



Universidade Federal da Bahia  
Escola Politécnica  
Instituto de Matemática  
Programa de Pós-graduação em Mecatrônica

Alexandre da Costa e Silva Franco

**GERAÇÃO E CONTROLE DE TRAJETÓRIA DE ROBÔS  
MÓVEIS OMNIDIRECIONAIS**

Salvador

2007

Alexandre da Costa e Silva Franco

# **GERAÇÃO E CONTROLE DE TRAJETÓRIA DE ROBÔS MÓVEIS OMNIDIRECIONAIS**

*Trabalho apresentado ao Programa de Pós-graduação em Mecatrônica, programa conjunto entre o Departamento de Engenharia Mecânica e o Departamento de Ciência da Computação, da Universidade Federal da Bahia como requisito parcial para obtenção do grau de Mestre em Mecatrônica.*

Orientador: Prof. Dr. Augusto Cesar Pinto Loureiro da Costa

Salvador

2007

*A toda minha família, em especial a meus pais, minha esposa Priscilla e minha filha Alyssa.*

## AGRADECIMENTOS

Em primeiro lugar a minha família, em especial a meus pais, minha esposa Priscilla e minha filha Alyssa, pelo apoio incondicional e pela paciência e compreensão nos momentos ausentes.

A Deus, doador da vida, pela saúde e força para que eu pudesse concretizar este trabalho.

Ao meu orientador, professor Dr. Augusto Cesar Pinto Loureiro da Costa, pela confiança depositada e pelo apoio técnico prestado.

A todos os professores do mestrado, em especial ao professor Dr. Leizer Schnitman, pelo conhecimento a mim passado, na área de controle, e por sugestões dadas ao meu trabalho. Não podendo passar em branco o meu agradecimento aos professores Herman Lepikson, Aline Andrade e Raimundo Macêdo, pelo reconhecimento e apoio necessários para que eu pudesse prosseguir no mestrado.

Ao colega Oswaldo Ludwig, pelo apoio técnico, pela agradável companhia, pelo prazer de ajudar e por ter contribuído com suas sugestões.

A todos os meus colegas do mestrado, em especial a Lourdes, Ebenezer e José Antônio, pela agradável convivência, pelas dificuldades enfrentadas juntos, durante as disciplinas do mestrado, e pelo conhecimento adquirido junto a eles.

Aos colegas Luciano Rebouças, Jorge Leite e Darmânio Cesar por me apoiarem, me dando força e ânimo nos momentos difíceis.

A todos os funcionários do PPGM, em especial a Lúcia Lago, pela paciência com

## AGRADECIMENTOS

---

os alunos e por estar sempre disposta a ajudar no que fosse preciso.

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

*Além da mente humana e como um impulso livre, cria-se a ciência.  
Esta se renova, assim como as gerações, frente a uma atividade que  
constitui o melhor jogo do homo ludens: a ciência é, no mais estrito e  
melhor dos sentidos, uma gloriosa diversão*

—JACQUES BARZUN

## RESUMO

Os robôs autônomos móveis são dotados de habilidades para executar tarefas com uma mínima intervenção do homem. Tais tarefas, no entanto, não seriam possíveis de serem realizadas se não fosse introduzida a estes robôs a capacidade da navegação autônoma. Uma simples tarefa de deslocar o robô de um ponto a outro, em um ambiente, requer a execução de um conjunto de etapas distintas. Em primeiro lugar é necessário o reconhecimento e a representação do ambiente. Esta representação é necessária para que possa ser feito, na etapa seguinte, um planejamento de trajetória entre um local de origem e de destino. A trajetória planejada é gerada na terceira etapa e o seu rastreamento é feito pelo robô, por meio de comandos enviados ao seu sistema de atuação, na etapa de mais baixo nível da navegação, o controle de trajetória.

Neste trabalho são propostas técnicas de geração e controle de trajetória para robôs móveis omnidirecionais, apresentando soluções para as duas etapas de mais baixo nível da navegação. Métodos de geração de trajetória por funções contínuas e por interpolação polinomial, são mostrados, e uma técnica composta por estes dois métodos é proposta. Esta técnica possibilita um planejamento de trajetórias por curvas e segmentos de reta, gerando trajetórias de formas variadas e suaves. São propostas técnicas de controle de trajetória usando leis de controle baseadas na linearização por realimentação de estado com compensadores de ruídos baseados em controladores difusos.

O robô usado como estudo de caso, o Axebot, um robô omnidirecional com três rodas, é apresentado. O desenvolvimento dos modelos cinemático e dinâmico da sua base móvel com inclusão do modelo dinâmico dos seus atuadores é mostrado.

Por fim, uma simulação em realidade virtual e os resultados experimentais são obtidos por meio de um módulo de geração e controle de trajetória, construído no ambiente Matlab *Simulink*<sup>®</sup>. Este módulo pode, no futuro, ser integrado ao sistema de navegação de um robô ou um veículo autônomo omnidirecional real.

**Palavras-chaves:** Robótica móvel, sistemas não-lineares, controle não-linear, cinemática, dinâmica, lógica difusa, inteligência artificial.

# ABSTRACT

The autonomous mobile robots are gifted of abilities to perform tasks with a minimum human interference. Therefore, those tasks would not be possible to be performed if was not introduced in the robots the autonomy navigation capacity. A simple task of dislocation of a robot from a certain point to other in an environment needs the execution of a group of distinct stages. In the first stage, it is necessary to recognize and to represent the environment. This environment representation is necessary to permit, in the next stage, the trajectory planning between the origin point and the final point destiny. The planned trajectory is generated in the third stage and the trajectory tracking is performed by the robot, sending commands for its actuators system, in the stage of lower level of the navigation, the trajectory control.

In this work are proposed generation and trajectory control techniques to omnidirectional mobile robots, showing solutions for the two lower level stage of the navigation. Trajectory generation methods by continuous functions and by polynomial interpolation are showed, and a technique composed by those both methods is proposed. This technique makes possible a planning of trajectories for curves and segments of straight line, generating smooth trajectories of varied forms. Trajectory control techniques are proposed using control laws based on feedback linearization and noise compensators based on fuzzy controllers.

The robot used in this study case, the Axebot, an omnidirectional robot with three wheels, is presented. The development of the mobile kinematics and dynamic models within the inclusion of its actuators dynamic model is showed.

Finally, a simulation in virtual reality and the experimental results are obtained

using a generation module and a trajectory control, built in the Matlab *Simulink*<sup>®</sup> environment. This module can be, in the future, integrated to the locomotion system of a robot or a real omnidirectional autonomy vehicle.

**Keywords:** Mobile Robotics, Non-Linear Systems, Non-Linear Control, Kinematics, Dynamics, fuzzy logic, artificial intelligence.

## LISTA DE FIGURAS

2.1	Etapas de planejamento, geração e controle de trajetória. . . . .	8
2.2	Interação entre as etapas de planejamento, geração e controle de trajetória. . . . .	9
2.3	Modelo do Axebot no <i>SolidWorks</i> . . . . .	14
3.1	Geração e controle de trajetória no espaço de juntas. . . . .	17
3.2	Geração e controle de trajetória no espaço de trabalho. . . . .	17
3.3	Curvas de posição, velocidade e aceleração usando a técnica de interpolação cúbica. . . . .	21
3.4	Curvas de posição, velocidade e aceleração para um polinômio de grau cinco. . . . .	23
3.5	Função <i>spline</i> cúbica natural $s(t)$ interpolando onze pontos $P_k$ . . . . .	29
3.6	Derivadas primeira e segunda da função <i>spline</i> cúbica natural $s(t)$ . . . . .	30
3.7	Diagrama de blocos do controle cinemático e dinâmico em cascata. . . . .	33
3.8	Diagrama de blocos do controle linearizante. . . . .	36
3.9	Funções de pertinência de uma variável X. . . . .	38
3.10	Exemplos de funções de pertinência das variáveis de um controlador difuso PI. . . . .	40
3.11	Diagrama de blocos de um controlador difuso PI. . . . .	41

4.1	Geometria do Axebot e seus sistemas de coordenadas. . . . .	45
4.2	Geometria da roda omnidirecional utilizada no Axebot. . . . .	48
4.3	Forças que atuam no Axebot. . . . .	52
4.4	Motor Maxon modelo Amax 22 R179-6V. . . . .	54
5.1	Diagrama de blocos do controle baseado na cinemática. . . . .	62
5.2	Diagrama de blocos do controle baseado na dinâmica. . . . .	64
5.3	Resposta do sistema às componentes de velocidade linear de referência do robô, não nula, e angular nula . . . . .	65
5.4	Resposta do sistema às velocidades angulares de referência dos motores com a velocidade angular de referência do robô nula . . . . .	66
5.5	Diagrama de blocos do controle em cascata. . . . .	67
5.6	Resposta do sistema às componentes de velocidade linear com a velocidade angular de referência do robô não nula . . . . .	68
5.7	Resposta do sistema às velocidades angulares de referência dos motores com a velocidade angular de referência do robô não nula, usando o controlador baseado na dinâmica . . . . .	69
5.8	Resposta do sistema às velocidades angulares de referência dos motores com a velocidade angular de referência do robô não nula, usando o sistema de controle em cascata . . . . .	70
5.9	Resposta do sistema às mudanças bruscas de posição de referência do robô . . . . .	71
5.10	Resposta do sistema às componentes de velocidade linear de referência do robô, geradas por mudanças bruscas de posição . . . . .	72

5.11 Resposta do sistema às velocidades angulares de referência dos motores, geradas por mudanças bruscas de posição, usando o controle baseado na dinâmica. . . . .	73
5.12 Resposta do sistema às velocidades angulares de referência dos motores, geradas por mudanças bruscas de posição, usando o controle em cascata	74
5.13 Diagrama de blocos dos compensadores difusos e sua conexão com os controladores. . . . .	76
5.14 Funções de pertinência das variáveis difusas . . . . .	77
5.15 Superfície de soluções. . . . .	78
5.16 Resposta do sistema, sujeito a ruídos de entrada, à posição de referência do robô, usando os controladores com e sem os compensadores de ruídos	79
5.17 Resposta do sistema, sujeito a ruídos de entrada, à velocidade de referência do robô, usando os controladores com e sem os compensadores de ruídos . . . . .	80
5.18 Resposta do sistema, sujeito a ruídos de entrada e saída, à posição de referência do robô, usando os controladores com e sem os compensadores de ruídos . . . . .	81
5.19 Resposta do sistema, sujeito a ruídos de entrada e saída, à velocidade de referência do robô, usando os controladores com e sem os compensadores de ruídos . . . . .	82
5.20 Resposta do sistema, sujeito a ruídos de entrada e saída, a sinais de posição senoidais, com e sem o uso dos compensadores de ruídos . . . .	83
6.1 Estrutura do gerador e controlador de trajetória. . . . .	88
6.2 Comportamento das trajetórias no eixo das ordenadas. . . . .	92
6.3 Comportamento das trajetórias no eixo das abscissas. . . . .	93

6.4	Primeira e segunda trajetória. . . . .	94
6.5	Função <i>spline</i> cúbica natural interpolando a ordenada dos onze pontos. . . . .	96
6.6	Função <i>spline</i> cúbica natural interpolando a abscissa dos onze pontos. . . . .	97
6.7	Trajétoria plana gerada pelos onze pontos. . . . .	98
6.8	Trajétoria elipsoidal. . . . .	99
6.9	Comportamento da trajetória elipsoidal no tempo. . . . .	100
6.10	Trajétoria baseada em comportamentos. . . . .	101
6.11	Geração de trajetórias baseadas em comportamentos com e sem o uso de funções <i>spline</i> . . . . .	102
6.12	Estrutura do módulo de geração de trajetória. . . . .	103
6.13	Resposta do sistema à trajetória de caminho livre fechada . . . . .	108
6.14	Esforço de controle do sistema à trajetória de caminho livre fechada . . . . .	109
6.15	Resposta do sistema à trajetória de caminho livre aberta . . . . .	110
6.16	Resposta do sistema à trajetória elipsoidal . . . . .	111
6.17	Resposta do sistema à trajetórias lineares com mudanças suaves e bruscas de velocidade e os esforços de controle . . . . .	112
6.18	Comportamento das trajetórias lineares, com mudanças bruscas e suaves, no tempo, sem a presença de ruídos . . . . .	113
6.19	Resposta do sistema às trajetórias baseadas em comportamentos, com e sem o uso da interpolação por funções <i>spline</i> . . . . .	114
6.20	Esforços de controle do sistema às trajetórias baseadas em comportamentos, com e sem o uso da interpolação por funções <i>spline</i> . . . . .	115
B.1	Blocos, no <i>simulink</i> , da geração e controle de trajetória . . . . .	139

B.2	Bloco do modelo cinemático em detalhes . . . . .	140
B.3	Bloco do modelo dinâmico em detalhes . . . . .	140
B.4	Bloco do controlador cinemático em detalhes . . . . .	141
B.5	Bloco do controlador dinâmico em detalhes . . . . .	141
B.6	Bloco dos compensadores de ruídos em detalhes . . . . .	142
B.7	Visões do Axebot geradas pelo bloco de realidade virtual . . . . .	142

## LISTA DE TABELAS

3.1	Base de regras do controlador difuso PI . . . . .	41
5.1	Valores dos parâmetros do Axebot e dos controladores . . . . .	64
5.2	Base de regras dos compensadores . . . . .	76
5.3	Análise dos controladores sob determinadas condições . . . . .	85

# SUMÁRIO

<b>Capítulo 1—Introdução</b>	<b>1</b>
1.1 Objetivo do trabalho . . . . .	3
1.2 Contribuições . . . . .	5
1.3 Estrutura do texto . . . . .	5
<b>Capítulo 2—Delineação do problema, objetos de estudo e a proposta</b>	<b>7</b>
2.1 Introdução . . . . .	7
2.2 O problema . . . . .	8
2.3 Objetos de estudo . . . . .	10
2.3.1 Robôs móveis autônomos . . . . .	10
2.3.2 Robôs móveis com rodas . . . . .	11
2.3.2.1 Estabilidade . . . . .	11
2.3.2.2 Manobrabilidade . . . . .	12
2.3.2.3 Mobilidade . . . . .	12
2.3.3 Robôs omnidirecionais . . . . .	12
2.3.4 O robô Axebot . . . . .	13
2.4 A proposta . . . . .	14

---

2.5	Conclusão . . . . .	15
<b>Capítulo 3—Geração e controle de trajetória de robôs móveis</b>		<b>16</b>
3.1	Introdução . . . . .	16
3.2	Geração de trajetória . . . . .	16
3.2.1	Interpolação polinomial . . . . .	19
3.2.1.1	Interpolação linear . . . . .	19
3.2.1.2	Interpolação cúbica . . . . .	20
3.2.1.3	Interpolação por polinômios de grau cinco . . . . .	22
3.2.2	Trajетória com <i>via points</i> . . . . .	24
3.2.3	Geração de trajetória por funções cúbicas <i>spline</i> . . . . .	25
3.3	Controle de trajetória . . . . .	30
3.3.1	Tipos de controles . . . . .	31
3.3.1.1	Controle baseado na cinemática . . . . .	31
3.3.1.2	Controle baseado na dinâmica . . . . .	32
3.3.1.3	Controle em cascata . . . . .	32
3.3.2	Controle linearizante . . . . .	33
3.3.2.1	Linearização Entrada-Saída por realimentação de estado . . . . .	34
3.3.2.2	Sistema na forma companheira . . . . .	37
3.3.3	Controladores difusos . . . . .	37
3.3.3.1	Introdução à lógica difusa . . . . .	37
3.3.3.2	Controlador difuso PI(Proporcional Integral) . . . . .	40

---

3.3.3.3	Controlador difuso PD(Proporcional Derivativo) . . . . .	41
3.4	Conclusão . . . . .	42
<b>Capítulo 4—Modelagem cinemática e dinâmica do robô Axebot</b>		<b>43</b>
4.1	Introdução . . . . .	43
4.2	Modelagem cinemática do Axebot . . . . .	44
4.2.1	Restrições cinemáticas da roda . . . . .	47
4.3	Modelagem dinâmica do Axebot . . . . .	51
4.3.1	Modelagem da base móvel . . . . .	52
4.3.2	Modelagem dos atuadores . . . . .	54
4.3.3	Modelo dinâmico da base móvel com inclusão dos atuadores . . . . .	57
4.4	Conclusão . . . . .	60
<b>Capítulo 5—Análise de desempenho dos controladores propostos. Análise baseada no modelo do robô Axebot</b>		<b>61</b>
5.1	Introdução . . . . .	61
5.2	Controle baseado na cinemática . . . . .	61
5.3	Controle baseado na dinâmica . . . . .	62
5.4	Controle em cascata . . . . .	67
5.5	Controle em cascata com compensadores difusos do tipo PI . . . . .	75
5.6	Análise de desempenho dos controladores . . . . .	84
5.7	Conclusão . . . . .	86

<b>Capítulo 6—Módulo de geração e controle de trajetória proposto para o robô Axebot, sua simulação e resultados</b>	87
6.1 Introdução . . . . .	87
6.2 Estrutura do módulo de geração e controle de trajetória . . . . .	88
6.3 Gerador de trajetória proposto . . . . .	89
6.3.1 Trajetórias lineares . . . . .	90
6.3.2 Trajetórias de caminho livre . . . . .	91
6.3.3 Trajetórias elipsoidais . . . . .	97
6.3.4 Trajetórias baseadas em comportamentos . . . . .	99
6.3.5 Implementação do gerador de trajetória . . . . .	102
6.4 Controlador de trajetória proposto . . . . .	105
6.5 Simulação e resultados . . . . .	105
6.6 Conclusão . . . . .	116
<b>Capítulo 7—Conclusão e trabalhos futuros</b>	117
<b>Apêndice A—Algoritmos de geração da trajetória</b>	125
<b>Apêndice B—Implementação dos modelos cinemático e dinâmico e dos controladores de trajetória, do Axebot, no Simulink</b>	138

-

# INTRODUÇÃO

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

A robótica móvel é um campo relativamente novo e em crescente desenvolvimento, envolvendo pesquisas e estudos de natureza multidisciplinar. No campo da engenharia, desperta pesquisas nas áreas de consumo de energia, teoria de controle, fadiga de materiais, nanotecnologia, entre outras. Na computação, sistemas em tempo real, tolerância a falhas, visão computacional, agentes autônomos e multiagentes e aprendizado por reforço são alguns dentre muitos exemplos de áreas que contribuem para a robótica móvel.

