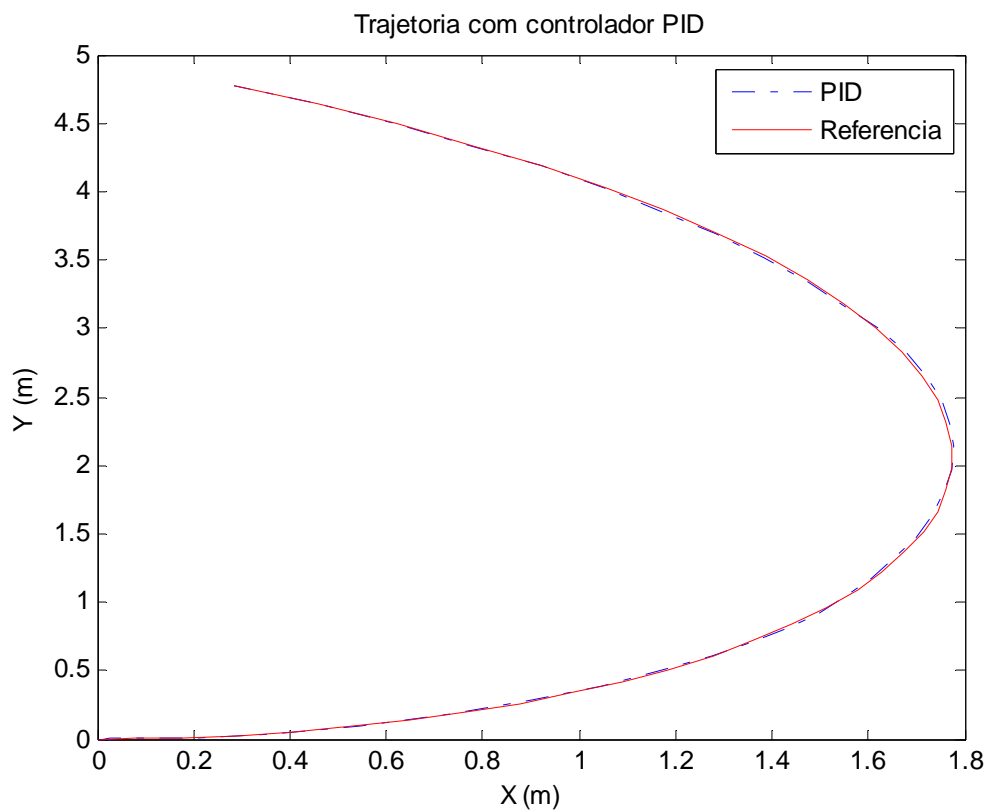


Figura 6.36 Trajetória com o controlador PID da malha externa

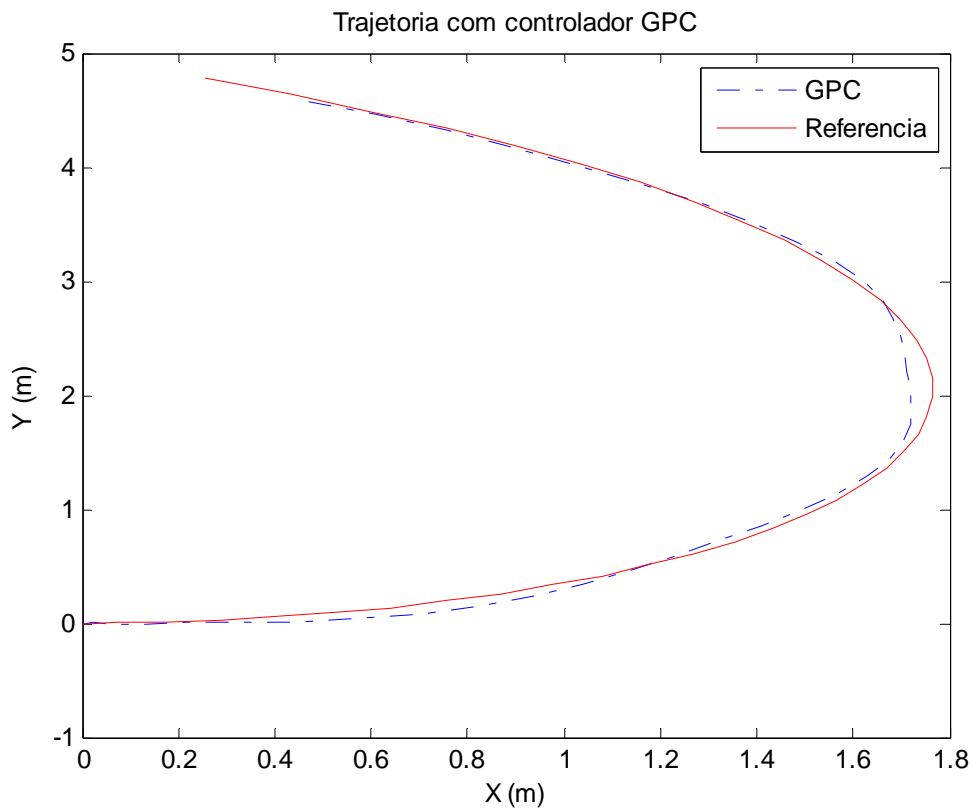


Figura 6.37 Trajetória com o controlador GPC da malha externa

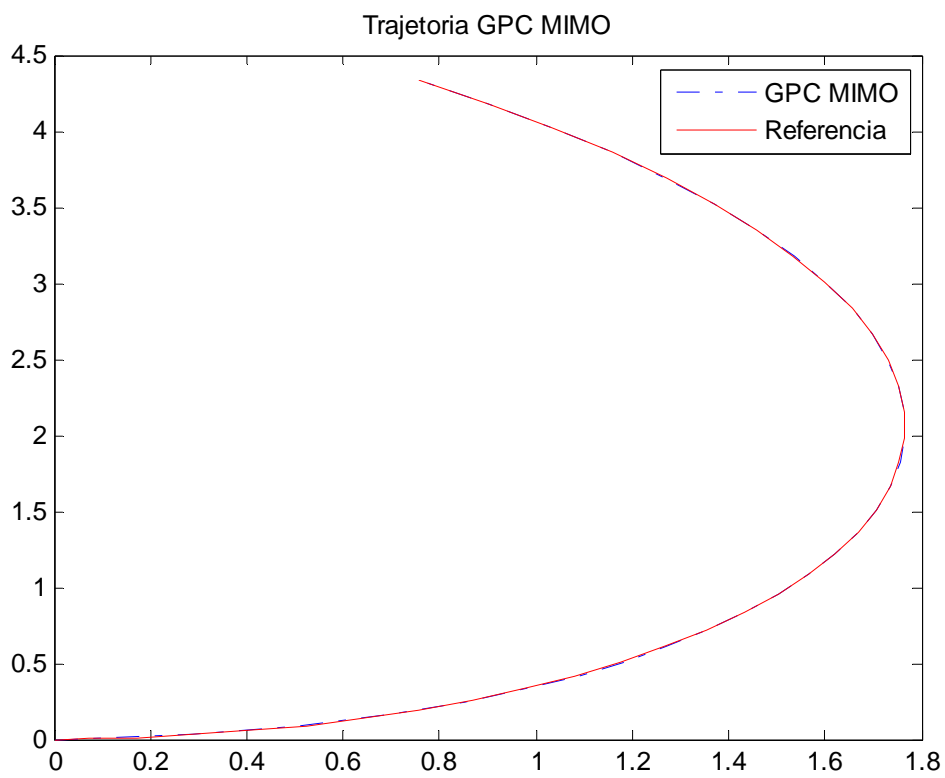


Figura 6.38 Trajetória com o controlador GPC MIMO

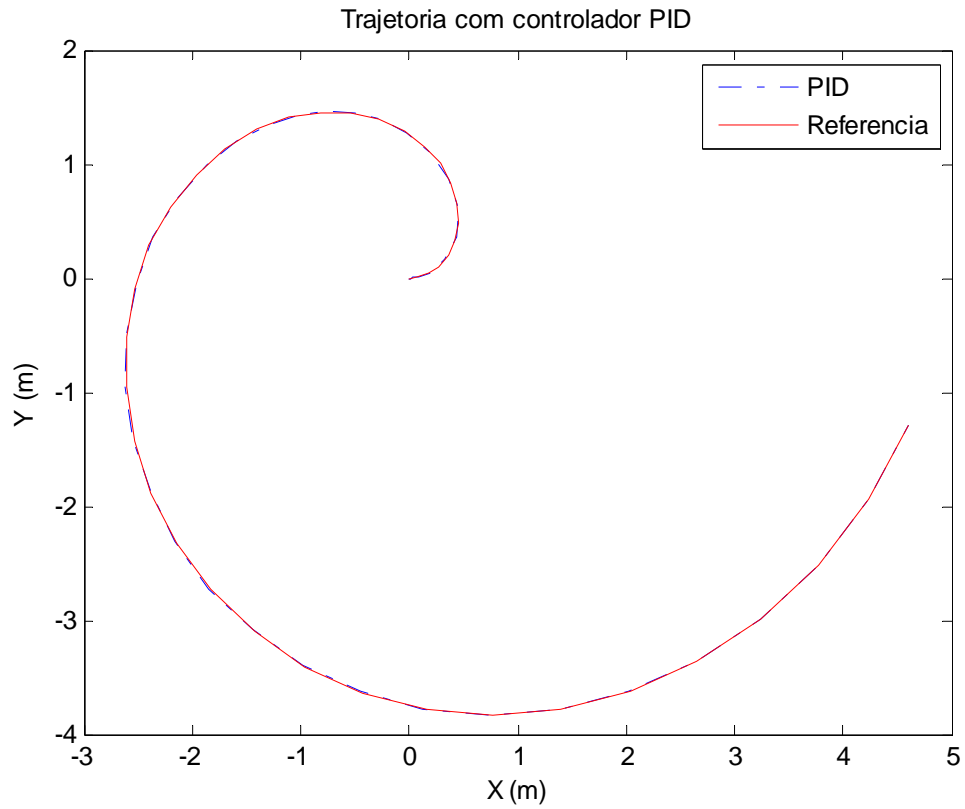


Figura 6.39 Trajetória com o controlador PID da malha externa

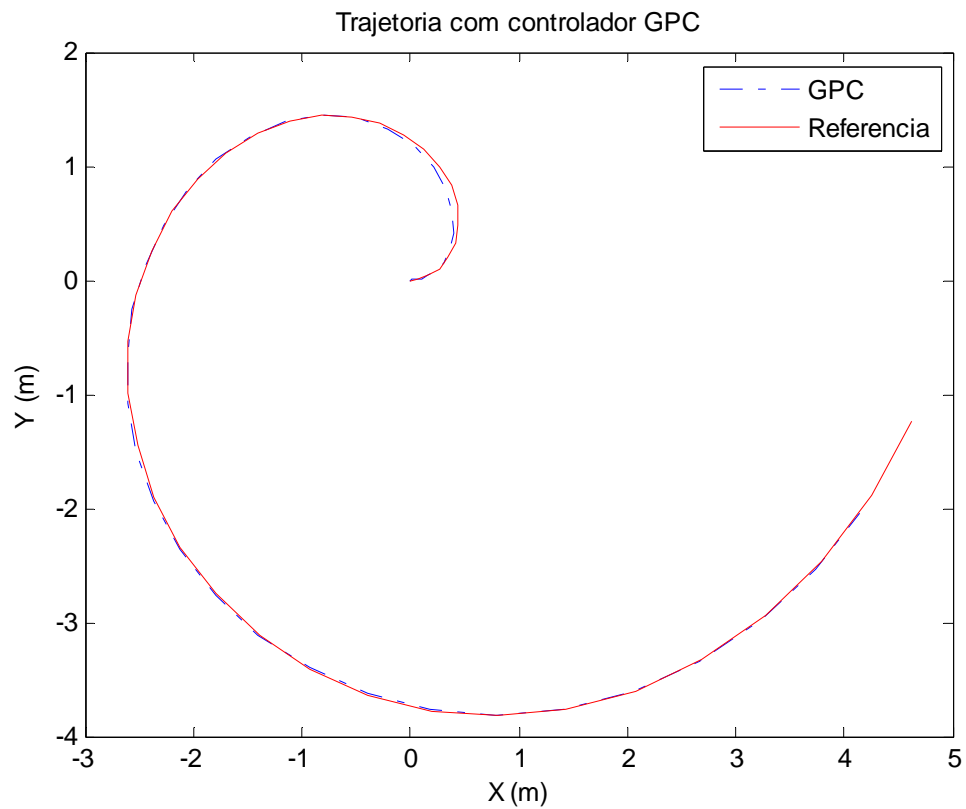


Figura 6.40 Trajetória com o controlador GPC da malha externa

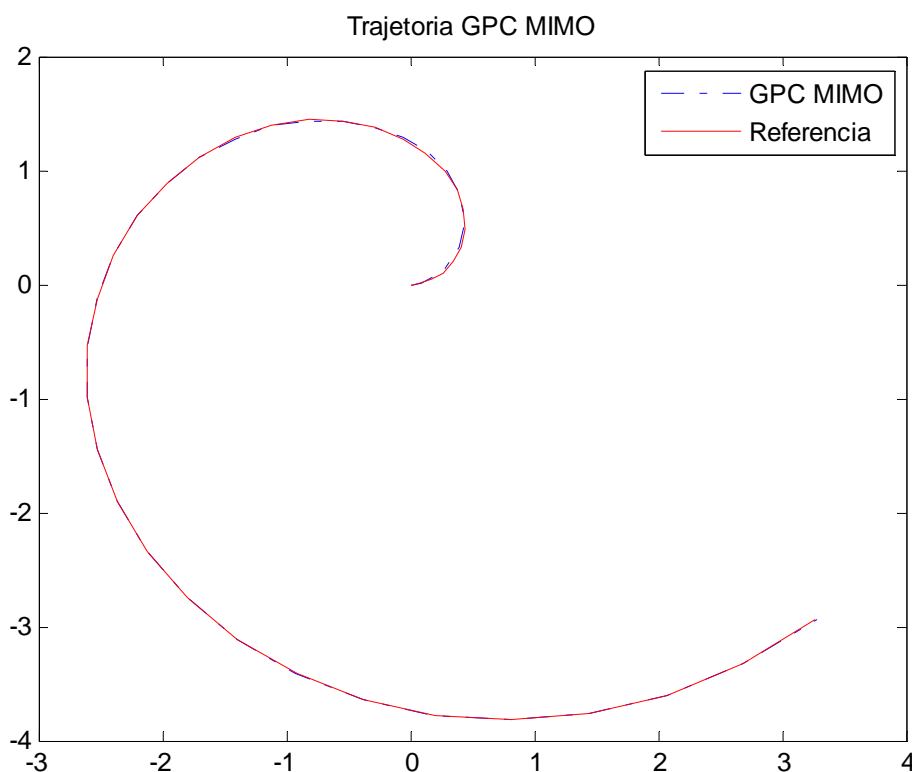


Figura 6.41 Trajetória com o controlador GPC MIMO

Em todos os casos, o controlador GPC MIMO apresentou os melhores resultados. O efeito do acoplamento pode ter influenciado no desempenho dos controladores SISO, entretanto esses controladores podem ser utilizados, pois mesmo apresentando resultados não tão próximos da referência, como o GPC MIMO, justifica-se pela simplicidade de implementação.

As figuras 6.42, 6.43e 6.44 mostram o desempenho do controlador GPC MIMO, estando o robô móvel fora da trajetória de referência. O controlador apresenta um desempenho bom no seguimento de trajetória, mesmo para trajetórias mais complexas.

A figura 6.45 mostra o seguimento de uma quase em linha reta. Essa trajetória pode ser utilizada para se verificar o comportamento das diferenças entre os parâmetros dos dois motores CC.

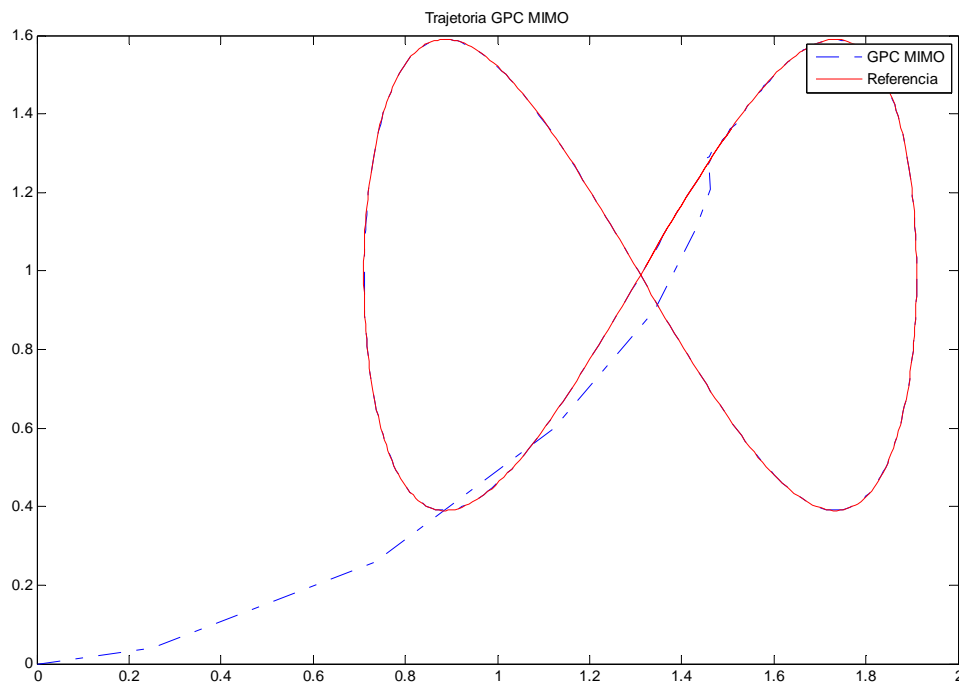


Figura 6.42 Trajetória com o controlador GPC MIMO

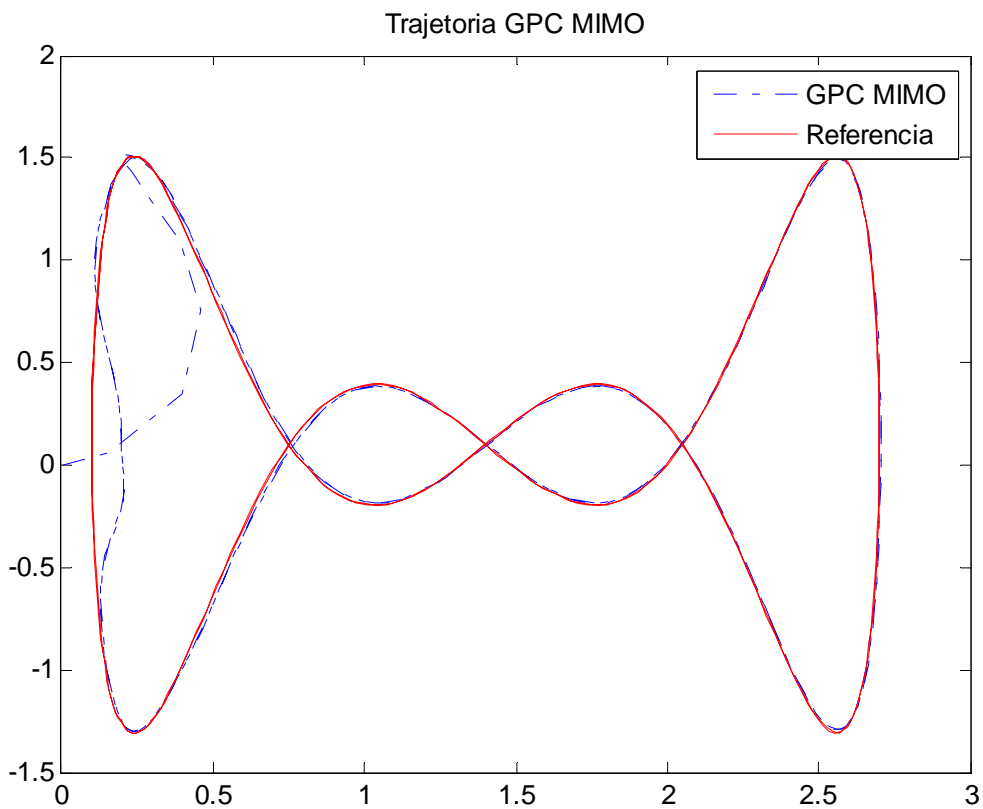


Figura 6.43 Trajetória com o controlador GPC MIMO

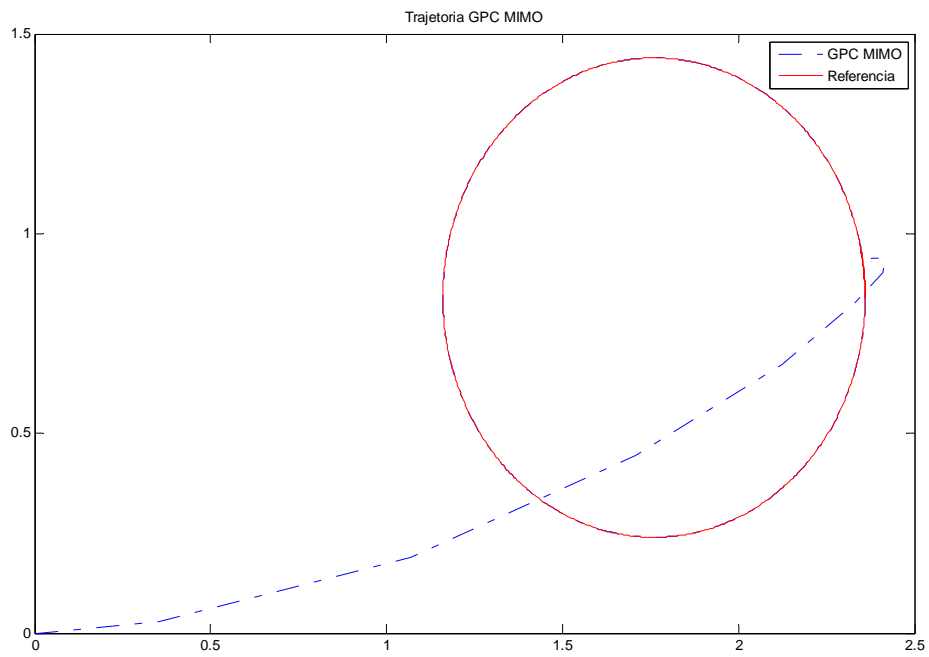


Figura 6.44 Trajetória com o controlador GPC MIMO

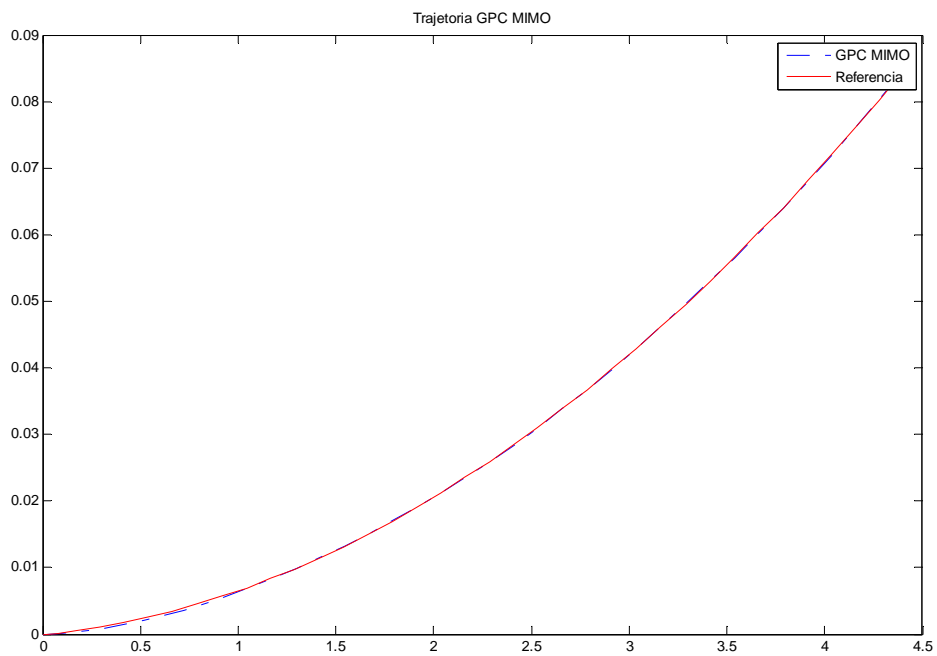


Figura 6.45 Trajetória com o controlador GPC MIMO

6.8 Resultados experimentais de seguimento de trajetória

As trajetórias experimentais estão mostradas nas figuras 6.46 a 6.49. As trajetórias escolhidas foram simples devido às limitações de posicionamento dos sonares. A medição de posição do robô é feita medindo-se a distância aos eixos de referência. À medida que o robô se desloca, faz-se necessário que os sonares sejam deslocados para se efetuar a medição de distância a estes eixos. Assim, os sonares foram montados em servomotores. Os servomotores de posição utilizados têm um ângulo de abertura de 300 graus.

Para trajetórias mais complexas, os servomotores não teriam condição de acompanhar o deslocamento angular.

Para essa trajetória, os resultados obtido pelo controlador GPC (monovariável) foram melhores. Na mudança de curvatura, o GPC (monovariável) consegue suavizar a execução da curva, apresentando um menor sobressinal.