Os robôs móveis autônomos se caracterizam geralmente por exercerem tarefas individualmente, ou em cooperação com outros robôs. A autonomia e a cooperação são características que envolvem pesquisas nas áreas de inteligência artificial, sobretudo no campo de agentes autônomos e multiagentes.

A principal característica de um robô móvel é a sua capacidade de locomoção em um determinado ambiente. Em geral, esta locomoção é feita de forma autônoma. Segundo Lages (1998), a capacidade de locomoção autônoma torna-se atrativa ba-

sicamente devido a dois fatores: necessidade de interação com um ambiente pouco estruturado; necessidade de operação sem supervisão.

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

O mote principal da navegação autônoma de robôs é permitir que o robô navegue pelo ambiente em segurança, evitando a colisão com obstáculos, até alcançar uma ou mais posições pré-estabelecidas no ambiente. O desenvolvimento de robôs autônomos torna-se mais complexo à medida que o conjunto de características intrínsecas ao ambiente, no qual o robô está inserido, cresce. Para a interação de um robô em ambientes reais, deve ser dado a este a capacidade de contornar situações imprevistas e adaptar-se às mudanças. Uma das abordagens propostas para o problema da navegação autônoma de robôs é dividir a navegação em quatro etapas: mapeamento do ambiente, planejamento, geração e controle da trajetória.

O mapeamento do ambiente é a etapa de mais alto nível. A sua função é mapear o ambiente identificando os objetos nele contidos e suas respectivas localizações. Um ambiente pode ser estático, se suas características não mudam com o tempo, ou dinâmico, caso contrário. A etapa seguinte é o planejamento de trajetória, onde se define uma trajetória segundo algum critério, como por exemplo, o menor caminho entre uma posição de origem e de destino, passando por posições intermediárias livres de obstáculos. Estas duas primeiras etapas não fazem parte do escopo deste trabalho, tendo na literatura uma vasta referência: Adorno *et al.* (2005) usa um algoritmo de mapa de rotas probabilístico para o planejamento de trajetória de um robô omnidirecional; Vaz e Frabro (1999) executa a integração entre a abordagem de planejamento de trajetória baseada em um pré-mapeamento do ambiente e uma abordagem baseada em sensores; Pires *et al.* (2004) utiliza técnicas de algoritmos genéticos para o

planejamento de trajetória de manipuladores.

Uma vez planejada, a trajetória deve ser construída pela etapa de geração de trajetória, por meio de uma interpolação polinomial, ou por funções contínuas. O grande desafio desta etapa é encontrar polinômios interpoladores contínuos ou um conjunto de funções contínuas que constituam, com fidelidade, a trajetória planejada. Pode-se apontar alguns trabalhos dentro deste escopo: em Paromtchik e Rembold (1994) e Paromtchik e Asama (2000) são abordadas a técnica de geração de trajetória por funções *spline*; Almeida (2005) propõe uma técnica de geração de trajetória por curvas parametrizadas, através da técnica *B-Spline* com a utilização de um *algoritmo cultural* para o ajuste da curva; Andrade *et al.*(2001) e Garrido *et al.*(2002) utilizam redes neurais para geração de trajetórias.

A geração da trajetória constrói um caminho que deve ser rastreado pelo robô. O comportamento deste rastreamento é função da etapa de controle de trajetória. Nesta etapa, características como a dinâmica do robô e de seus atuadores, bem como a sua cinemática e as restrições cinemáticas do seu sistema mecânico de locomoção têm que ser levadas em consideração. Exemplos de trabalho sobre controle de trajetória de robôs móveis podem ser encontrados em Reis (2006) que implementa controladores robustos para o rastreamento de trajetória, Oubbati (2006) que aplica um controlador adaptativo para o controle de robôs móveis. Podem ser encontrados em alguns trabalhos o uso de controladores adaptativos baseados em redes neurais (GOMES, 2006; OLIVEIRA, 2001; NELSON, 1989, 1998; GU; HU, 2002; KIM; NECSULESCU; SASIADEK, 2001).

## 1.1 OBJETIVO DO TRABALHO

Dos diversos trabalhos sobre as etapas de geração e controle de trajetória, encontrados na literatura, poucos abordam as duas etapas em conjunto. Fatores que degradam o desempenho dos controladores, como, por exemplo, o deslizamento das rodas

do robô sobre o solo, podem ser minimizados por um comportamento de trajetória suave. Perturbações não modeladas e parâmetros variantes no tempo, ou difíceis de serem mensurados, no modelo do robô, podem fazer com que este não siga de maneira eficiente a trajetória. Estes problemas podem ser minimizados por compensadores de ruídos e ou técnicas de controle adaptativo.

O objetivo deste trabalho é prover, de forma parametrizada e de fácil integração com a etapa de planejamento, um módulo composto por um gerador e um controle de trajetória de um robô móvel omnidirecional.

Geralmente, é mais intuitivo planejar uma trajetória através de ações como: “seguir em frente, ou recuar a uma determinada distância, em uma determinada direção”, “Fazer uma curva fechada ou aberta, à direita ou à esquerda” ou, por exemplo, “Seguir em frente e fazer uma curva à direita”. Estas ações podem ser representadas por comportamentos, parametrizados e bem definidos, do robô. Uma técnica de geração de trajetórias por composição de funções lineares e elipsoidais é proposta para a execução destes comportamentos. A trajetória gerada por esta técnica é suavizada, por funções interpoladoras *spline*, eliminando mudanças bruscas de posição entre dois trechos de trajetória, evitando um alto esforço de controle dos atuadores e, conseqüentemente, a sua saturação.

É proposta uma estratégia de controle de trajetória híbrida, por controladores em cascata baseados nos modelos cinemático e dinâmico do robô, e compensadores de ruídos baseados em controladores difusos do tipo PI (proporcional integral).

Uma simulação foi feita no MatLab *Simulink*<sup>®</sup> com um módulo de realidade virtual mostrando os resultados obtidos. Foi usado como modelo de referência um robô omnidirecional, com três rodas, denominado de Axebot.

Este trabalho é parte de um projeto de escopo mais abrangente, denominado de Axebot, que tem como finalidade a concepção de um robô móvel omnidirecional dotado de habilidades para executar tarefas autônomas, em cooperação com réplicas do mesmo modelo, tendo como laboratório de testes o futebol de robôs.

## 1.2 CONTRIBUIÇÕES

Podem ser consideradas como contribuições deste trabalho:

- A integração, pouco abordada na literatura, de técnicas de geração e controle de trajetória para robôs móveis omnidirecionais com três rodas, mostrando a importância da geração de trajetórias por polinômios interpoladores contínuos, no desempenho dos controladores de trajetória.
- A concepção de um controle híbrido de trajetória, usando um módulo de controle difuso PI para minimizar os efeitos das perturbações não modeladas e das incertezas dos parâmetros do modelo do robô.
- A concepção de uma estrutura modular, composta por um gerador e um controle de trajetória, de fácil integração com um módulo de planejamento de trajetória.
- A concepção de uma técnica integrada de geração de trajetórias. Geração por meio de técnicas de funções contínuas e o uso de interpolações por funções cúbicas *spline*.

## 1.3 ESTRUTURA DO TEXTO

No capítulo 2 são apresentados a delimitação do problema, os objetos de estudo e as soluções propostas neste trabalho. O capítulo 3 apresenta uma introdução à geração e controle de trajetórias de robôs móveis, mostrando algumas técnicas de geração de trajetórias e algumas abordagens e técnicas de controle, utilizadas pelo gerador e controle de trajetória proposto. O capítulo 4 mostra o desenvolvimento dos modelos cinemático e dinâmico do Axebot e os aspectos relacionados com o controle de trajetória. O capítulo 5 faz uma análise dos controladores propostos, baseados na cinemática e na dinâmica do Axebot, e os compensadores de ruídos, baseados em controladores difusos do tipo PI. No capítulo 6 é abordado o módulo de geração e controle de trajetória

proposto, com sua simulação e resultados, mostrando a importância da geração de trajetória suaves e a eficiência dos compensadores difusos, em resposta a algumas trajetórias de referência. Por fim, são apresentadas as conclusões e trabalhos futuros, no capítulo 7.

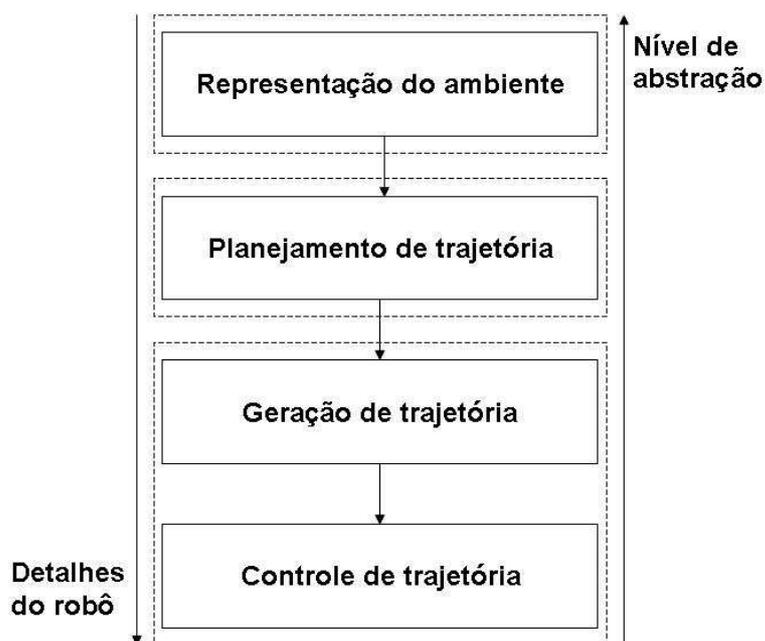
# DELINEAÇÃO DO PROBLEMA, OBJETOS DE ESTUDO E A PROPOSTA

## 2.1 INTRODUÇÃO

A função mais básica de um robô móvel é a sua locomoção, que geralmente é regida por um planejamento de trajetória e executada por uma estratégia de controle. Existem quatro etapas que devem ser cumpridas por um robô móvel autônomo para que esta função seja executada com eficiência, segundo a abordagem de navegação deliberativa. Estas etapas estão bem definidas em níveis hierárquicos de abstração, tendo no nível mais alto o mapeamento e a representação do ambiente, seguido da etapa de planejamento da trajetória. No terceiro e no quarto nível se encontram, respectivamente, a geração e o controle da trajetória (ver figura 2.1).

O planejamento da trajetória depende explicitamente do ambiente no qual o robô está inserido e da tarefa a ser executada por ele, podendo ser considerados alguns critérios como, menor gasto de energia na execução da trajetória, caminhos com menor número de obstáculos, tempo de execução do trajeto e diversos outros critérios que dependem do objetivo a ser alcançado.

A etapa de geração de trajetória constrói a trajetória especificada pela etapa de planejamento. Esta etapa recebe da etapa de planejamento parâmetros de referência da trajetória, sendo esta gerada por técnicas de interpolação polinomial ou de funções contínuas. Estas técnicas serão abordadas no próximo capítulo. Até esta etapa não é necessário um estudo sobre a mecânica e a parte elétrica do robô.

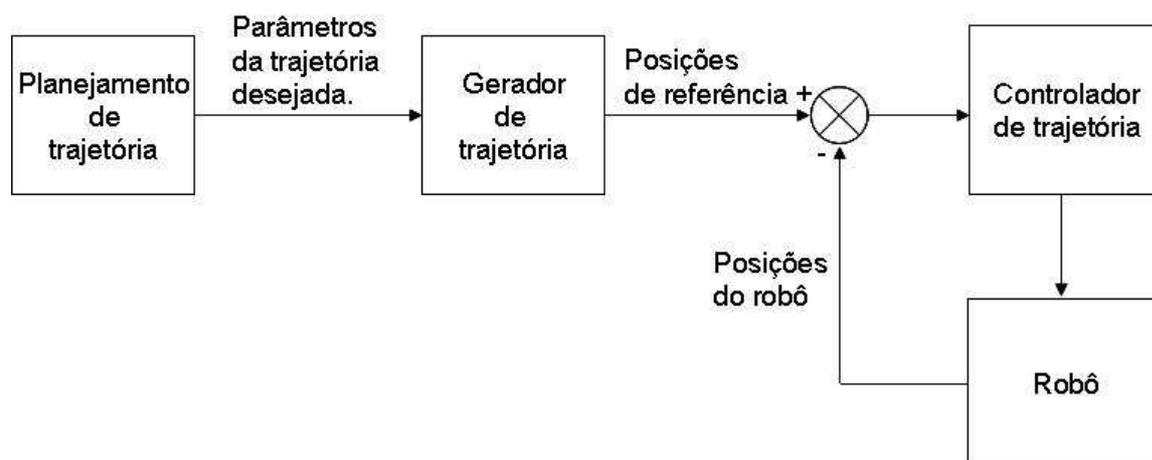


**Figura 2.1.** Etapas de planejamento, geração e controle de trajetória.

O nível de menor abstração, o controle de trajetória, tem como objetivo manter o robô móvel sobre uma trajetória de referência. É necessário, neste nível, conhecer detalhes sobre os aspectos mecânicos, geométricos e elétricos do robô e de seus atuadores, formulando os seus modelos dinâmico e cinemático e encontrando leis de controle que façam o robô convergir para a trajetória desejada, através de sinais elétricos enviados para os seus atuadores. A figura 2.2 mostra a interação entre as três etapas acima mencionadas.

## 2.2 O PROBLEMA

Algumas questões estão envolvidas no estudo da geração e do controle de trajetória para robôs móveis com rodas e devem ser analisadas cuidadosamente. Essas questões envolvem aspectos que vão desde como as trajetórias devem ser geradas, até como devem ser os projetos elétrico e mecânico do robô. Algumas dessas questões são citadas abaixo:



**Figura 2.2.** Interação entre as etapas de planejamento, geração e controle de trajetória.

- Como a disposição física dos componentes e a geometria do chassi do robô vão influenciar no seu centro de massa?
- Quantas rodas devem ser usadas, quantos graus de liberdade elas possuem e como elas devem estar dispostas sobre a base do robô?
- Quais tipos de atuadores devem ser utilizados?
- Quais as leis de controle de trajetória mais apropriadas?
- Quais devem ser as técnicas de geração de trajetória usadas?