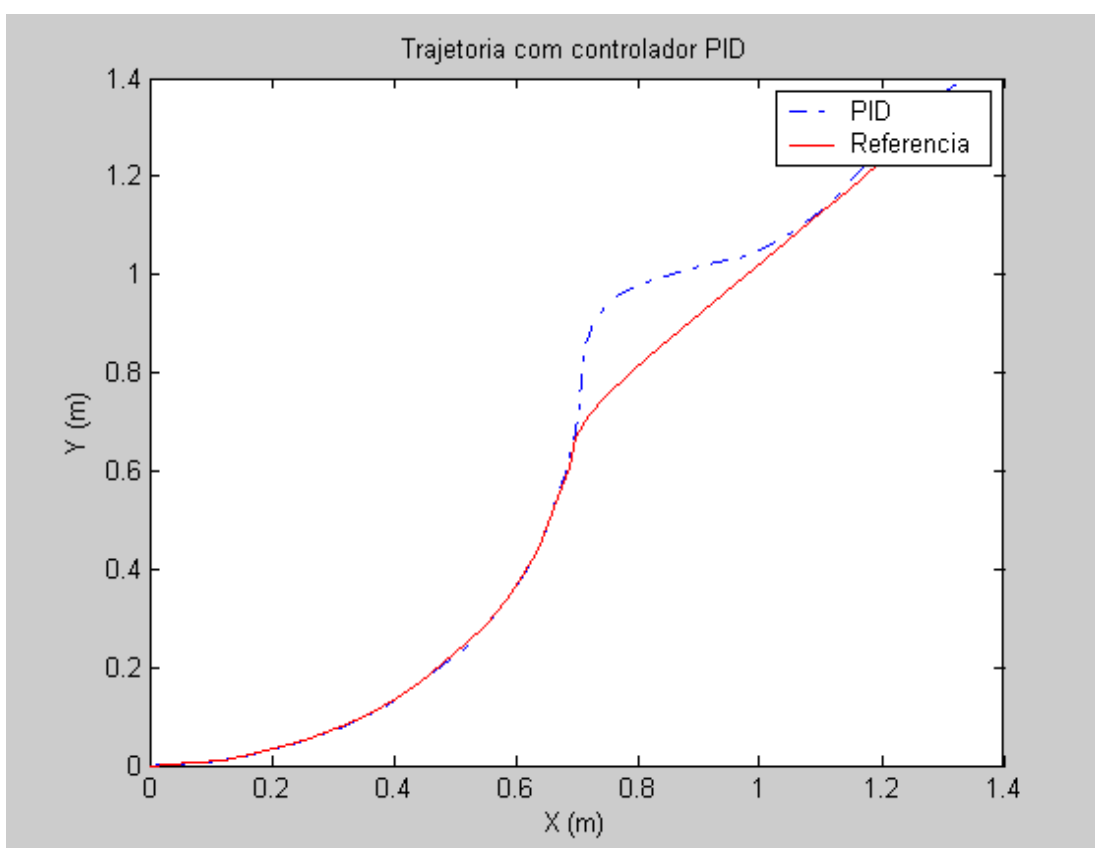


Figura 6.46 Trajetória real executada com o PID

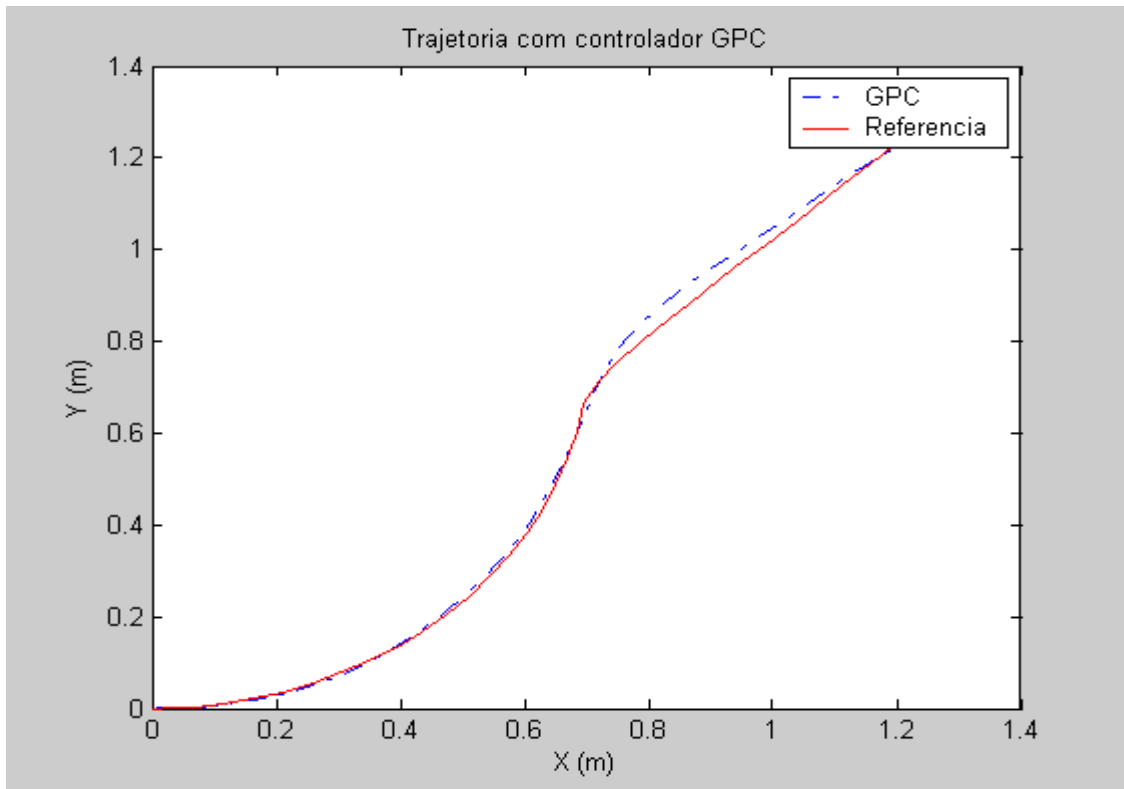


Figura 6.47 Trajetória real executada com o GPC

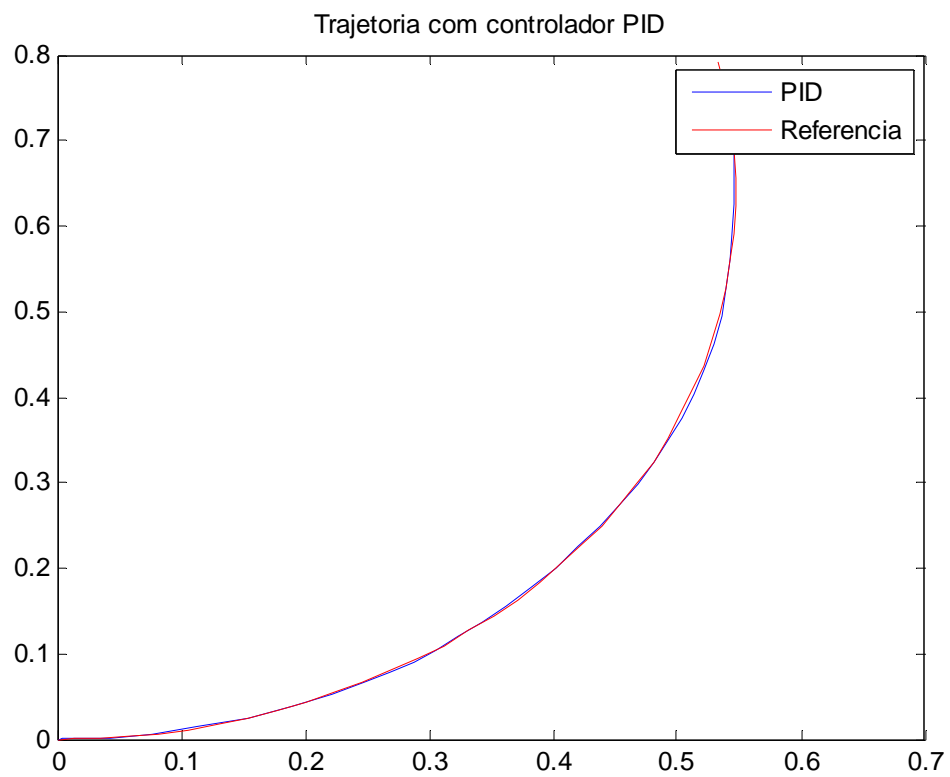


Figura 6.48 Trajetória real com o PID

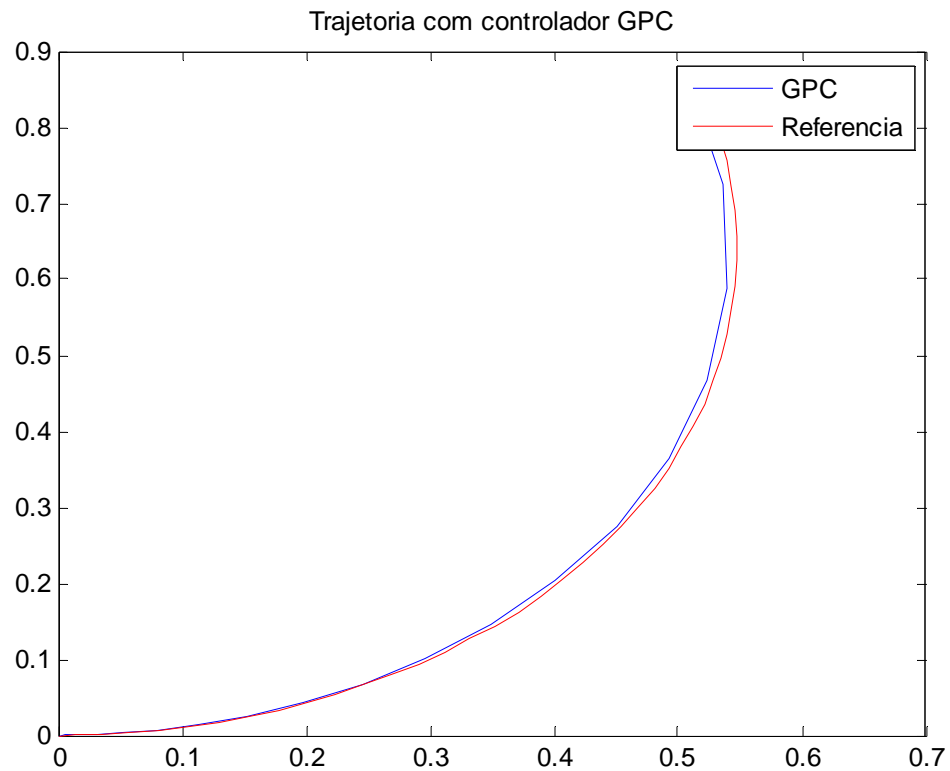


Figura 6.49 Trajetória real com o GPC

As figuras 6.48 e 6.49 mostram a execução de outra trajetória com os controladores PID e GPC. O desempenho foi satisfatório para ambos os controladores.

Conclusão do capítulo

O capítulo mostra os resultados obtidos com a simulação, bem como os valores obtidos experimentalmente. As técnicas de controle utilizadas e as dificuldades na utilização dos sensores. O simulador permite a mudança de parâmetros do sistema, como o raio das rodas do robô ou a distância entre as duas rodas, entre outras, de modo a se verificar o efeito destas na prática.

Conclusão e Sugestões de Trabalhos Futuros

O trabalho executado tem finalidade didática, sendo seu objetivo mostrar a utilização de técnicas utilizadas no controle de trajetória de pequenos robôs móveis.

Os resultados obtidos foram satisfatórios, considerando que os modelos utilizados foram simples. Em sistemas que exigem maior precisão, devem-se ser revistos os modelos e técnicas de identificação.

O controle atuou ainda de maneira satisfatória para a aplicação obtendo-se bons resultados tendo atingido os objetivos.

A arquitetura do sistema é relativamente simples, deixando todo o controle centralizado, liberando os controladores do robô de cálculos mais complexos; ficando para estes as tarefas de coleta de dados e execução da trajetória, permitindo a implantação de outras técnicas de controle futuras sem a necessidade de reformular o hardware do robô.

As principais dificuldades encontradas na execução do projeto foram referentes à tentativa de se construir um projeto que pudesse ser expandido, bem como o material utilizado, de baixo custo, como os sensores com certo nível de ruído e sem precisão.

Sugere-se como trabalhos futuros a inclusão de obstáculos, bem como a utilização de outros tipos de sensores, como câmeras, permitindo o uso de visão artificial. Pode-se ainda, partir para outras técnicas de identificação de parâmetros ou outros tipos de controladores.

A substituição dos sonares por câmera para localização, permitindo a execução de trajetórias maiores, bem como uma melhoria no desempenho, pois não é necessário o ajuste no ângulo dos servos que carregam os sonares, além de reduzir o peso do robô, melhorando o equilíbrio do mesmo.

Ainda será executada a redução do tamanho e confecção de outros robôs a fim de utilizar o trabalho em grupo, como o futebol de robôs, permitindo aspectos de cooperação entre eles.

Os controladores PID e GPC SISO mostraram um desempenho semelhante no controle de trajetória do robô móvel, sugerindo que o uso de qualquer um deles em sistemas desse tipo pode ser feito, entretanto O GPC MIMO apresentou resultados mais precisos,

provavelmente devido ao acoplamento existente no sistema, que não foi considerado nos anteriores.

A implementação em tarefas de controle sem a inclusão de obstáculos se mostrou, portanto satisfatória. Outros algoritmos de previsão podem ser utilizados, como métodos não-lineares, como o NEPSAC (*Non-linear Extended Self Adaptive Control*).

Os métodos utilizados mostraram simplicidade de implementação, sem sobrecarga computacional do sistema de controle.

ARTIGOS PUBLICADOS

Almeida, O. M. & Andrade, F. V. “Controle de Trajetória de Robôs Móveis: Implementação por Redes Neurais”, VIII Induscon, Poços de Caldas – MG, 2008.

Andrade, F. V. & Almeida, O. M. “Controle Neuro-Adaptativo de robôs móveis”, IX CBRN, Florianópolis – SC, 2007.

Andrade, F. V. & Torrico, B. C. “Controle Preditivo multi-variável de trajetórias de robôs móveis”, VIII SNCA, Salvador – BA, 2011.

REFERÊNCIAS BIBLIOGRÁFICAS

Aguirre, L.A. “Introdução a identificação de sistemas. Técnicas lineares e não-lineares aplicados aos sistemas reais”, 2ª ed. Editora da UFMG, Belo Horizonte - MG, 2004.

Almeida, O. M. “Métodos Frequentiais para ajuste de controladores PID”, Dissertação de Mestrado - UNICAMP, Campinas – SP, 1990.

Alves, A.C.R. “Estudo e Implementação de Técnicas de Sensoriamento Ultra-Sônico por Varredura em Robô Móvel”, Dissertação de Mestrado - UFC, Fortaleza - CE, 2002.

Astrom, K. J. & Haggund, T.: “PID Controllers”, 2ª ed Prentice Hall.

Astrom, K. J. & Witternmark B. “Computer Controlled Systems Theory and Design”, Prentice Hall, Englewood Cliffs, 1984.

Barreto, G.A., Araújo, A.F.R. & Rosa, M.O. “Algoritmo de Busca Heurística usando Redes Neurais Competitivas para Planejamento de Trajetória de um Robô Móvel”, CBRN, 1997.

Barreto, G.A., Araújo, A.F.R. & Rosa, M.O. “Algoritmo de busca heurística usando redes neurais competitivas para planejamento ótimo de trajetória de um robô móvel”, USP, São Carlos - SP.

Barreto, J.M. “Introdução às Redes Neurais Artificiais”, UFSC, Florianópolis - SC, 2002.

Batista, J.C. “Controlador PID auto-ajustável”, Dissertação de Mestrado - UNICAMP, Campinas – SP, 1985.

Bianchi, R.A.C., Simões, A.S. & Costa, A.H.R. “Comportamentos reativos para seguir pistas em um robô móvel guiado por visão”, USP, São Paulo – SP.

Borges, G.A., Deep, G.S. & Lima, A.M.N. “Controladores Cinemáticos de Trajetória para Robôs Móveis com Tração Diferencial”, VI SBAI, Bauru - SP, 2003.

Camacho, E.F. & Bordons, C. “Model Predictive Control”, Springer, London, 1999.

Camacho, E.F. & Normey-Rico, J.E. “Control of Dead Time Processes”, Springer, London, 2007.

Chen, C.T. “Linear System Theory and Design”, 3a ed. Oxford University Press, New York, 1999.

Chen, C.T. “Analog and Digital Control System Design: Transfer-Function, State-Space, and Algebraic Methods”, New York, 2004.

Coelho, A.A.R. & Coelho, L.S. “Identificação de Sistemas Lineares”, Editora da UFSC, Florianópolis - SC, 2004.

Deccó, C.C.G. “Construção de Mapas de Ambiente para Navegação de robôs móveis com visão omnidirecional estereo”, Tese de Doutorado - USP, Sao Paulo – SP, 2004.

Dorf, R.C. & Bishop, R.H. “Sistemas de Controle Modernos”, 8a ed. LTC, Rio de Janeiro - RJ, 2001.

Filho, O.G. “Contribuições à Análise de Robustez em Sistemas de Controle usando Redes Neurais”, Tese de Doutorado - UFRN, Natal - RN, 2004.

Groover, M. P. “Robótica: Tecnologia e Programação”, McGrawHill, São Paulo - SP, 1988.

Haykin, S. “Stable Adaptive Control and Estimation for Nonlinear Systems”, John Wiley & Sons, New York, 2002.

Haykin, S. “Redes Neurais Princípios e Prática”, 2^a ed. Bookman, Porto Alegre – RS, 2004.

Haykin, S. & Veen, B.V. “Sinais e Sistemas”, Bookman, Porto Alegre - RS, 2005.

Hoffmann, L.T. “Desenvolvimento de um Agente Móvel de Aprendizagem Neural para Ambientes Não-Estruturados”, Dissertação de Mestrado - INPE, São José dos Campos - SP, 2004.

Hoffmann, L.T. & Silva, J.D.S. “Aprendendo por esforço a evitar colisões em um robô móvel” INPE, 2004.

Huamani, I.R.L. “Redes Neurais Fuzzy Aplicadas em Identificação e Controle de Sistemas”, Dissertação de Mestrado - UNICAMP, Campinas – SP, 2003.

Júnior, A.V.S & Hemerly, E.M. “Control of Mobile Robots via based Wavelet Networks”, Revista da Sociedade Brasileira de Redes Neurais, Vol.1 No 4, pp. 235-252, 2004.

Junior, V.G. “Arquitetura híbrida para robôs móveis baseada em funções de navegação com interação humana”, Tese de doutorado - USP, São Paulo - SP, 2006.

Junior, V.G. & Jr, J.O. “Desenvolvimento de Software para Sistema de Controle em Tempo Real de Robô Móvel”, IV SBAI, São Paulo - SP, 1999.

Jung, C.R., Osório, F.S., Kelber, C.R. & Heinem, F.J. “Computação Embarcada: Projeto e Implementação de Veículos Autônomos Inteligentes”, XXV SBC, São Leopoldo - RS 2005.

Krose, B. & Smagt, P. Van der. “An Introduction to Neural Networks” 8^a ed., Amsterdam, 1996.

Kuhne, F. “Controle Preditivo de Robôs Móveis Não Holonômicos”, Dissertação de Mestrado - UFRS, Porto Alegre - RS, 2005.

Lages, W.F. “Controle e Estimação de Posição e Orientação de Robôs Móveis”, Tese de Doutorado - ITA, São José dos Campos - SP, 1998.

Liu, J. & Wu, J. “Multi-Agent Robotic Systems”, CRC Press, 2001.

Lu, C. H & Tsai, C. C. “Adaptive Neural Predictive Control for Nonlinear Systems with Time-Delay”, ICCA, Taipei, 2004.

Lu, C.H & Tsai, C.C. “Design and Experimental Evaluation of an Adaptive Predictive Controller using Recurrent Neural Network”, 2005.

Machado, J. B. “Modelagem e controle preditivo utilizando multimodelos”, Dissertação de Mestrado - UNICAMP, Campinas - SP, 2007.

Melo, L.F. “Proposta de um simulador virtual para sistema de navegação de robôs móveis utilizando conceitos de prototipagem rápida”, Tese de doutorado - UNICAMP, Campinas - SP, 2007.

Meleiro, L.A.C. “Projeto e Aplicação de Controladores Baseados em Modelos Lineares, Neurais e Nebulosos”, Tese de Doutorado - UNICAMP, Campinas - SP, 2002.