Quanto mais baixo e mais próximo do centro geométrico do robô estiver o seu centro de massa, melhor será a sua estabilidade. Os aspectos mecânicos, elétricos e físicos do robô são relevantes para a formulação das leis de controle usadas no projeto do controlador de trajetória, pois influenciarão no modelo dinâmico e cinemático do robô. O número de rodas e as suas características físicas, além de influenciarem no projeto e no desempenho dos controladores, devido a sua influência no modelo cinemático do robô, são essenciais na estabilidade do mesmo sobre sua superfície de contato. Sua escolha vai depender também do tipo de terreno onde o robô se encontra.

Um outro aspecto relevante no desempenho do controlador de trajetória é o modo como a trajetória é gerada. Trajetórias compostas por funções não suaves podem gerar

mudanças bruscas de posição, velocidade e/ou aceleração do robô, o que eventualmente podem ocasionar um alto esforço de controle nos atuadores. Trajetórias longas devem ser evitadas, em ambientes dinâmicos, para diminuir a probabilidade de ser necessário um novo planejamento antes do cumprimento do planejamento atual.

### 2.3 OBJETOS DE ESTUDO

Os objetos de estudo deste trabalho são os robôs móveis omnidirecionais. O modelo de robô adotado e o motivo do uso de robôs deste tipo são discutidos na seção 2.3.4. As próximas seções discutem conceitos gerais sobre os robôs móveis, abordando o modelo de robô utilizado neste trabalho.

#### 2.3.1 Robôs móveis autônomos

Um robô móvel autônomo necessita de mecanismos de locomoção que o possibilite mover-se pelo seu ambiente de maneira autônoma, ou seja, sem a necessidade de ser controlado pelo homem. Existe uma grande variedade de movimentos que podem ser executados por um robô e a escolha de um sistema apropriado de locomoção é um aspecto importante para o seu projeto. A maioria dos mecanismos de locomoção usados nos robôs móveis é inspirada em modelos biológicos. Existe porém uma categoria que não segue este modelo, os robôs com rodas, também conhecidos como veículos, extremamente eficientes em superfícies planas.

Siegwart e Nourbakhsh (2004) classifica os robôs móveis em duas categorias, os robôs acionados por pernas e os robôs com rodas. Os robôs acionados por pernas são de mecânica complexa, com vários graus de liberdade e com a vantagem de se adaptar e se locomover em terrenos irregulares. Existem diversas classes de robôs dentro desta categoria dos quais podem se destacar os robôs humanóides, com duas pernas, os de quatro pernas e os de seis pernas, também conhecidos como robôs insetos. A principal

desvantagem da locomoção com pernas é a complexidade no mecanismo do robô e conseqüentemente no seu sistema de controle de posição e estabilidade. Os robôs que se enquadram nesta categoria são geralmente mais lentos que os robôs com rodas.

Os robôs móveis com rodas, ou simplesmente veículos, são bastante utilizados na robótica móvel por serem rápidos e de fácil locomoção e com uma mecânica de complexidade relativamente baixa, porém com difícil adaptação em certos terrenos. As rodas têm sido o mais popular mecanismo de locomoção. Além de se mostrarem bastante eficientes são mecanismos relativamente simples de serem implementados e garantem uma estabilidade melhor do que os robôs com pernas.

### 2.3.2 Robôs móveis com rodas

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

**2.3.2.1 Estabilidade** São necessárias, pelo menos, duas rodas para garantir a estabilidade de um robô. Robôs com duas rodas padrão fixa, chamados de robôs diferenciais, têm uma estabilidade garantida caso o seu centro de massa esteja abaixo do eixo das rodas. A garantia da estabilidade de um robô com três rodas é satisfeita quando o centro de massa do robô está localizado dentro do triângulo formado por elas. Robôs com mais do que três rodas necessitam de um mecanismo de suspensão de modo a garantir que todas as rodas permaneçam em contato com o solo diante de possíveis irregularidades do terreno ou de deformações de algumas das partes do robô. Este problema não acontece com três rodas, visto que, três pontos sempre definem um

só plano.

**2.3.2.2 Manobrabilidade** A manobrabilidade é a capacidade do veículo de mudar de direção em seu movimento. Esta capacidade depende do grau de liberdade das rodas e da sua disposição sobre a base do robô. Alguns robôs possuem a habilidade para mover-se em qualquer direção sobre um plano  $(x, y)$  sem a necessidade de mudar a sua orientação em relação ao eixo vertical ao plano. Por outro lado, os robôs diferenciais possuem restrições no seu movimento, dependendo da sua orientação para se deslocar em uma determinada direção.

**2.3.2.3 Mobilidade** Existem basicamente duas restrições impostas pelas rodas. A primeira aborda contato-rolamento, a roda deve rolar quando o movimento ocorre no sentido apropriado. A segunda reforça o conceito da não existência de deslizamento lateral, onde a roda não deve deslizar-se ortogonalmente ao seu plano (LAURA, 2006). A mobilidade é a capacidade que um robô tem de mover-se no ambiente. As restrições básicas que limitam a mobilidade são regras que cada roda deve satisfazer, sendo esta a restrição de deslizamento lateral. Uma análise mais detalhada pode ser vista em Champion *et al.*(1996) e Siegwart e Nourbakhsh (2004).

### 2.3.3 Robôs omnidirecionais

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

Os robôs omnidirecionais usam rodas que possuem rolamentos sobre a sua superfície de contato. Estes rolamentos diminuem o atrito de deslizamento lateral da

roda fazendo com que esta passe a ter um grau de liberdade a mais. Um robô com pelo menos três rodas deste tipo pode se deslocar em qualquer direção, independentemente da sua orientação.

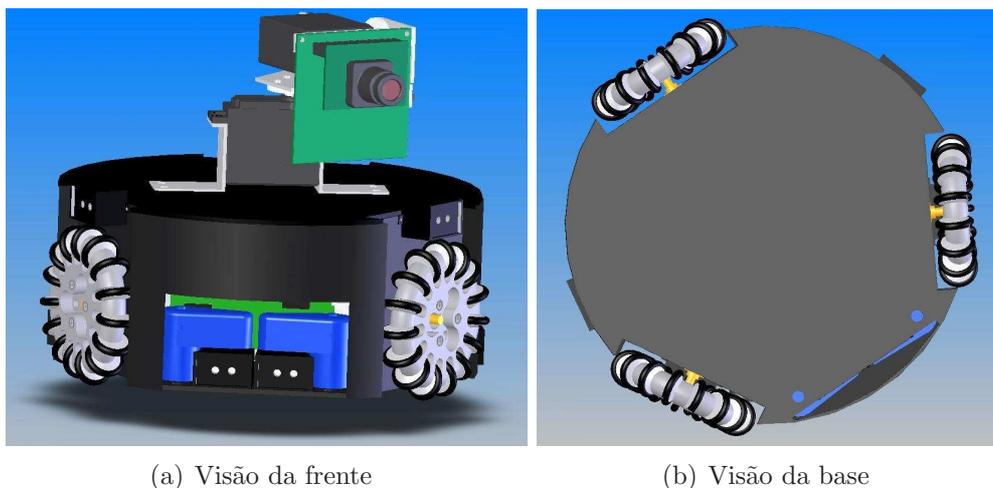
### 2.3.4 O robô Axebot

O robô Axebot é um robô omnidirecional, dotado de três rodas omnidirecionais dispostas a 120° umas das outras. Este robô faz parte de um projeto, cujo objetivo é desenvolver estudos nas áreas de robôs autônomos e de sistemas multiagentes. O laboratório para testes é o desafio proposto pela *F180 League* que visa a concepção de robôs dotados de habilidades para participar de partidas de futebol de robôs. O robô está em fase de construção. Seus modelos cinemático e dinâmico são mostrados no capítulo 4.

O sistema de atuação do Axebot é composto por motores de corrente contínua da marca Maxon, modelo *AMax-22*, controlados por duas placas *BrainStem Moto 1.0*. Seu sistema computacional é um microcontrolador *Strongarm*. O seu mecanismo de chute é acionado por dois servo-motores e o seu sistema de visão é composto por uma WebCam *CMUCam2* e 6 sensores de distância da sharp *GP2D120*. A figura 2.3 mostra um modelo reproduzido no *SolidWorks*, do Axebot.

Dentro do desafio proposto pela *F180 League*, como laboratório de testes para o projeto Axebot, a escolha de um robô omnidirecional com três rodas foi considerada a mais adequada devido a alguns fatores citados abaixo:

- O sistema de locomoção omnidirecional permite que o robô translate em qualquer direção sem a necessidade de se reorientar, ou seja, a translação é independente da rotação, permitindo uma maior flexibilidade no deslocamento do robô. Esta flexibilidade é fundamental para a tarefa proposta ao robô.
- O sistema de equações do movimento de um robô omnidirecional, com três rodas,



**Figura 2.3.** Modelo do Axebot no *SolidWorks*

é determinado e admite uma única solução, facilitando encontrar uma lei de controle por realimentação.

- Três rodas sempre estão em contato com o plano, mesmo que o chassi do robô, apresente uma falha mecânica no seu projeto. A falta de contato de uma ou mais rodas, com o solo, acarretará em um erro de cálculo da postura do robô, prejudicando o seu sistema de controle.
- Um robô com quatro ou mais rodas, necessita de um mecanismo de suspensão, para que todas as rodas se mantenham em contato com o solo.
- O controle independente da orientação e posição de um robô com duas rodas não é possível. Como mencionado anteriormente, a estabilidade de um robô com duas rodas só é garantida se o seu centro de massa estiver abaixo do eixo das rodas. Geralmente, é usada uma roda livre, para a garantia da estabilidade.

## 2.4 A PROPOSTA

A proposta deste trabalho é a concepção de um módulo de geração e controle de trajetória, levando em consideração os problemas discutidos na seção 2.2.

## 2.5 CONCLUSÃO

---

O módulo de geração e controle de trajetória que se propõe neste trabalho possui as seguintes características:

- Módulo parametrizado e de fácil integração com o planejamento de trajetória.
- Geração de trajetórias suaves, através da técnica de interpolação por funções cúbicas *spline*.
- Geração de trajetórias por conjuntos de funções contínuas.
- Controle de trajetória com compensadores de ruídos baseado em controladores difusos do tipo proporcional integral.

Detalhes sobre este módulo, sua simulação e resultados, são mostrados no capítulo 6

## 2.5 CONCLUSÃO

Este capítulo apresentou, de forma geral, o problema tratado no escopo dessa dissertação, os objetos de estudo e propostas que serão mostradas ao longo deste trabalho. Alguns conceitos relevantes para o restante do texto foram introduzidos e o robô AxeBot, estudo de caso deste trabalho, foi apresentado. Foi justificada a escolha do modelo de robô adotado com base no desafio proposto, como laboratório de testes, dentro do projeto Axebot.

# GERAÇÃO E CONTROLE DE TRAJETÓRIA DE ROBÔS MÓVEIS

## 3.1 INTRODUÇÃO

São apresentadas, neste capítulo, técnicas de geração de trajetórias por funções contínuas e por interpolação polinomial. Estas técnicas são utilizadas pelo gerador de trajetória proposto, na geração de trajetórias planas.

Uma introdução ao controle de trajetória de robôs móveis é feita, mostrando abordagens de controle baseadas na cinemática e na dinâmica e controladores baseados em lógica difusa. É mostrada a técnica de linearização entrada-saída por realimentação que é utilizada no controlador proposto, baseado na dinâmica, mostrado no capítulo 5.

## 3.2 GERAÇÃO DE TRAJETÓRIA

A trajetória de um robô é definida por um caminho sobre o qual o robô deve mover. Um caminho é uma sequência de posições ao longo do espaço de trabalho do robô ou ao longo do seu espaço de juntas. O espaço de trabalho é definido por posições em relação a um sistema cartesiano enquanto que o espaço de juntas tem como referência a posição angular dos atuadores. Segundo Khalil e Dombre (2002) as trajetórias de um robô podem ser classificadas como:

- Trajetória entre dois pontos com um caminho livre entre eles.

### 3.2 GERAÇÃO DE TRAJETÓRIA

- Trajetória entre dois pontos através de uma sequência de pontos intermediários chamados de *via points*.
- Trajetória entre dois pontos com restrições de caminho entre eles.
- Trajetória entre dois pontos com pontos intermediários entre eles e com restrições de caminho entre os pontos intermediários.

As figuras 3.1 e 3.2 mostram uma arquitetura de geração e controle de trajetória nos dois espaços de referência, sendo  $q(t)$  a trajetória no espaço de juntas,  $X(t)$  a

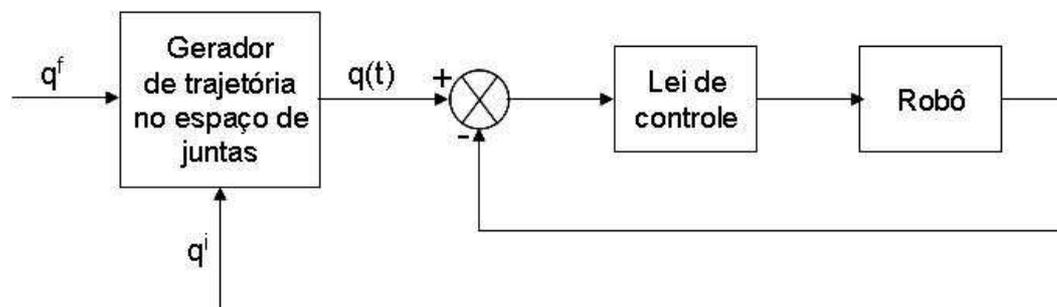


Figura 3.1. Geração e controle de trajetória no espaço de juntas.

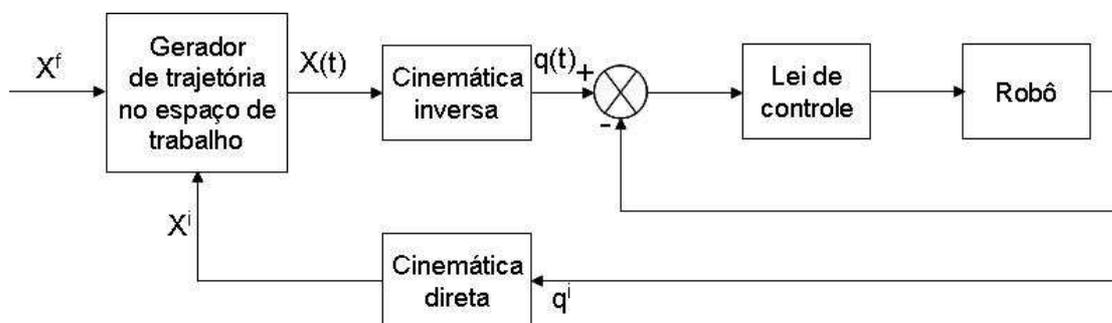


Figura 3.2. Geração e controle de trajetória no espaço de trabalho.

trajetória no espaço de trabalho,  $q^i$  e  $q^f$ , respectivamente, a posição inicial e final da trajetória no espaço de juntas e  $X^i$  e  $X^f$ , respectivamente a posição inicial e final da trajetória no espaço de trabalho.

É importante notar que a geração da trajetória no espaço de trabalho precisa ser convertida para o espaço de juntas caso a referência do controlador seja a posição ou a

velocidade angular do atuador. Esta conversão necessita do modelo cinemático inverso do robô, discutido no capítulo seguinte. Da mesma forma, o modelo cinemático direto é necessário para converter a saída do modelo da planta do espaço de juntas para o espaço de trabalho.

As trajetórias devem ser geradas evitando movimentos bruscos do robô. O movimento de um sistema mecânico robótico deve ser o mais suave possível. Mudanças bruscas na posição, velocidade ou aceleração podem acarretar sérios problemas (ANGELES, 2003). Alguns destes problemas são listados abaixo:

- Os movimentos bruscos requerem uma quantidade ilimitada de potência a ser executada, que o atuador pode não fornecer devido a sua limitação física.
- Uma mudança brusca da velocidade pode causar, no caso dos robôs com rodas, o deslizamento destas sobre o solo, prejudicando o desempenho do controlador de trajetória.
- Mudanças bruscas causam altas vibrações nos robôs podendo ser prejudiciais à sua parte mecânica.
- Uma mudança brusca na velocidade é equivalente a uma colisão do robô com um obstáculo, o que deve ser evitado.

Movimentos suaves podem ser implementados através de técnicas simples e de baixo custo computacional, ao mesmo tempo em que minimizam os problemas apontados acima.

Uma trajetória pode ser gerada por segmentos de reta, por uma função polinomial qualquer ou por interpolação de um grupo de funções polinomiais sobre os pontos discretos do caminho no espaço de trabalho ou no espaço de juntas. Algumas técnicas de interpolação polinomial utilizadas para geração de trajetórias são abordadas a seguir.

### 3.2.1 Interpolação polinomial

A interpolação é um método que permite fazer a reconstituição aproximada de uma função conhecendo apenas alguns de seus pontos (suas abscissas e respectivas ordenadas). A função interpoladora passa necessariamente sobre estes pontos. Uma interpolação é dita polinomial quando a função interpoladora é um polinômio.

Uma interpolação polinomial pode ter um polinômio interpolador de qualquer grau,  $n$ , em princípio. Quanto maior o grau do polinômio interpolador mais aproximada será a função interpoladora da função real, porém o custo para o cálculo dos seus parâmetros será elevado. As interpolações mais comumente usadas para a geração de trajetória de robôs são as de polinômios cúbicos e as de polinômios de grau cinco.

Uma trajetória entre dois pontos  $s^i$  e  $s^f$ , no espaço de trabalho ou no espaço de juntas, sendo  $s^i$  o ponto inicial e  $s^f$  o ponto final, pode ser determinada pela seguinte equação:

$$s(t) = s^i + r(\tau)D \quad \text{para } 0 \leq t \leq t_f \quad \text{e} \quad \tau = \frac{t}{t_f} \quad (3.1)$$

$$\dot{s}(t) = \dot{r}(\tau)D \quad (3.2)$$

Com  $D = s^f - s^i$  e  $r(\tau)$  uma função interpoladora normalizada.

É importante ressaltar que  $s(t)$  é o comportamento, no tempo, da trajetória em uma das coordenadas de um sistema cartesiano qualquer. Para uma trajetória plana são necessários dois polinômios interpoladores deste tipo.

As condições de contorno para a função interpoladora  $r(\tau)$  são dadas por:

$$\begin{cases} r(0) = 0 \\ r(1) = 1 \end{cases} \quad (3.3)$$

**3.2.1.1 Interpolação linear** Uma interpolação linear de dois polinômios  $s(t)$  é a maneira mais simples de geração de uma trajetória entre dois pontos.

Um polinômio linear pode ser representado pela seguinte expressão:

$$s(t) = s^i + \frac{t}{t_f} D \quad (3.4)$$

O problema neste tipo de interpolação é que a derivada nos pontos de interseção entre dois polinômios pode não existir, o que significa que as velocidades geradas podem ser descontínuas nestes pontos. Estas discontinuidades geram mudanças bruscas na velocidade resultando nos problemas anteriormente discutidos.

**3.2.1.2 Interpolação cúbica** Para resolver o problema da descontinuidade da velocidade é necessário que a velocidade inicial e final da trajetória sejam nulas, ou seja, que as seguintes condições de contorno sejam satisfeitas:

$$\begin{cases} \dot{s}(0) = 0 \\ \dot{s}(t_f) = 0 \end{cases} \quad (3.5)$$

Para que as condições em 3.5 e em 3.3 sejam atendidas, o polinômio  $r(\tau)$  tem que ser, no mínimo, de grau três, tendo a seguinte forma:

$$r(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 \quad (3.6)$$

Sabendo-se que:

$$\dot{s}(t) = D\dot{r}(\tau)\dot{\tau}(t) = D\frac{1}{t_f}\dot{r}(\tau) \quad (3.7)$$

$$\ddot{s}(t) = D\ddot{r}(\tau)\dot{\tau}^2(t) = D\frac{1}{t_f^2}\ddot{r}(\tau) \quad (3.8)$$

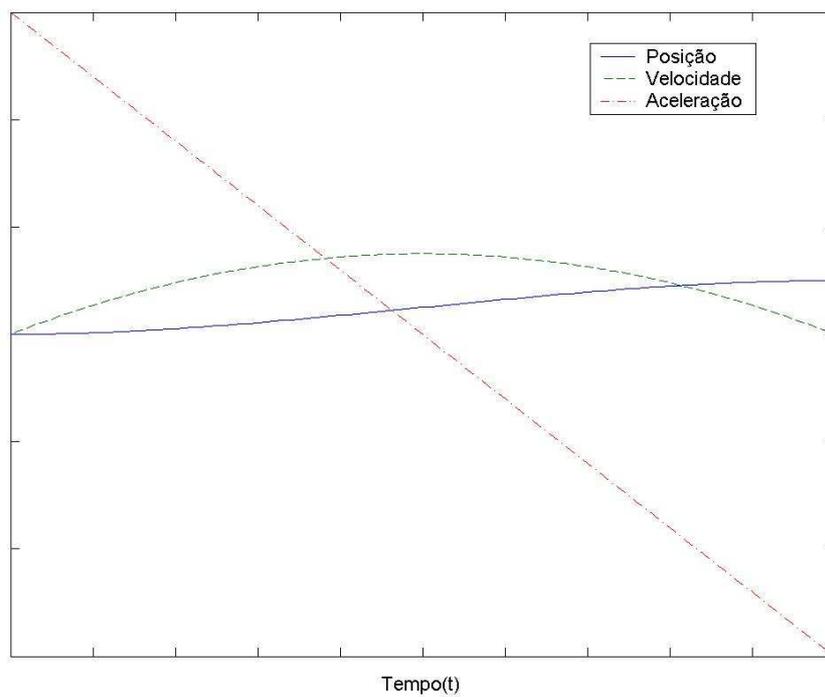
É fácil observar que os coeficientes do polinômio  $r(\tau)$  devem ter os seguintes valores:

$$\begin{cases} a_0 = a_1 = 0 \\ a_2 = 3 \\ a_3 = -2 \end{cases} \quad (3.9)$$

A expressão 3.6 pode também ser escrita na forma 3.1 com a seguinte função interpoladora:

$$r(\tau) = 3\left(\frac{t}{t_f}\right)^2 - 2\left(\frac{t}{t_f}\right)^3 \quad (3.10)$$

A figura 3.3 mostra as curvas de posição, velocidade e aceleração usando a técnica da interpolação cúbica.



**Figura 3.3.** Curvas de posição, velocidade e aceleração usando a técnica de interpolação cúbica.

**3.2.1.3 Interpolação por polinômios de grau cinco** A interpolação cúbica resolve o problema da descontinuidade da velocidade, porém não garante a continuidade da aceleração. Para que este problema seja solucionado é necessário adicionar duas novas condições de contorno à segunda derivada temporal do polinômio  $s(t)$ . Essas condições são apresentadas pelas expressões:

$$\begin{cases} \ddot{s}(0) = 0 \\ \ddot{s}(t_f) = 0 \end{cases} \quad (3.11)$$

Essas novas condições implicam em um aumento no grau do polinômio  $r(\tau)$  que passa agora a ser um polinômio de grau cinco, representado pela expressão:

$$r(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + a_4\tau^4 + a_5\tau^5 \quad (3.12)$$

Para que as condições em 3.3, 3.5 e 3.11 sejam atendidas os novos valores dos coeficientes de  $r(\tau)$  devem ser:

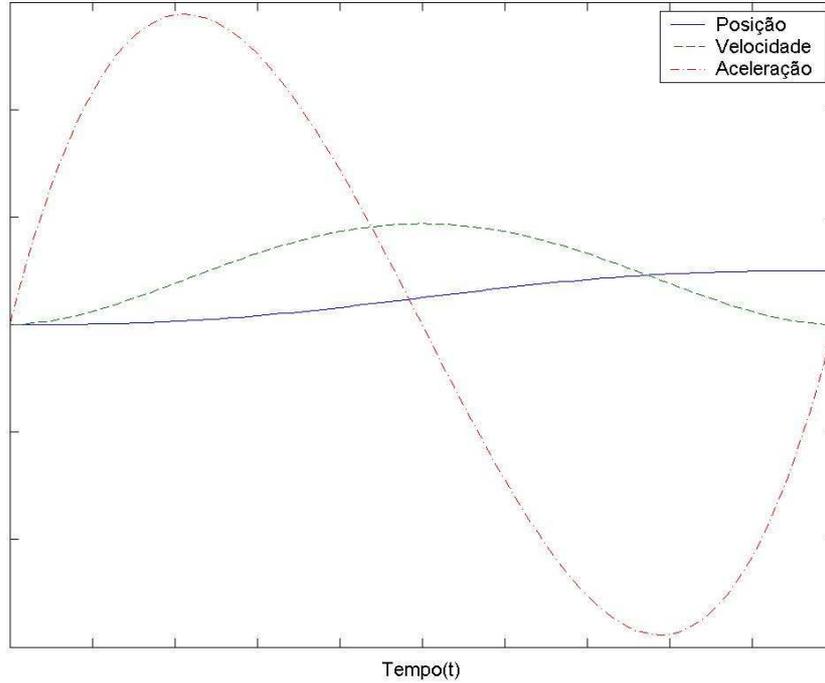
$$\begin{cases} a_0 = a_1 = a_2 = 0 \\ a_3 = 10 \\ a_4 = -15 \\ a_5 = 6 \end{cases} \quad (3.13)$$

Com essas novas condições, o polinômio interpolador pode ser agora representado por:

$$r(\tau) = 10\left(\frac{t}{t_f}\right)^3 - 15\left(\frac{t}{t_f}\right)^4 + 6\left(\frac{t}{t_f}\right)^5 \quad (3.14)$$

Este polinômio é também chamado de *interpolador 3-4-5* devido aos valores dos coeficientes de segundo e primeiro grau serem nulos.

A posição, velocidade e aceleração, usando a técnica de interpolação de grau cinco, são mostradas na figura 3.4.



**Figura 3.4.** Curvas de posição, velocidade e aceleração para um polinômio de grau cinco.

Note que as condições de suavidade impostas são respeitadas.

Pode-se determinar a evolução do movimento do robô sobre a trajetória se são dados os valores de  $t_f$  e de  $s^f$ , da equação 3.1. Se nenhuma condição extra for imposta, é possível executar o movimento desejado com o menor valor possível de  $t_f$ . No entanto, este valor não pode ser escolhido arbitrariamente devido às limitações impostas pelos atuadores do robô. Para calcular o valor máximo do módulo da velocidade e da aceleração basta achar, respectivamente, o valor de  $t$  para o qual a segunda e terceira derivadas temporais de  $r(\tau)$  sejam zero e substituir o seu valor na primeira e segunda derivada de  $s(t)$ , respectivamente. As expressões a seguir mostram esses valores.

$$|\dot{s}_{max}| = \frac{15}{8t_f}|D| \quad (3.15)$$

$$|\ddot{s}_{max}| = \frac{10}{\sqrt{3}t_f^2}|D| \quad (3.16)$$

### 3.2.2 Trajetória com *via points*

As técnicas de interpolação anteriormente discutidas não permitem especificar pontos intermediários entre a posição inicial e final da trajetória. Esta limitação, no entanto, pode ser contornada considerando a trajetória como sendo formada por pares de pontos entre um ponto inicial e final especificado, aplicando a cada par de pontos uma das técnicas de interpolação abordadas acima. O problema deste tipo de solução é que não existe uma evolução da velocidade e aceleração do robô durante o percurso da trajetória, devido ao fato de seus valores caírem a zero a cada novo par de pontos.

Uma solução para o problema apontado acima é determinar uma trajetória pela sua posição inicial e final e por algumas posições intermediárias chamadas de *via points*, interpolando todos estes pontos por um único polinômio a cada coordenada da trajetória. A introdução de *via points*, no entanto, acarreta um aumento no número de condições de contorno para a geração do polinômio interpolador.

Podem ser introduzidos, por exemplo, duas *via points* em uma trajetória polinomial desenvolvida para haver continuidade na sua primeira e segunda derivadas, atendendo as seguintes condições:

$$r(\tau_1) = r_1, \quad r(\tau_2) = r_2 \quad (3.17)$$

Onde  $\tau_1$ ,  $\tau_2$ ,  $r_1$  e  $r_2$  dependem do *via points* prescrito e do instante no qual são designados para ocorrer. Somando as condições em 3.17, 3.5, 3.11 e 3.3 existem agora oito condições de contorno a serem atendidas o que exige um polinômio interpolador de grau sete:

$$r(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + a_4\tau^4 + a_5\tau^5 + a_6\tau^6 + a_7\tau^7 \quad (3.18)$$

Para que as oito condições sejam atendidas é necessário que

$$\begin{cases} a_0 = a_1 = a_2 = 0 \\ a_3 + a_4 + a_5 + a_6 + a_7 = 1 \\ 7a_7 + 6a_6 + 5a_5 + 4a_4 + 3a_3 = 0 \\ 42a_7 + 30a_6 + 20a_5 + 12a_4 + 6a_3 = 0 \\ \tau_1^7 a_7 + \tau_1^6 a_6 + \tau_1^5 a_5 + \tau_1^4 a_4 + \tau_1^3 a_3 = r_1 \\ \tau_2^7 a_7 + \tau_2^6 a_6 + \tau_2^5 a_5 + \tau_2^4 a_4 + \tau_2^3 a_3 = r_2 \end{cases} \quad (3.19)$$

para cada conjunto de valores  $\tau_1$ ,  $\tau_2$ ,  $r_1$  e  $r_2$ , dados.

### 3.2.3 Geração de trajetória por funções cúbicas spline

A técnica de interpolação com *via points* se torna inviável quando o número de *via points* é grande, devido ao alto grau do polinômio interpolador e conseqüentemente ao alto custo computacional para o cálculo dos seus parâmetros. Uma alternativa para interpolar  $n$  pontos é o uso de uma técnica conhecida como funções *spline*.

Uma função *spline*, ou simplesmente *spline*, é composta por um conjunto de polinômios de grau  $m$ , sendo  $m$  baixo, com condições de continuidade nos pontos que ligam os polinômios da função e das suas derivadas de ordem até  $m - 1$ . O uso da técnica de *spline* para interpolar  $n$  pontos não torna a técnica mais complexa quando  $n$  é grande. Deste modo, *spline* oferece a possibilidade de interpolação sobre um conjunto virtualmente ilimitado de pontos sem produzir sérios problemas de resoluções numéricas.

Os polinômios que compõem uma função *spline* podem ser definidos com um grau qualquer. Nesta seção será mostrada a interpolação por funções *spline* com polinômios cúbicos chamadas de funções cúbicas *spline*. Uma função cúbica *spline*  $s(t)$  conectando  $N$  pontos  $P_k(t_k, s_k)$ , para  $k = 1, 2, \dots, N$ , é uma função definida por um conjunto de  $N - 1$  polinômios cúbicos unidos nos pontos  $P_k$  e contínuos nestes pontos. As condições

de continuidade de um *spline* cúbico é dada à função  $s(t)$  e às suas primeira e segunda derivadas, levando a ser portanto uma função de classe<sup>1</sup>  $C^2$ .

Sendo  $P_k(t_k, s_k)$  e  $P_{k+1}(t_{k+1}, s_{k+1})$  dois pontos consecutivos sobre a função  $s(t)$ , o  $k$ -ésimo polinômio cúbico  $s_k(t)$  entre esses pontos é dado por

$$s_k(t) = A_k(t - t_k)^3 + B_k(t - t_k)^2 + C_k(t - t_k) + D_k \quad (3.20)$$

para  $t_k \leq t \leq t_{k+1}$ . Assim, para o *spline*  $s(t)$ ,  $4(N-1)$  coeficientes  $A_k, B_k, C_k, D_k$  terão que ser determinados. Esses coeficientes serão calculados através dos valores dados de  $s_k(t_k)$  e da segunda derivada da função  $s_k$  nos pontos  $t_k$  e  $t_{k+1}$ , como explicado adiante.