Muniz, L.A.R. “Controle Preditivo Adaptativo Aplicado a um Reator de Pirólise Operando em Regime Semibatelada”, Tese de Doutorado - UFSC, Florianópolis - SC, 2004.

Nise, N.S. “Engenharia de Sistemas de Controle”. 3a ed. LTC, Rio de Janeiro - RJ, 2002.

Ogata, K. “Engenharia de Controle Moderno”, 3ª ed. LTC, Rio de Janeiro - RJ, 2000.

Oliveira, V.M., Pieri, E.R. & Lages, W.F. “Wheeled Mobile Robot Control Using Sliding Modes and Neural Networks”, Revista da Sociedade Brasileira de Redes Neurais, vol.1 No 1, pp 117-134, 2003.

Ottoni, G.L. & Lages, W.F. “Navegação de Robôs Móveis em Ambientes Desconhecidos Utilizando Sonares de Ultra-Som”, Revista Controle & Automação, vol 14 No 4 pp 402-411, 2003.

Ou, J. & Rhinehart, R.R. “Grouped Neural Network model-predictive control”, 2002

Pazos, F. “Automação de Sistemas e Robótica” Axcel Books do Brasil , Rio de Janeiro - RJ, 2002.

Pedrosa, D.P., Medeiros, A.A.D. & Alsina, P.J. “Um Método de Geração de Trajetória para Robôs Não-Holonômicos com Acionamento Diferencial”, VI SBAI, Bauru - SP, 2003.

Pereira, F. “PIC Programação em C”, 1ª ed. Editora Érica, São Paulo - SP, 2003.

Pereira, G.A.S. “Identificação e controle de micro-robôs móveis”, Dissertação de mestrado - UFMG, Belo Horizonte - MG, 2000.

Quadrelli, G., Tanscheit, R. & Vellasco, M. M. “Neuro-Fuzzy Modelling and Control of Nonlinear Dynamic System”, Revista da Sociedade Brasileira de Redes Neurais, vol.1 No 1, pp 1-9, 2003.

Rosenwasser, Y. & Yusopov, R. “Sensitivity of automatic control systems”, CRC Press, 2000.

Santana, R.E.S. “Projeto de um robô bípede para a reprodução da marcha humana”, Dissertação de mestrado - USP, São Paulo - SP, 2005.

Selig, J.M. “Introductory Robotics” Prentice Hall, 1992.

Selvatici, A.H.P. “AREACT Uma arquitetura Comportamental Adaptativa para Robôs Móveis que integra visão, sonares e odometria”, Dissertação de mestrado - USP, São Paulo-SP, 2005.

Siegwart, R. & Nourbakhsh, I.R. “Introduction to Autonomous Mobile Robots”, Bradford books, London, 2004.

Simões, E.V. & Globo, C.A.M. “Um Controlador Neural Evolutivo Embarcado em FPGA para Robôs Móveis Autônomos”, XXVI SBC, Campo Grande - MS, 2006.

Souza, C.P. & Filho, J.T.C. “Controle por Computador Desenvolvendo Sistema de Aquisição de Dados para PC”, EDUFMA, São Luis - MA, 2001.

Tinos, R. “Tolerância a falhas em robôs manipuladores cooperativos”, Tese de doutorado - USP, Sao Carlos - SP, 2003.

Torrico, B.C. “Contribuições ao controle preditivo robusto de sistemas com atraso”, Tese de doutorado - UFSC, Florianópolis - SC, 2007.

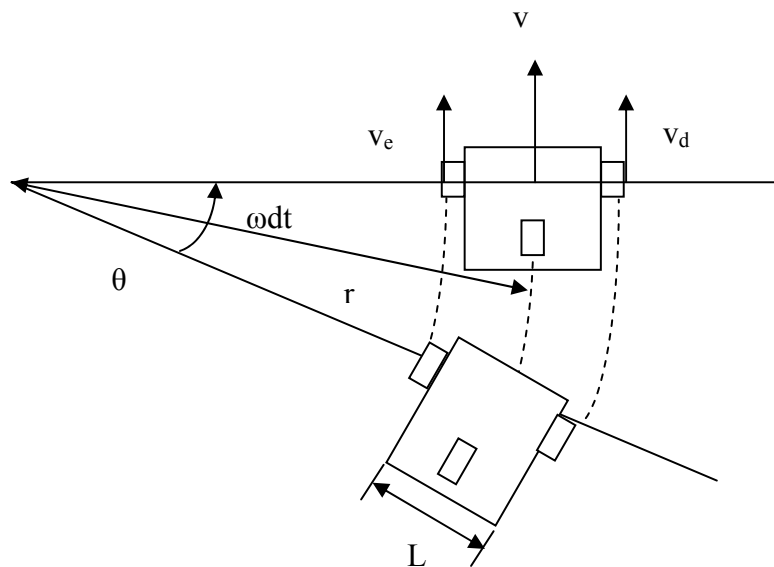
Vargas, J.A.R & Hemerly, E. M. “Identificação neural robusta de sistemas não-lineares” XII CBA, vol IV, pp.1285-1290, São José dos Campos - SP, 1998.

Vaz, M. J & Fabro, J.A. “SNMAP Sistema Neural de Navegação em ambientes pré-mapeados” IV CBRN, pp 118-123, São José dos Campos - SP: 1999.

Victorino, A.C. “Controle de trajetória e estabilização de robôs móveis não-holonômicos”, Dissertação de Mestrado - UNICAMP, Campinas - SP, 1998.

Vieira, F.C. “Controle Dinâmico de Robôs Móveis com Acionamento Diferencial”, Dissertação de Mestrado - UFRN, Natal - RN 2005.

APÊNDICE A – EQUAÇÕES CINEMÁTICAS



Os espaços percorridos pelas rodas direita e esquerda são dados por:

$$\Delta S_d = v_d dt \text{ e } \Delta S_e = v_e dt$$

As velocidades lineares das rodas são dadas por:

$$v_d = \omega_d r_d \text{ e } v_e = \omega_e r_e \quad (\text{A.1})$$

Onde: $r_d = r + \frac{L}{2}$ e $r_e = r - \frac{L}{2}$, substituindo-se em (A.1), resulta:

$$v_d = \omega \left(r + \frac{L}{2} \right) \text{ e } v_e = \omega \left(r - \frac{L}{2} \right)$$

Somando-se as equações anteriores, resulta:

$$v_d + v_e = \omega \left(r + \frac{L}{2} \right) + \omega \left(r - \frac{L}{2} \right) \Rightarrow v_d + v_e = 2\omega r \Rightarrow \omega r = v = \frac{v_d + v_e}{2}$$

Subtraindo-se as mesmas equações, resulta:

$$v_d - v_e = \omega \left(r + \frac{L}{2} \right) - \omega \left(r - \frac{L}{2} \right) \Rightarrow v_d - v_e = \omega L \Rightarrow \omega = \frac{v_d - v_e}{L}$$

APENDICE B – DIAGRAMAS ESQUEMÁTICOS

APÊNDICE C – FOTOS DO ROBÔ

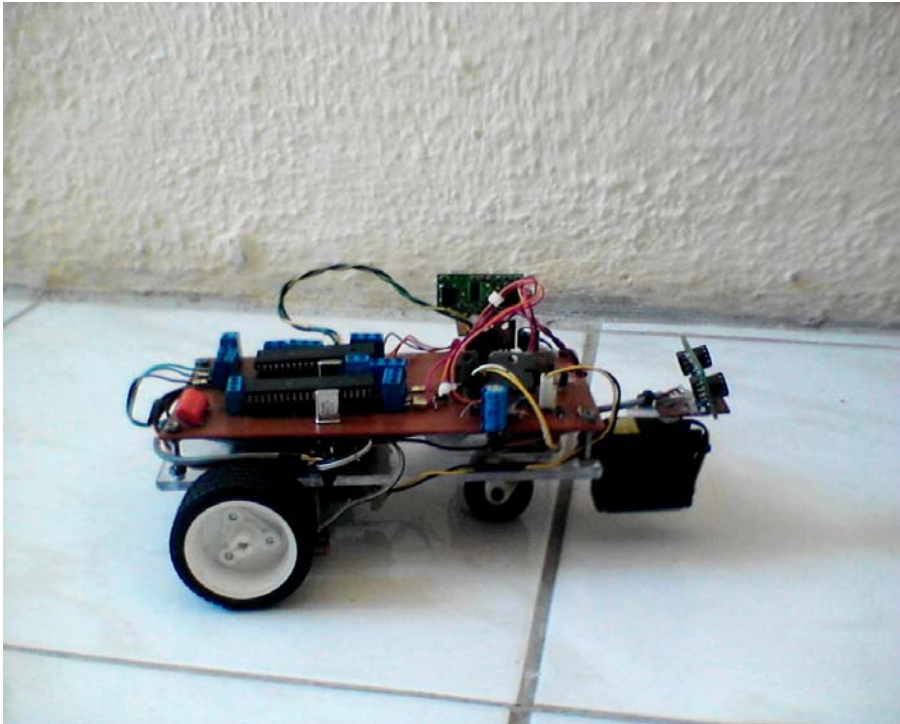


Figura C.1 Vista lateral esquerda

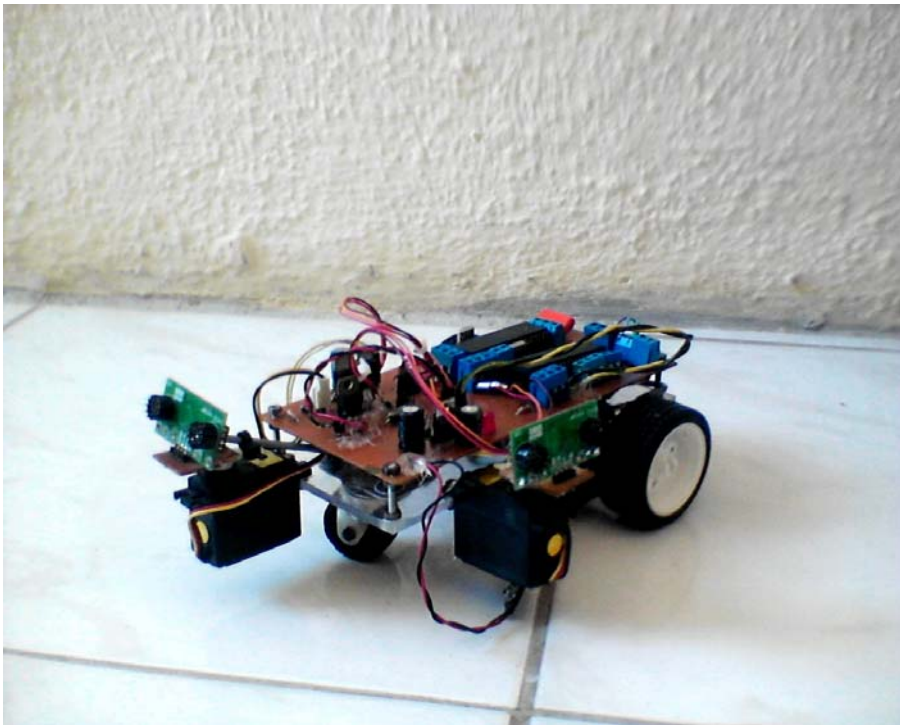


Figura C.2 Vista superior traseira

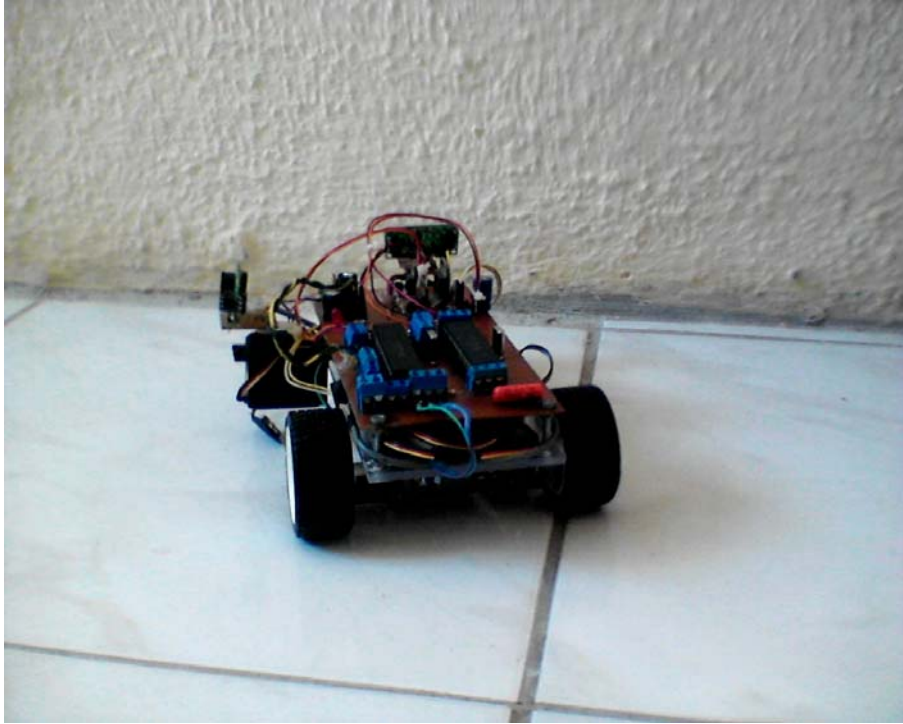


Figura C.3 Vista frontal

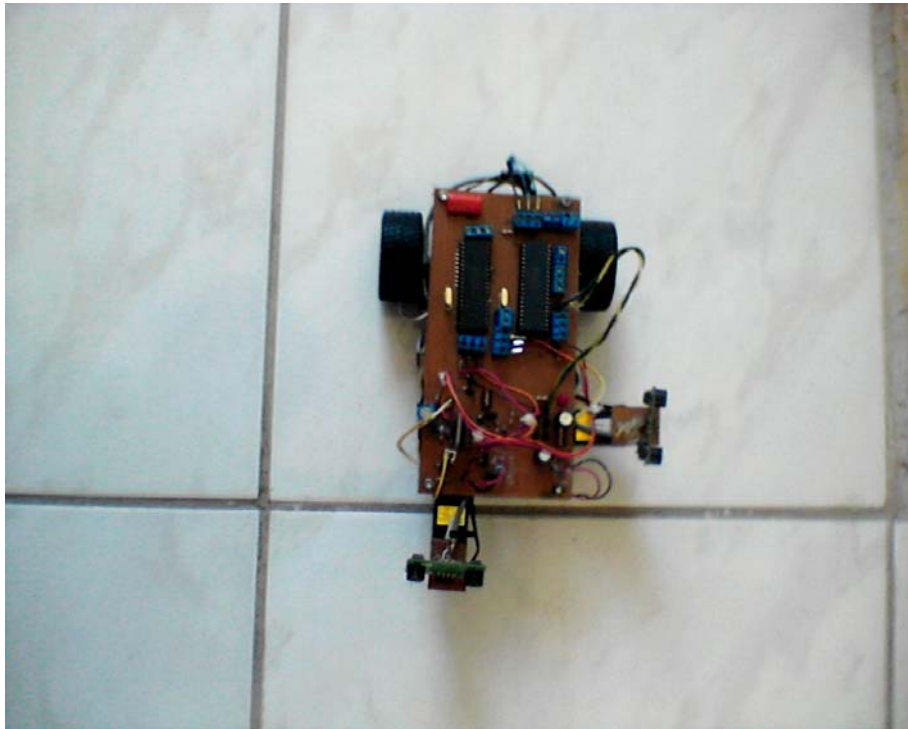


Figura C.4 Vista superior