Sendo

$$\dot{s}_k(t) = 3A_k(t - t_k)^2 + 2B_k(t - t_k) + C_k \quad (3.21)$$

$$\ddot{s}_k(t) = 6A_k(t - t_k) + 2B_k \quad (3.22)$$

a primeira e segunda derivada de  $s_k$ , respectivamente. Os coeficientes  $B_k, C_k$  e  $D_k$ , de acordo com as equações acima, podem ser determinados pelas expressões 3.23, 3.24 e 3.25.

$$B_k = \frac{1}{2}\ddot{s}_k(t_k) \quad (3.23)$$

$$C_k = \dot{s}_k(t_k) \quad (3.24)$$

$$D_k = s_k(t_k) \quad (3.25)$$

Das relações acima, encontram-se as expressões para  $B_k$  e  $D_k$  em termos de  $\ddot{s}_k(t_k)$  e  $s_k(t_k)$ , porém a expressão para  $C_k$  é dada em termo de  $\dot{s}_k(t_k)$ . Pode-se encontrar uma expressão para  $C_k$  e  $A_k$  em termos da primeira e segunda derivadas de  $s_k$  nos pontos  $t_k$  e  $t_{k+1}$ , impondo as condições de continuidade em  $s_k, \dot{s}_k$  e  $\ddot{s}_k$  em  $t_{k+1}$ , sendo

---

<sup>1</sup>Uma função é dita de classe  $C^n$  se esta for diferenciável  $n$  vezes.

$k = 1, 2, \dots, N - 1$ . Estas condições são mostradas em 3.26

$$\begin{cases} s_k(t_{k+1}) = s_{k+1}(t_{k+1}) \\ \dot{s}_k(t_{k+1}) = \dot{s}_{k+1}(t_{k+1}) \\ \ddot{s}_k(t_{k+1}) = \ddot{s}_{k+1}(t_{k+1}) \end{cases} \quad (3.26)$$

Resolvendo a equação 3.22 para  $\ddot{s}_k(t_{k+1})$  e de acordo com a equação 3.23, obtém-se:

$$6A_k \Delta t_k + 2B_k = 2B_{k+1} \quad (3.27)$$

Sendo  $\Delta t_k = t_{k+1} - t_k$ .

Achando os valores de  $B_k$  e  $B_{k+1}$  através da equação 3.23 e substituindo-os na equação 3.27,  $A_k$  pode ser expresso pela relação abaixo:

$$A_k = \frac{1}{6\Delta t_k} (\ddot{s}_k(t_{k+1}) - \ddot{s}_k(t_k)) \quad (3.28)$$

Resolvendo a equação 3.20 para  $s_k(t_{k+1})$  obtém-se:

$$A_k(\Delta t_k)^3 + B_k(\Delta t_k)^2 + C_k \Delta t_k + D_k = s_k(t_{k+1}) \quad (3.29)$$

Isolando  $C_k$  da equação acima com os valores de  $A_k$ ,  $B_k$  e  $D_k$  das respectivas equações 3.28, 3.23 e 3.25, encontra-se a relação expressa pela equação

$$C_k = \frac{\Delta s_k}{\Delta t_k} - \frac{1}{6} \Delta t_k (\ddot{s}_k(t_{k+1}) + 2\ddot{s}_k(t_k)) \quad (3.30)$$

onde  $\Delta s_k = s_k(t_{k+1}) - s_k(t_k)$ .

De acordo com a condição de continuidade da primeira derivada de  $s_k$  em  $t_{k+1}$ , expressa pela segunda equação de 3.26 e de acordo com a equação 3.21, a seguinte expressão é obtida:

$$3A_k(\Delta t_k)^2 + 2B_k \Delta t_k + C_k = C_{k+1} \quad (3.31)$$

ou deslocando o polinômio acima para o instante anterior,

$$3A_{k-1}(\Delta t_{k-1})^2 + 2B_{k-1}\Delta t_{k-1} + C_{k-1} = C_k \quad (3.32)$$

Substituindo as expressões 3.28, 3.23 e 3.30 na expressão acima, um sistema linear indeterminado com  $N - 2$  equações para  $N$  variáveis  $\ddot{s}_k$  é obtido. Este sistema é mostrado em 3.33

$$\Delta t_k \ddot{s}_k(t_{k+1}) + 2(\Delta t_{k-1} + \Delta t_k) \ddot{s}_k(t_k) + \Delta t_{k-1} \ddot{s}_{k-1} = 6\left(\frac{\Delta s_k}{\Delta t_k} - \frac{\Delta s_{k-1}}{\Delta t_{k-1}}\right) \quad (3.33)$$

para  $k = 2, \dots, N - 1$

O sistema acima pode ser representado na forma matricial,

$$\mathbf{A}\ddot{\mathbf{s}} = 6\mathbf{C}\mathbf{s} \quad (3.34)$$

sendo  $\mathbf{s} = [s(t_1), \dots, s(t_N)]^T$ ,  $\ddot{\mathbf{s}} = [\ddot{s}(t_1), \dots, \ddot{s}(t_N)]^T$  e  $\mathbf{A}$  e  $\mathbf{C}$  matrizes  $(N - 2) \times N$ , representadas em 3.35 e 3.36, respectivamente:

$$\mathbf{A} = \begin{bmatrix} \alpha_1 & 2\alpha_{1,2} & \alpha_2 & 0 & \dots & 0 & 0 \\ 0 & \alpha_2 & 2\alpha_{2,3} & \alpha_3 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \alpha_{N-3} & 2\alpha_{N-3,N-2} & \alpha_{N-2} & 0 \\ 0 & 0 & 0 & \dots & \alpha_{N-2} & 2\alpha_{N-2,N-1} & \alpha_{N-1} \end{bmatrix} \quad (3.35)$$

$$\mathbf{C} = \begin{bmatrix} \beta_1 & -\beta_{1,2} & \beta_2 & 0 & \dots & 0 & 0 \\ 0 & \beta_2 & -\beta_{2,3} & \beta_3 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \beta_{N-3} & -\beta_{N-3,N-2} & \beta_{N-2} & 0 \\ 0 & 0 & 0 & \dots & \beta_{N-2} & -\beta_{N-2,N-1} & \beta_{N-1} \end{bmatrix} \quad (3.36)$$

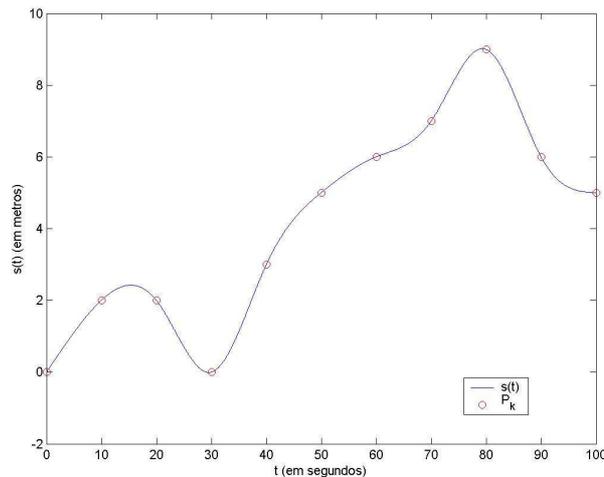
### 3.2 GERAÇÃO DE TRAJETÓRIA

sendo  $\alpha_k = \Delta t_k$ ,  $\alpha_{i,j} = \alpha_i + \alpha_j$ ,  $\beta_k = \frac{1}{\alpha_k}$  e  $\beta_{i,j} = \beta_i + \beta_j$ , para  $i, j, k = 1, \dots, N - 1$ .

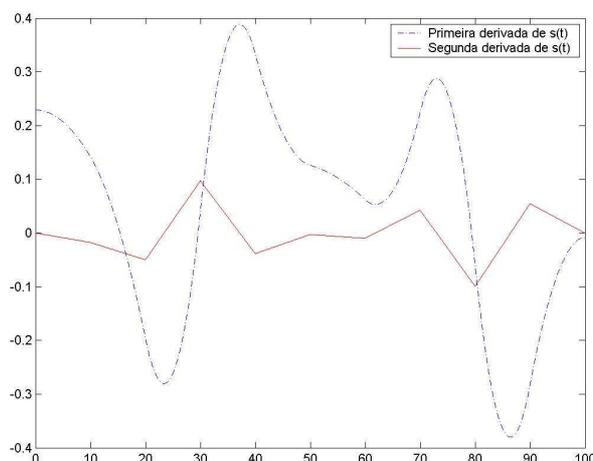
Para que o sistema 3.33 seja determinado é necessário acrescentar duas novas equações a ele, de acordo com algumas condições. Essas condições dependem das restrições que se deseja impor à função interpoladora, representando o tipo de *spline* a ser usado. Por exemplo, se  $\ddot{s}_1(t_1)$  e  $\ddot{s}_{N-1}(t_N)$  forem definidas como zero, a *spline* obtida será chamada de *spline* cúbica natural. Este tipo de *spline* define que as acelerações nos pontos extremos da trajetória  $s(t)$  serão nulas. Na *spline* cúbica natural a matriz  $\mathbf{A}$  em 3.34 passa ser uma matriz  $(N - 2) \times (N - 2)$  tornando, portanto, o sistema 3.34 determinado. A nova matriz  $\mathbf{A}$  é mostrada abaixo:

$$\mathbf{A} = \begin{bmatrix} 2\alpha_{1,2} & \alpha_2 & 0 & \dots & 0 \\ \alpha_2 & 2\alpha_{2,3} & \alpha_3 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \alpha_{N-3} & 2\alpha_{N-3,N-2} & \alpha_{N-2} \\ 0 & 0 & \dots & \alpha_{N-2} & 2\alpha_{N-2,N-1} \end{bmatrix} \quad (3.37)$$

As figuras 3.5 e 3.6 mostram uma função *spline* cúbica natural interpolando onze pontos e suas derivadas primeira e segunda, respectivamente.



**Figura 3.5.** Função *spline* cúbica natural  $s(t)$  interpolando onze pontos  $P_k$ .



**Figura 3.6.** Derivadas primeira e segunda da função spline cúbica natural  $s(t)$ .

### 3.3 CONTROLE DE TRAJETÓRIA

Grande parcela dos trabalhos, na área de veículos autônomos, é dedicada ao estudo e desenvolvimento de estratégias de controle de trajetória. O objetivo do controle de trajetória é posicionar o robô sobre um caminho previamente definido pelo gerador de trajetória. Este caminho, como mencionado anteriormente, deve ser desenvolvido por polinômios contínuos no tempo, de modo a evitar mudanças bruscas de velocidade e aceleração do robô. O controlador de trajetória está intimamente ligado aos modelos cinemático e dinâmico de robô.

O projeto de leis de controle para robôs móveis tem sido desenvolvido com base em três metodologias (WIT et al., 1993):

*Estabilização em um ponto:* A metodologia de estabilização em um ponto é adotada quando se deseja encontrar uma lei de controle de velocidade no qual o  $\lim_{t \rightarrow \infty} (\xi - \xi_r) = 0$ , sendo  $\xi = (x, y, \theta)$  a postura do robô e  $\xi_r = (x_r, y_r, \theta_r)$  a postura de referência a ser seguida pelo robô.

*Rastreamento de trajetória:* Dadas as coordenadas de postura  $\xi = (x, y, \theta)$  e de referência  $\xi_r = (x_r, y_r, \theta_r)$  do robô, deve-se encontrar uma lei de controle para as

velocidades angular e linear de modo que o  $\lim_{t \rightarrow \infty} (x(t) - x_r(t)) = 0$ ,  $\lim_{t \rightarrow \infty} (y(t) - y_r(t)) = 0$  e o  $\lim_{t \rightarrow \infty} (\theta(t) - \theta_r(t)) = 0$  (OUBBATI, 2006).

*Seguimento de trajetória:* Dado um caminho no plano  $\xi$ , e assumindo que  $e_\theta$  e  $e_{xy}$  são o erro de orientação e a distância entre um ponto de referência no robô e o plano  $\xi$ , respectivamente, objetiva-se encontrar uma lei de controle de velocidade a qual o  $\lim_{t \rightarrow \infty} |e_\theta| = 0$  e o  $\lim_{t \rightarrow \infty} |e_{xy}| = 0$  (OUBBATI, 2006).

### 3.3.1 Tipos de controles

Os tipos de controle de trajetória de robôs móveis podem ser resumidos em três categorias: Controle baseado na cinemática, controle baseado na dinâmica e um controle em cascata da cinemática e da dinâmica do robô.

**3.3.1.1 Controle baseado na cinemática** Este tipo de controle leva em consideração apenas o modelo cinemático do robô, considerando que o rastreamento de velocidade é perfeito. O principal objetivo é encontrar entradas de controle de velocidade ( $\dot{\xi}_r$ ) que estabilizem o controle em malha fechada da cinemática, dada uma postura de referência do robô ( $\xi_r$ ). O grande problema é que, na prática, não se tem um rastreamento de velocidade perfeito, devido a certas características físicas do robô, como massa e momento de inércia e a certas características mecânicas e elétricas dos seus atuadores. Este tipo de controle funciona bem em baixas velocidades e com robôs pequenos (RAFFO, 2005). Os modelos cinemáticos descrevem o comportamento do veículo em função da velocidade e orientação das rodas.

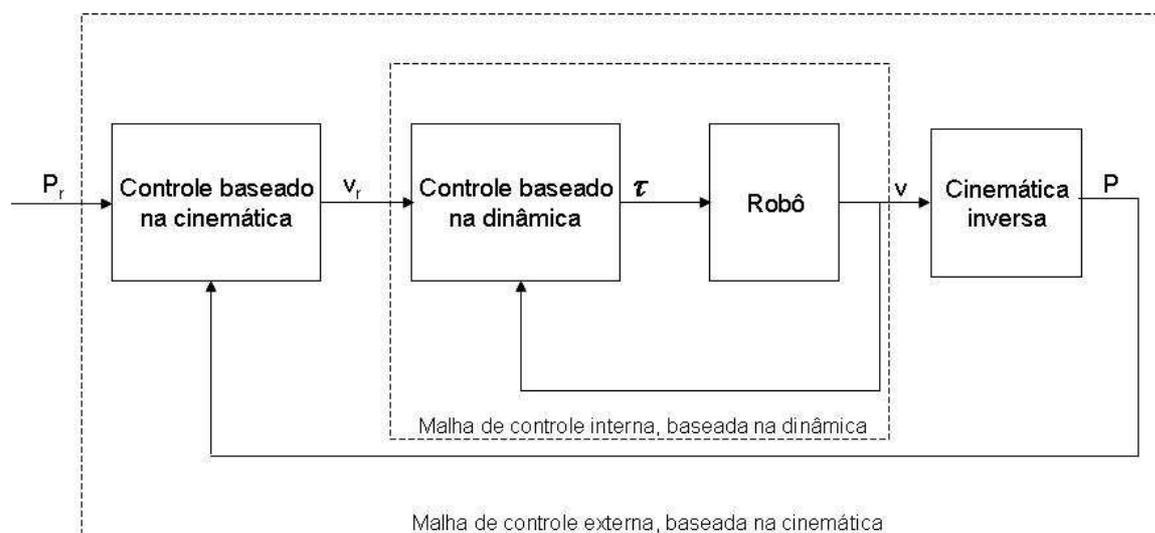
Podem ser citados alguns trabalhos dentro deste contexto: Borenstein (1995) aborda o controle baseado na cinemática de veículos não-holonômicos. Kanayama *et al.*(1990), Fierro e Lewis (1995), Fierro e Lewis (1998) e Fukao *et al.*(2000) mostram o controle baseado na cinemática, de robôs diferenciais.

**3.3.1.2 Controle baseado na dinâmica** As características da dinâmica do robô não podem ser negligenciadas quando este se encontra em alta velocidade e aceleração e quando o robô é de médio a grande porte. O controle baseado no modelo cinemático para este caso não é adequado para representar com exatidão o comportamento do robô.

O controle baseado na dinâmica tem como objetivo fazer o robô rastrear uma velocidade de referência, em função dos torques ( $\tau$ ) aplicados aos seus atuadores. Os parâmetros do modelo dinâmico, porém, podem mudar com o tempo e seus valores podem ser difíceis de ser mensurados. Não linearidades como atrito de superfície, consumo de energia, entre outros, degradam o desempenho de controladores deste tipo. O controle baseado na dinâmica de diversas classes de robôs móveis pode ser encontrado em Reis (2006) e Lages (1998). Oubbati (2006) utiliza um controlador adaptativo, baseado na dinâmica, para um robô omnidirecional. Franco e Costa (2006), usam um controlador por realimentação de estados, baseado na dinâmica de um robô omnidirecional. Outros trabalhos com controladores deste tipo, para robôs omnidirecionais, podem ser encontrados em Watanabe (1998) e Watanabe *et al.*(1998)

**3.3.1.3 Controle em cascata** Os controles baseados apenas na dinâmica não fecham a malha de controle de postura, o que pode gerar erros em regime estacionário, fazendo o robô desviar da postura de referência. Uma solução para este problema é criar uma estratégia de controle baseada em controladores em cascata, tendo duas malhas de controle. Uma malha interna, responsável pelo rastreamento de velocidade, baseada no controle dinâmico, e uma malha externa, baseada no controle cinemático, para rastrear a postura de referência, gerando velocidade de referência para a malha de controle interna. A figura 3.7 mostra o esquema deste tipo de controle.

Controladores de trajetória em cascata, de robôs móveis, podem ser encontrados em Gomes (2006) e Liu *et al.* (2003)



**Figura 3.7.** Diagrama de blocos do controle cinemático e dinâmico em cascata.

### 3.3.2 Controle linearizante

O modelo dinâmico do robô é não-linear, o que dificulta a aplicação de leis de controles convencionais, como o PID, no projeto de controles baseados no modelo dinâmico. Esta seção apresenta alguns conceitos básicos de leis controle usando linearização por realimentação de estado, que serão utilizados no projeto do controle baseado na dinâmica do robô.

Existem basicamente duas técnicas de linearização. Uma delas consiste em expandir o sistema em série de Taylor em torno de um ponto, desprezando os termos de ordem superior. A outra técnica consiste em cancelar os termos de ordem superior, através de uma linearização por realimentação do modelo. A vantagem desta última técnica é que, a princípio, obtém-se uma descrição linear exata para o sistema não linear, ao invés de uma aproximação linear para o sistema não linear. Por outro lado, a principal desvantagem da linearização por realimentação reside no fato de que se os parâmetros e o estado do sistema não forem bem conhecidos, o cancelamento dos termos de ordem superior não será exato e portanto, a descrição linear obtida não corresponderá ao sistema original. Conseqüentemente, a lei de controle linear projetada para o sistema linearizado, poderá perder eficiência e o sistema poderá ter o seu

desempenho alterado ou até mesmo tornar-se instável (LAGES, 1998).

**3.3.2.1 Linearização Entrada-Saída por realimentação de estado** A técnica de linearização por realimentação de estados (ISIDORI, 1989) mostra-se útil para controle de robôs móveis. Uma das técnicas de linearização por realimentação é a linearização Entrada-Saída. Esta técnica será abordada a seguir.

Dado um sistema multivariável com múltiplas entradas e múltiplas saídas (MIMO) e com o mesmo número de entradas e saídas, da forma

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}_1(\mathbf{x})u_1 + \dots + \mathbf{g}_m(\mathbf{x})u_m \\ y_1 &= h_1(\mathbf{x}) \\ &\vdots \\ y_m &= h_m(\mathbf{x})\end{aligned}\tag{3.38}$$

onde  $\mathbf{x}$  é o vetor de estado,  $u_i (i = 1, \dots, m)$  são as entradas de controle,  $y_j (j = 1, \dots, m)$  são as saídas,  $\mathbf{f}$  e  $\mathbf{g}_i$ , são campos de vetores suaves, e  $h_j$  são funções escalares suaves, pode-se agrupar as entradas de controle  $u_i$  dentro de um vetor  $\mathbf{u}$ , o correspondente vetor  $\mathbf{g}_i$  dentro de uma matriz  $\mathbf{G}$ , e as saídas  $y_j$  dentro de um vetor  $\mathbf{y}$  de modo a que o sistema passa a ser escrito como:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\tag{3.39}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x})\tag{3.40}$$

Para se obter a linearização entrada-saída do sistema MIMO 3.38, deve-se derivar  $y_j$  no tempo até que a entrada do sistema apareça na equação resultante. Derivando a saída  $y_j$  no tempo, obtém-se

$$\dot{y}_j = L_{\mathbf{f}}h_j + \sum_{i=1}^m (L_{\mathbf{g}_i}h_j)u_i\tag{3.41}$$

sendo  $L_{\mathbf{f}}h_j$  a derivada de Lie da saída  $h_j$  correspondente ao termo de  $\mathbf{f}(\mathbf{x})$  e  $L_{\mathbf{g}_i}h_j$  a derivada de Lie da saída  $h_j$  correspondente aos termos de  $\mathbf{g}_i(\mathbf{x})$ , representadas por

$$L_{\mathbf{f}}h_j(\mathbf{x}) = \sum_{i=1}^m \frac{\partial h_j(\mathbf{x})}{\partial \mathbf{x}_i} \dot{\mathbf{x}}_{i_f} \quad (3.42)$$

$$L_{\mathbf{g}_i}h_j(\mathbf{x}) = \sum_{i=1}^m \frac{\partial h_j(\mathbf{x})}{\partial \mathbf{x}_i} \dot{\mathbf{x}}_{i_g} \quad (3.43)$$

onde  $\dot{\mathbf{x}}_{i_f}$  e  $\dot{\mathbf{x}}_{i_g}$  representam, respectivamente, as partes relativas a  $\mathbf{f}(\mathbf{x})$  e  $\mathbf{g}(\mathbf{x})$  da  $i$ -ésima equação de estado do sistema representado por 3.38.

Se  $L_{\mathbf{g}_i}h_j(\mathbf{x}) = 0$  para todo  $i$ , as entradas não aparecem e a saída tem que ser diferenciada novamente. Considerando que  $r_j$  é o menor inteiro tal que no mínimo uma das entradas apareçam em  $y_j^{(r_j)}$ , então

$$y_j^{(r_j)} = L_{\mathbf{f}}^{(r_j)}h_j + \sum_{i=1}^m L_{\mathbf{g}_i}L_{\mathbf{f}}^{r_j-1}h_j u_i \quad (3.44)$$

com  $L_{\mathbf{g}_i}L_{\mathbf{f}}^{r_j-1}h_j(\mathbf{x}) \neq 0$  para no mínimo um  $i$ , para todo o  $\mathbf{x}$  pertencente à região onde a linearização é válida.

A equação 3.44 pode ser reescrita na forma

$$\begin{bmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} L_{\mathbf{f}}^{r_1}h_1(\mathbf{x}) \\ \vdots \\ L_{\mathbf{f}}^{r_m}h_m(\mathbf{x}) \end{bmatrix} + \mathbf{E}(\mathbf{x}) \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \quad (3.45)$$

onde  $\mathbf{E}(\mathbf{x})$  é uma matriz  $m \times m$  definida como

$$\mathbf{E}(\mathbf{x}) = \begin{bmatrix} L_{\mathbf{g}_1}L_{\mathbf{f}}^{r_1-1}h_1 & \dots & L_{\mathbf{g}_m}L_{\mathbf{f}}^{r_1-1}h_1 \\ \vdots & \ddots & \vdots \\ L_{\mathbf{g}_1}L_{\mathbf{f}}^{r_m-1}h_m & \dots & L_{\mathbf{g}_m}L_{\mathbf{f}}^{r_m-1}h_m \end{bmatrix} \quad (3.46)$$

A matriz  $\mathbf{E}(\mathbf{x})$  é chamada de matriz de desacoplamento do sistema MIMO. Se esta

matriz é não-singular, a transformação da entrada

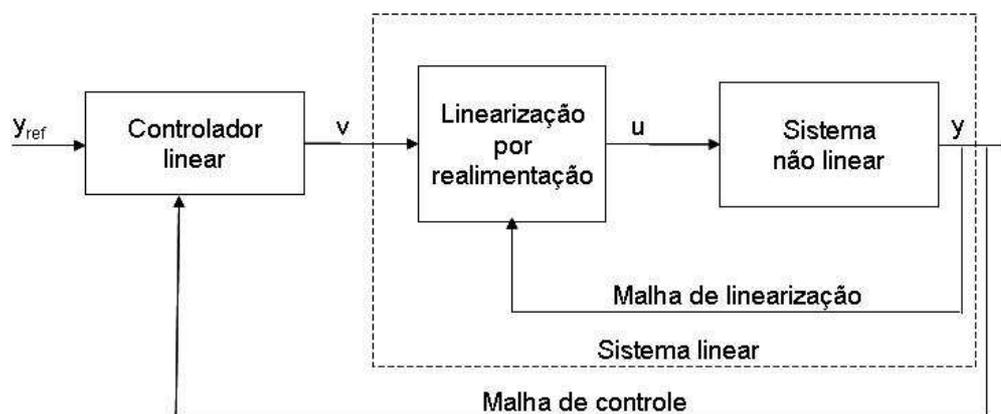
$$\begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} = -\mathbf{E}^{-1}(\mathbf{x}) \begin{bmatrix} L_{\mathbf{f}}^{r_1} h_1(\mathbf{x}) \\ \vdots \\ L_{\mathbf{f}}^{r_m} h_m(\mathbf{x}) \end{bmatrix} + \mathbf{E}^{-1}(\mathbf{x}) \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \quad (3.47)$$

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

$$\begin{bmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \quad (3.48)$$

Note que a relação entrada-saída acima é desacoplada e linear. Uma lei de controle na forma 3.47 é chamada de lei de controle desacoplada ou lei de controle não-interativa. Como resultado do desacoplamento, pode ser usado um sistema de uma entrada e uma saída(SISO) para cada canal  $y_j$ - $v_j$  da dinâmica desacoplada acima, de modo a obter controladores estáveis (SLOTINE; LI, 1991).

A figura 3.8, de acordo com Lages (1998), mostra um diagrama de blocos do controle linearizante.



**Figura 3.8.** Diagrama de blocos do controle linearizante.

**3.3.2.2 Sistema na forma companheira** Um sistema é dito estar na forma companheira se sua dinâmica puder ser representada por uma equação do tipo

$$\mathbf{x}^{(n)} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \quad (3.49)$$

Se  $\mathbf{G}(\mathbf{x})$  admitir inversa, a linearização por realimentação estática de estado pode ser obtida através de uma lei de controle baseada na inversa do modelo dinâmico, sob a forma

$$\mathbf{u} = -\mathbf{G}^{-1}(\mathbf{x})\mathbf{f}(\mathbf{x}) + \mathbf{G}^{-1}(\mathbf{x})\mathbf{v} \quad (3.50)$$

Substituindo 3.50 em 3.49, obtém-se uma simples relação entrada-saída expressa por:

$$\mathbf{x}^{(n)} = \mathbf{v} \quad (3.51)$$

Fazendo

$$\mathbf{v} = \mathbf{x}_d^{(n)} - k_1\mathbf{e} - k_2\dot{\mathbf{e}} - \dots - k_{n-1}\mathbf{e}^{(n-1)} \quad (3.52)$$

com  $k_i$  sendo constantes,  $\mathbf{x}_d$  as saídas desejadas e  $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$ . Se as constantes  $k_i$  forem positivas o  $\lim_{t \rightarrow \infty} |\mathbf{e}| = 0$  sendo, portanto, o erro assintoticamente estável<sup>2</sup>.

### 3.3.3 Controladores difusos

**3.3.3.1 Introdução à lógica difusa** A lógica difusa pode ser vista como uma linguagem que permite traduzir estruturas sofisticadas da linguagem natural dentro de um formalismo matemático (MCNEILL; THRO, 1994).

Segundo Sandri e Correa (1999), as teorias mais conhecidas para tratar da imprecisão e da incerteza são, respectivamente, a teoria dos conjuntos e a teoria de probabilidades. Estas teorias, embora muito úteis, nem sempre conseguem captar a

---

<sup>2</sup>Uma outra maneira de analisar a estabilidade seria calculando os valores das raízes do polinômio característico. Nota-se que as raízes são negativas quando os valores de  $k_i$  são positivos, ou seja, os pólos são localizados no semi-plano esquerdo do plano complexo.

riqueza da informação fornecida por seres humanos. A lógica difusa trabalha com a imprecisão e as relações complexas entre as variáveis envolvidas no processo. Estas variáveis podem ser expressas em termos lingüísticos (TSOUKALAS; UHRIG, 1996).

Os termos lingüístico de cada variável, também conhecidos como conjuntos difusos são representados por funções de pertinência, que definem o grau de possibilidade dos valores analíticos das variáveis dentro no seu termo lingüístico correspondente. As funções de pertinência são comumente triangulares, trapezoidais ou gaussianas. A figura 3.9 mostra as funções de pertinência de uma variável  $X$ .

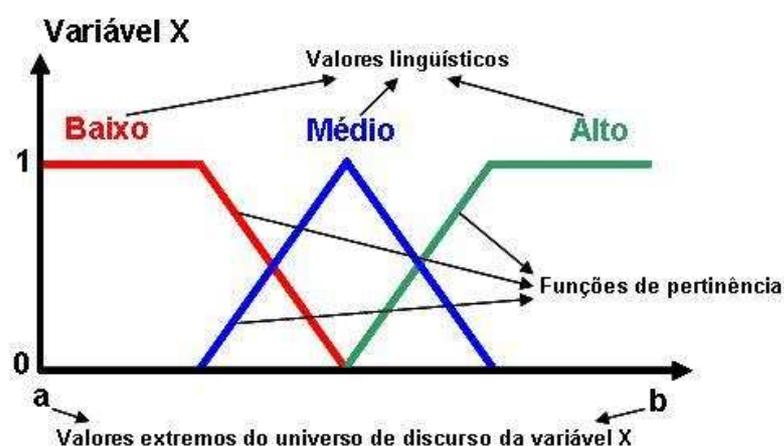


Figura 3.9. Funções de pertinência de uma variável  $X$ .

O processamento de um sistema difuso é baseado em um conjunto de regras do tipo *Se <antecedente> Então <consequente>* definindo a base de conhecimento do modelo difuso proposto. No modelo difuso Mamdani as regras têm o seguinte formato

$$\text{Regra } n: \text{ Se } X \text{ é } A_i \text{ e } Y \text{ é } B_j \text{ então } Z \text{ é } C_k$$

onde  $X$  e  $Y$  são as variáveis de entrada,  $Z$  a variável de saída,  $A_i$  e  $B_j$  os valores lingüísticos das variáveis de entrada e  $C_k$  os valores lingüísticos da variável de saída.  $A_i$ ,  $B_j$  e  $C_k$  são as funções de pertinência da sua respectiva variável.

Cada regra  $n$  pode contribuir ou não para o resultado do valor de saída de um sistema difuso. Se o valor da antecedente da regra  $n$  não for zero esta participa na

contribuição, sendo denominada de regra ativa. O cálculo da antecedente da regra  $n$  pode ser feito por meio de um função *min* como mostrado na expressão

$$\mu_p = \min(\mu_{A_i}(x), \mu_{B_j}(y)) \quad (3.53)$$

sendo  $x$  o valor da variável  $X$ ,  $y$  o valor da variável  $Y$ ,  $\mu_{A_i}(x)$  o valor da função de pertinência  $A_i$  no ponto  $x$ ,  $\mu_{B_j}(y)$  o valor da função de pertinência  $B_j$  no ponto  $y$  e  $\mu_p$  o valor da antecedente da regra  $p$

Cada  $\mu_p$  limitará a função de pertinência  $C_k$  da regra  $n$  ativa. A agregação das funções de pertinência  $C_k$  limitadas por  $\mu_p$  formam uma função  $D$ . O valor de saída de um sistema difuso é calculado por um processo denominado de defuzificação. Um dos métodos de defuzificação, bastante usado no modelo Mamdani é achar o centro de área da função agregada  $D$ . Este método é denominado de *COA* (*Central of Area*) e o seu cálculo é mostrado na expressão

$$u = \frac{\sum_{k=1}^n \mu_D(x_k) x_k}{\sum_{k=1}^n \mu_D(x_k)} \quad (3.54)$$

sendo  $\mu_D(x_k)$  o valor da função  $D$  no ponto  $x_k$ ,  $n$  o número de pontos escolhidos dentro do universo de discurso de  $D$  e  $u$  o valor de saída do sistema difuso.

A lógica difusa vem sendo utilizada com bastante eficiência na área de controle, substituindo ou auxiliando os controladores convencionais (SUGISAKA, 2003; SCHNITMAN; YONEYAMA, 2000; WATANABE; IZUMI; HAN, 1998; WATANABE, 1998; KODAGODA; WIJESOMA; TEOH, 2002). As técnicas de controle difuso originaram-se com as pesquisas e projetos de Mamdani (1976). Estes controladores utilizam regras lógicas no algoritmo de controle, com a intenção de descrever numa rotina a experiência humana (ZADEH, 1965).

Geralmente, os controladores difusos são usados em sistemas não lineares, sendo capazes de superar perturbações e plantas com níveis de ruídos. Além disso, mesmo

em sistemas onde a incerteza se faz presente de maneira intrínseca, agregam uma robustez característica. No entanto, provar determinadas propriedades de robustez é uma tarefa difícil neste tipo de abordagem (SANDRI; CORREA, 1999). As seções 3.3.3.2 e 3.3.3.3 abordam os controladores difusos do tipo PI(Proporcional Integral) e PD(Proporcional Derivativo)

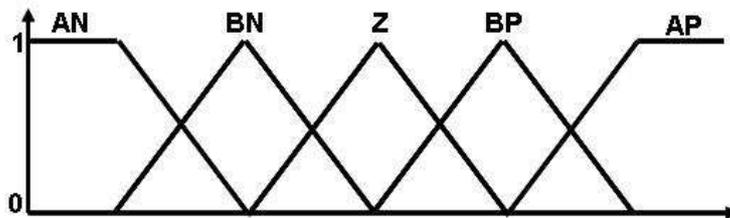
**3.3.3.2 Controlador difuso PI(Proporcional Integral)** Um controlador PI digital é representado pela equação

$$u(k) = K_p e(k) + K_i \sum_{i=0}^k e(i) \quad (3.55)$$

onde  $k$  é um instante de tempo discreto qualquer,  $e(k)$  é o erro no instante  $k$ ,  $K_p$  a constante de proporção e  $K_i$  a constante de integração. Subtraindo a equação 3.55 pela mesma equação no instante  $k - 1$  obtém-se

$$\Delta u(k) = K_p \Delta e(k) + K_i e(k) \quad (3.56)$$

Em um controlador difuso PI,  $\Delta e(k)$  e  $e(k)$  são as variáveis linguísticas de entrada, enquanto que  $\Delta u(k)$  é a variável linguística de saída. Cada variável pode assumir, por exemplo, os seguintes valores linguísticos: AN - Alto Negativo; BN - Baixo Negativo; Z - Zero; BP - Baixo Positivo; AP - Alto Positivo, representados pelas funções de pertinência mostradas na figura 3.10



**Figura 3.10.** Exemplos de funções de pertinência das variáveis de um controlador difuso PI.

Um exemplo de base de regras de um controlador difuso PI, com as funções de pertinência definidas pela figura 3.10, pode ser resumida na tabela 3.1

		$\Delta e(k)$				
		AN	BN	Z	BP	AP
$e(k)$	AN	AN	AN	AN	Z	BP
	BN	AN	AN	BN	Z	BP
	Z	AN	BN	Z	BP	AP
	BP	BN	Z	BP	AP	AP
	AP	BN	Z	AP	AP	AP

Tabela 3.1. Base de regras do controlador difuso PI

Um cruzamento de uma coluna com uma linha da tabela 3.1 define uma regra, como por exemplo, o cruzamento da linha 1 com a coluna 1 representa a regra **Se**  $e(k)$  é AN e  $\Delta e(k)$  é AN **então**  $\Delta u(k)$  é AN

O diagrama de blocos de um controlador difuso PI é mostrado na figura 3.11

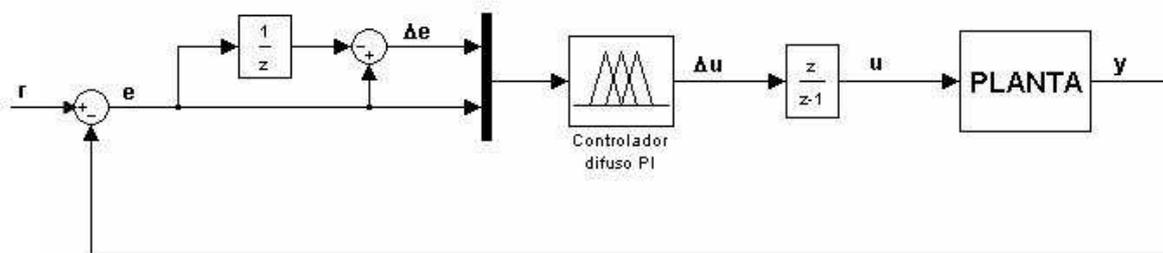


Figura 3.11. Diagrama de blocos de um controlador difuso PI.

**3.3.3.3 Controlador difuso PD(Proporcional Derivativo)** Um controlador PD digital é representado pela equação

$$u(k) = K_p e(k) + K_d \Delta e(k) \tag{3.57}$$

Em um controlador difuso PD,  $e(k)$  e  $\Delta e(k)$  são as entradas do sistema difuso, enquanto  $u(k)$  é a saída. É fácil notar que a diferença entre este controlador e o

### 3.4 CONCLUSÃO

---

controlador PI está na variável de saída. A saída de um controlador PD é o valor do sinal de controle, enquanto que a saída de um controlador PI é variação deste valor. É mais comum usar controladores difusos PI, pois é mais intuitivo trabalhar com a variação do sinal de controle.

O controle linearizante abordado na seção anterior depende do modelo dinâmico da planta, o que pode estar sujeito a ruídos e a parâmetros incertos. Controladores difusos PI podem ser utilizados como compensadores de ruídos e das incertezas dos parâmetros do modelo dinâmico do robô.

### 3.4 CONCLUSÃO

Foram apresentados neste capítulo técnicas de geração de trajetórias e uma introdução ao controle de trajetória de robôs móveis, mostrando os conceitos de controladores baseados na cinemática e na dinâmica e a lei de controle baseada na linearização entrada-saída por realimentação. Foram apresentados, também, uma introdução à lógica difusa e os conceitos de controladores difusos do tipo PI e PD.

As técnicas de geração de trajetórias apresentadas são utilizadas pelo gerador de trajetória proposto, para a geração de trajetórias planas, mostradas no capítulo 6. Os conceitos de controladores baseados na cinemática e na dinâmica, bem como a lei de controle baseada na linearização entrada-saída e os conceitos de controladores difusos apresentados, são aplicados no controlador de trajetória proposto, mostrando os resultados e o desempenho dos controladores nos capítulos 5 e 6.

# MODELAGEM CINEMÁTICA E DINÂMICA DO ROBÔ AXEBOT

## 4.1 INTRODUÇÃO

Aspectos de caráter mecânico, elétrico e a disposição física das partes que compõem um robô são características relevantes na escolha e na implementação de seus controladores. A capacidade de um robô se locomover e as equações que regem o seu movimento são influenciadas pela sua geometria e pela sua dinâmica.

Os modelos cinemáticos descrevem a equação do movimento do robô em função das velocidades das rodas sem considerar as forças que atuam sobre o mesmo. Existem dois tipos de modelos: modelo cinemático de postura e o modelo cinemático de configuração. Segundo Lages (1998), os modelos de postura consideram como estado apenas a posição e a orientação do robô, enquanto que os modelos de configuração consideram, além da postura, outras variáveis internas, como por exemplo o deslocamento angular das rodas. O modelo de configuração, no entanto, não se mostra necessário do ponto de vista do controle da posição e orientação do robô.

Os modelos dinâmicos descrevem as relações dinâmicas entre as coordenadas de postura e orientação do robô e os torques desenvolvidos pelos seus atuadores. O modelo dinâmico pode ser formulado segundo o formalismo Newton-Euler (YAMADA et al., 2001; LIU et al., 2003; CARTER et al., 2001; WILSON et al., 2001) ou o formalismo Euler-Lagrange (CHENG; TSAI, 2003; SCHROEDER et al., 2005). De acordo com Yamada *et al.* (2001), enquanto as equações de Newton tratam cada

corpo rígido separadamente e modelam explicitamente as restrições mecânicas de cada corpo através das forças que atuam sobre estes, Lagrange e d' Alembert fornecem um procedimento sistemático para eliminar as restrições mecânicas, através de um sistema de equações geral. As expressões 4.1 e 4.2 mostram as equações de Newton enquanto que a expressão 4.3 mostra a equação de Lagrange e d' Alembert:

$$\sum F = ma \quad (4.1)$$

$$\sum M_o = J\alpha \quad (4.2)$$

onde  $\Sigma F$ ,  $\Sigma M_o$ , são respectivamente, as resultantes das forças e momentos que atuam sobre o centro de massa de um corpo,  $m$  a sua massa,  $a$  a sua aceleração linear,  $J$  sua matriz de inércia e  $\alpha$  a sua aceleração angular.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} T - \frac{\partial L}{\partial q} T = \Sigma Q \quad (4.3)$$

sendo  $q$  uma matriz de coordenadas genéricas,  $L$  o Lagrangeano do sistema, dado pela diferença entre a energia cinética e a energia potencial do sistema e  $\Sigma Q$  o somatório dos esforços externos ou não conservativos.

Para a formulação do modelo dinâmico do Axebot, foi adotado o formalismo de Newton-Euler, devido a uma vasta quantidade de referências encontradas na literatura sobre modelos dinâmicos de robôs semelhantes (LIU et al., 2003; CARTER et al., 2001; OUBBATI, 2006; FRANCO; COSTA, 2006; LAURA, 2006).

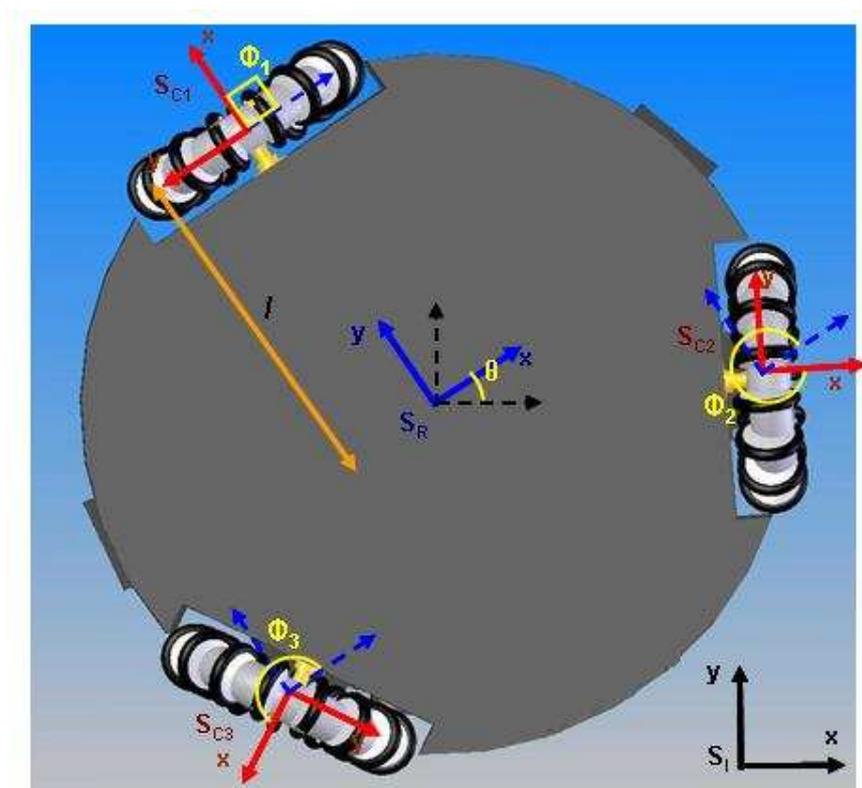
## 4.2 MODELAGEM CINEMÁTICA DO AXEBOT

O modelo cinemático de postura foi o modelo adotado no Axebot, mostrando ser, juntamente com os aspectos dinâmicos do robô, suficiente para o controle de sua posição e orientação. Para a formulação das equações cinemáticas é necessário conhecer as relações entre as componentes de velocidade do robô em alguns pontos de referência,

adotando em cada referencial um sistema de coordenadas cartesianas.

Na formulação do modelo cinemático foram feitas algumas considerações: Só existe um ponto de contato da roda com o solo e a mesma não sofre nenhum deslizamento; o plano da roda permanece sempre na vertical e não existe nenhuma deformação das partes que compõem o robô. Williams *et al.*(2002) formula o modelo cinemático levando em consideração o deslizamento das rodas sobre o plano de contato.

Para o robô Axebot, foram definidos cinco sistemas de coordenadas de referência, sendo quatro deles sistemas de referência locais ou solidários ao robô e um sistema de referência inercial  $S_I$ . O sistema de referência local  $S_R$ , possui sua origem sobre o centro de massa do robô e cada sistema de referência local  $S_{C_i}$  é fixado na roda  $i$ , sendo  $i = 1, 2, 3$  (Ver figura 4.1).



**Figura 4.1.** Geometria do Axebot e seus sistemas de coordenadas.

A figura acima mostra a visão do topo do Axebot com suas características geométricas,

sendo  $l$  a distância entre os centros dos sistemas de coordenadas  $S_R$  e  $S_{Ci}$ ,  $\phi_i$  o ângulo de inclinação da roda  $i$  em relação ao sistema  $S_R$  e  $\theta$  o ângulo de orientação do robô em relação ao sistema  $S_I$ .

A posição e a orientação do centro de massa do robô em relação ao sistema de coordenada  $S_I$  pode ser representada pela equação

$$\xi_I = \begin{bmatrix} x_I \\ y_I \\ \theta \end{bmatrix} \quad (4.4)$$

onde  $\xi_I$  representa o vetor de postura do robô, sendo  $x_I$  e  $y_I$ , respectivamente a posição do robô sobre os eixos  $x$  e  $y$  do sistema  $S_I$  e  $\theta$  a sua orientação. A relação das componentes de velocidade do centro de massa do robô entre o sistema de referência  $S_I$  e o sistema de referência  $S_R$  é mostrada pela equação 4.5.

$$\dot{\xi}_I = R^{-1}(\theta)\dot{\xi}_R \quad (4.5)$$

$\dot{\xi}_R$  são as componentes de velocidade do centro de massa do robô sobre o sistema  $S_R$ , representadas pela equação 4.6,  $\dot{\xi}_I$  as componentes de velocidade do centro de massa do robô sobre o sistema  $S_I$ , representadas pela equação 4.7 e  $R(\theta)$  a matriz ortogonal de rotação do sistema  $S_R$  em relação ao sistema  $S_I$ , representada pela equação 4.8:

$$\dot{\xi}_R = \begin{bmatrix} v_{xR} \\ v_{yR} \\ \dot{\theta} \end{bmatrix} \quad (4.6)$$

$$\dot{\xi}_I = \begin{bmatrix} v_{xI} \\ v_{yI} \\ \dot{\theta} \end{bmatrix} \quad (4.7)$$

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

$v_{xR}$  e  $v_{yR}$  são, respectivamente, as componentes de velocidade do robô em relação aos eixos  $x$  e  $y$  do sistema  $S_R$  e  $v_{xI}$  e  $v_{yI}$ , respectivamente, as componentes de velocidade do robô sobre os eixos  $x$  e  $y$  do sistema  $S_I$ .

As componentes de velocidade do centro de massa do robô sobre o sistema  $S_R$  são função das componentes de velocidade dos pontos sobre a origem do sistema  $S_{Ci}$ . A relação entre estas componentes de velocidade são mostradas na equação 4.9.

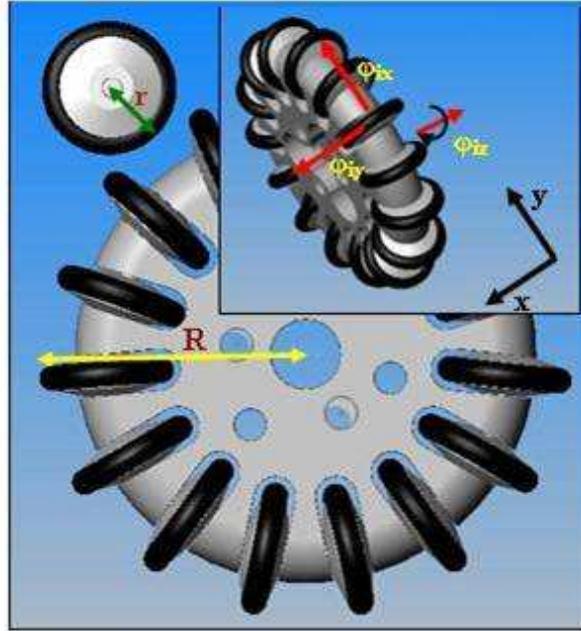
$$\begin{bmatrix} v_{xR} \\ v_{yR} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\phi_i) & -\sin(\phi_i) & l \sin(\phi_i) \\ \sin(\phi_i) & \cos(\phi_i) & -l \cos(\phi_i) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{xCi} \\ v_{yCi} \\ \dot{\theta} \end{bmatrix} \quad (4.9)$$

### 4.2.1 Restrições cinemáticas da roda

A roda, ao mesmo tempo em que é responsável pelo movimento do robô, restringe alguns movimentos do mesmo. Essa restrição é determinada pelo modelo da roda e pelo número de graus de liberdade que ela possui. O grau de mobilidade do robô, portanto, é definido pela geometria e pela disposição das rodas sobre o seu chassi.

A escolha do modelo da roda vai depender da tarefa que se deseja executar com o robô e qual a flexibilidade do seu movimento durante a execução da tarefa. Um estudo mais detalhado sobre os modelos de rodas e suas restrições cinemáticas pode

ser encontrado em Siegwart e Nourbakhsh (2004).



**Figura 4.2.** Geometria da roda omnidirecional utilizada no Axebot.

A figura 4.2 mostra o protótipo das rodas utilizadas no Axebot e a equação 4.10 as suas restrições cinemáticas.

$$\begin{bmatrix} v_{xCi} \\ v_{yCi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & r \sin(\eta) & 0 \\ R & -r \cos(\eta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_{ix} \\ \dot{\varphi}_{iy} \\ \dot{\varphi}_{iz} \end{bmatrix} \quad (4.10)$$

sendo  $\dot{\varphi}_{ix}$  a velocidade angular da roda  $i$  em relação ao eixo  $x$  do sistema  $S_{Ci}$ ,  $\dot{\varphi}_{iy}$  a velocidade angular do rolamento da roda  $i$  em contato com o solo,  $\dot{\varphi}_{iz}$  a velocidade angular da roda  $i$  em relação ao eixo  $z$  (perpendicular aos eixos  $x$  e  $y$ ) do sistema  $S_{Ci}$ ,  $R$  a distância entre a extremidade da roda e o seu centro,  $r$  o raio dos rolamentos e  $\eta$  o ângulo de inclinação dos rolamentos em relação ao eixo  $y$  do sistema  $S_{Ci}$ . Este ângulo é de  $90^\circ$  nas rodas usadas pelo Axebot. É importante notar que, como a roda do Axebot não possui o grau de liberdade de rotação sobre o eixo  $z$  do sistema  $S_{Ci}$ ,  $\dot{\varphi}_{iz}$  é igual a velocidade angular do robô.

Conhecendo os valores de  $\phi_i$ , substituindo a equação 4.10 na equação 4.9 e o resultado na equação 4.5 encontram-se as relações representadas pelas equações 4.11 a 4.13.

Em relação à roda 1:

$$\begin{bmatrix} v_{xR} \\ v_{yR} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -R & 0 & l \\ 0 & r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_{1x} \\ \dot{\varphi}_{1y} \\ \dot{\varphi}_{1z} \end{bmatrix} \quad (4.11)$$

Em relação à roda 2:

$$\begin{bmatrix} v_{xR} \\ v_{yR} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{\sqrt{3}r}{2} & -\frac{l}{2} \\ \frac{\sqrt{3}R}{2} & -\frac{r}{2} & -\frac{\sqrt{3}l}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_{2x} \\ \dot{\varphi}_{2y} \\ \dot{\varphi}_{2z} \end{bmatrix} \quad (4.12)$$

Em relação à roda 3:

$$\begin{bmatrix} v_{xR} \\ v_{yR} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & -\frac{\sqrt{3}r}{2} & -\frac{l}{2} \\ -\frac{\sqrt{3}R}{2} & -\frac{r}{2} & \frac{\sqrt{3}l}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_{3x} \\ \dot{\varphi}_{3y} \\ \dot{\varphi}_{3z} \end{bmatrix} \quad (4.13)$$

Isolando  $\dot{\varphi}_{ix}$  das equações 4.11 a 4.13 obtém-se uma relação entre as componentes de velocidade do robô, relativa ao sistema  $S_R$ , e a velocidade angular das rodas,

representadas por

$$\begin{bmatrix} v_{xR} \\ v_{yR} \\ \dot{\theta} \end{bmatrix} = R \begin{bmatrix} -\frac{2}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{3l} & \frac{1}{3l} & \frac{1}{3l} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_{1x} \\ \dot{\varphi}_{2x} \\ \dot{\varphi}_{3x} \end{bmatrix} \quad (4.14)$$

A equação acima pode ser simplificada pelas equações 4.15 e 4.16, que são respectivamente conhecidas, segundo Siegwart e Nourbakhsh (2004), como restrições cinemáticas de rolamento e restrições cinemáticas de deslizamento lateral da roda. Estas equações são mostradas abaixo:

$$\begin{bmatrix} \sin(\alpha + \beta + \gamma) & -\cos(\alpha + \beta + \gamma) & -l \cos(\beta + \gamma) \end{bmatrix} R(\theta) \dot{\xi}_I - R \dot{\varphi} \cos \gamma = 0 \quad (4.15)$$

$$\begin{bmatrix} \cos(\alpha + \beta + \gamma) & \sin(\alpha + \beta + \gamma) & l \sin(\beta + \gamma) \end{bmatrix} R(\theta) \dot{\xi}_I - R \dot{\varphi} \sin \gamma - r \dot{\varphi}_{sw} = 0 \quad (4.16)$$

onde  $\beta$  é o ângulo de inclinação da roda sobre o eixo  $x$  do sistema  $S_{Ci}$  ( $0^\circ$ , no Axebot),  $\alpha$  equivalente ao ângulo  $\phi_i$ ,  $\gamma$  igual a  $\eta - \frac{\pi}{2}$ ,  $\dot{\varphi}$  equivalente a  $\dot{\varphi}_{ix}$  e  $\dot{\varphi}_{sw}$  equivalente a  $\dot{\varphi}_{iy}$ .

A equação 4.14 é conhecida como modelo cinemático direto. O modelo cinemático inverso é apresentado pela expressão

$$\begin{bmatrix} \dot{\varphi}_{1x} \\ \dot{\varphi}_{2x} \\ \dot{\varphi}_{3x} \end{bmatrix} = \frac{1}{R} \begin{bmatrix} -1 & 0 & l \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & l \\ \frac{1}{2} & -\frac{\sqrt{3}}{2} & l \end{bmatrix} \begin{bmatrix} v_{xR} \\ v_{yR} \\ \dot{\theta} \end{bmatrix} \quad (4.17)$$

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

### 4.3 MODELAGEM DINÂMICA DO AXEBOT

Os movimentos de um corpo rígido são causados pelas forças que atuam sobre ele. Estes movimentos podem ser classificados em movimentos de translação e movimentos de rotação. Um sistema mecânico sofre translação se os vetores deslocamento de todas as partículas são iguais. Se um corpo rígido executa apenas um movimento de translação, a resultante da força externa passa pelo centro de massa do corpo, sendo nulo o momento das forças externas em relação a qualquer eixo que passa pelo centro de massa do corpo. Um sistema mecânico sofre uma rotação de um ângulo  $\theta$  em torno de um eixo  $x$  se todas as partículas do sistema descrevem arcos circulares de ângulo  $\theta$  com os seus centros no eixo  $x$  e com os seus planos perpendiculares a este eixo (BORESI; SCHMIDT, 2001).

O modelo dinâmico do Axebot foi formulado segundo as leis de Newton, utilizando a equação 4.1 para a análise do robô em movimento de translação e a equação 4.2 para a análise do robô em movimento de rotação. As forças que atuam no robô são geradas pelos torques dos atuadores, que sofrem reações das forças de atrito viscoso do contato entre as partes mecânicas do motor, das forças de atrito linear e angular da base móvel do robô e das forças de tração das rodas sobre a superfície.

A seção 4.3.1 descreve o modelo dinâmico da base móvel do Axebot, sem considerar explicitamente o torque gerado pelos atuadores. A seção 4.3.2 descreve o modelo dinâmico dos atuadores, levando em consideração sua parte mecânica e elétrica. A seção 4.3.3 mostra o modelo dinâmico final do Axebot, resultado da união do modelo da base móvel com a inclusão do modelo dos atuadores. O modelo dinâmico de um robô diferencial com inclusão de não linearidades de entrada pode ser visto em Laura (2006).

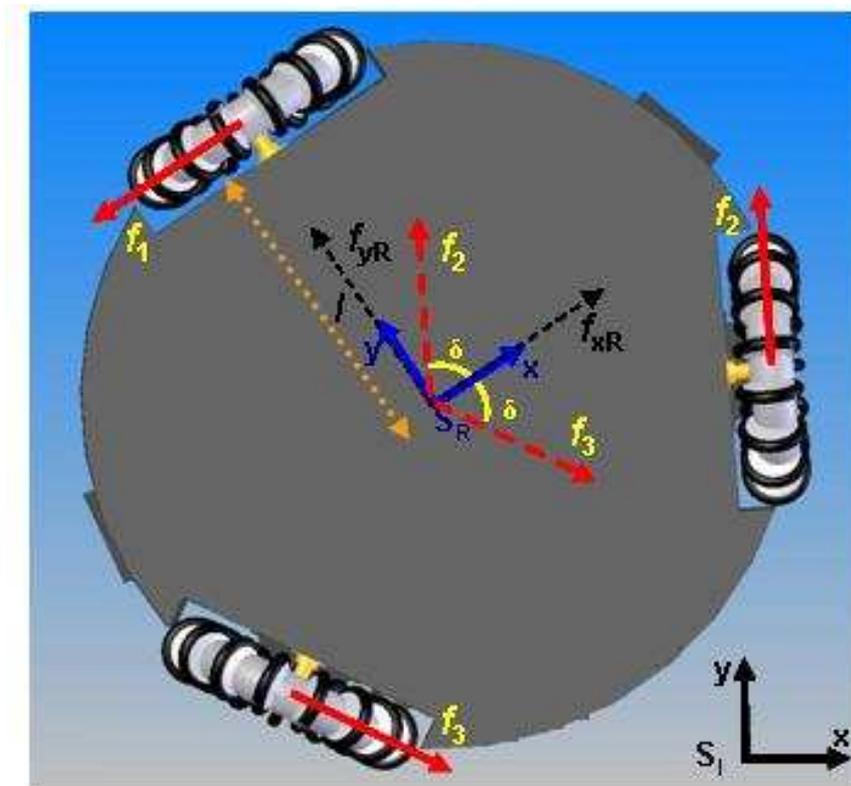


Figura 4.3. Forças que atuam no Axebot.

#### 4.3.1 Modelagem da base móvel

A figura 4.3 mostra a visão do topo do Axebot e as forças de tração  $f_1$ ,  $f_2$  e  $f_3$  que atuam, respectivamente, sobre as rodas 1, 2 e 3. As componentes da força resultante sobre o centro de massa do robô,  $f_{xR}$  e  $f_{yR}$ , são as projeções dessa força sobre o sistema  $S_R$ .  $\delta$  é o ângulo formado entre as forças  $f_2$  e  $f_3$  e o eixo  $x$  do sistema  $S_R$ . De acordo com a equação 4.1 e as forças que atuam no robô, pode-se observar as relações abaixo:

$$f_{xI} = ma_{xI} \quad (4.18)$$

$$f_{yI} = ma_{yI} \quad (4.19)$$

$$\sum M_o = f_1l + f_2l + f_3l = J_r\ddot{\theta} \quad (4.20)$$

$\sum M_o$  é o momento resultante das forças  $f_1$ ,  $f_2$  e  $f_3$  sobre o centro de massa do robô,  $J_r$  o seu momento de inércia em relação ao eixo  $z$  do sistema  $S_R$  (perpendicular aos seus eixos  $x$  e  $y$ ),  $f_{xI}$  e  $f_{yI}$  são as projeções das componentes da força resultante em relação ao sistema inercial  $S_I$ ,  $m$  a massa do robô e  $a_{xI}$  e  $a_{yI}$  as suas componentes de aceleração em relação ao sistema  $S_I$ .

Representando as equações 4.18 a 4.20 em forma matricial obtém-se:

$$\begin{bmatrix} f_{xI} \\ f_{yI} \\ \sum M_o \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J_r \end{bmatrix} \begin{bmatrix} a_{xI} \\ a_{yI} \\ \ddot{\theta} \end{bmatrix} \quad (4.21)$$

Pode-se observar que:

$$F_I = R(\theta)^{-1} F_R \quad (4.22)$$

$$\ddot{\xi}_I = R(\theta)^{-1} \ddot{\xi}_R + R(\dot{\theta})^{-1} \dot{\xi}_R \quad (4.23)$$

Sendo  $F_I = [f_{xI} \ f_{yI} \ \sum M_o]^T$ ,  $F_R = [f_{xR} \ f_{yR} \ \sum M_o]^T$  e  $R(\dot{\theta})^{-1}$  o jacobiano da inversa da matriz  $R(\theta)$ . Substituindo as equações 4.22 e 4.23 em 4.18 e 4.19, chegam-se às seguintes expressões:

$$f_{xR} = m(a_{xR} - v_{yR}\dot{\theta}) \quad (4.24)$$

$$f_{yR} = m(a_{yR} + v_{xR}\dot{\theta}) \quad (4.25)$$

Escrevendo  $f_{xR}$  e  $f_{yR}$  em função de  $f_1$ ,  $f_2$  e  $f_3$  obtém-se:

$$f_{xR} = -f_1 + f_2 \cos \delta + f_3 \cos \delta \quad (4.26)$$

$$f_{yR} = f_2 \sin \delta + f_3 \sin \delta \quad (4.27)$$

Substituindo as equações 4.26 e 4.27 em 4.24 e 4.25 e isolando as componentes de aceleração das equações resultantes e da equação 4.20, obtém-se o modelo dinâmico da

base móvel do robô, expresso pelas equações:

$$a_{xR} = \frac{-f_1 + f_2 \cos \delta + f_3 \cos \delta}{m} + v_{yR} \dot{\theta} \quad (4.28)$$

$$a_{yR} = \frac{f_2 \sin \delta - f_3 \sin \delta}{m} - v_{xR} \dot{\theta} \quad (4.29)$$

$$\ddot{\theta} = \frac{f_1 l + f_2 l + f_3 l}{J_r} \quad (4.30)$$

#### 4.3.2 Modelagem dos atuadores

Os atuadores do robô Axebot são motores de corrente contínua (DC), modelo Amax 22 R179-6V, da marca Maxon. O motor é acoplado a uma caixa de redução com um fator de redução de 19:1 (ver figura 4.4).



**Figura 4.4.** Motor Maxon modelo Amax 22 R179-6V.

O modelo dinâmico de um motor DC é formulado levando em conta os seus aspectos mecânicos e elétricos. De acordo com Ogata (2003), as equações dinâmicas de um motor DC podem ser expressas por:

Pela parte elétrica:

$$u_i = L_a \frac{di_{a_i}}{dt} + R_a i_{a_i} + e_i \quad (4.31)$$

Pela parte mecânica:

$$\tau_i = J_m \dot{\omega}_{m_i} + B_{vis} \omega_{m_i} \quad (4.32)$$

sendo  $L_a$  e  $R_a$ , respectivamente, a indutância e a resistência dos motores,  $J_m$  e  $B_{vis}$ , respectivamente, o momento de inércia e a constante de atrito viscoso da parte mecânica dos atuadores e  $u_i$ ,  $i_{a_i}$ ,  $\tau_i$  e  $\omega_{m_i}$ , respectivamente, a tensão de entrada, a corrente de armadura, o torque e a velocidade do motor acoplado a roda  $i$ .

O torque do motor é diretamente proporcional à sua corrente de armadura e a força contra-eletromotriz proporcional à sua velocidade angular. As constantes  $k_T$  e  $K_{em}$ , das equações abaixo, são, respectivamente, as constantes de torque e contra-eletromotriz do motor.

$$\tau_i = k_T i_{a_i} \quad (4.33)$$

$$e_i = k_{em} \omega_{m_i} \quad (4.34)$$

A parte mecânica do sistema de atuação do Axebot é composta pelo conjunto motor-roda-redução. O modelo da parte mecânica dos atuadores é formulado de acordo com a equação 4.32, considerando o motor com carga. A equação 4.35 representa este modelo.

$$\tau_i = J_m \dot{\omega}_{m_i} + B_{vis} \omega_{m_i} + \frac{R f_i}{\kappa n} \quad (4.35)$$

Observando que a única diferença entre a equação 4.35 e a equação 4.32 é o termo  $\frac{R f_i}{\kappa n}$ , que aparece devido ao momento de resistência ao torque do motor exercida pela força de tração  $f_i$  da roda  $i$  (ver figura 4.3). A constante  $n$  é o fator de redução da caixa de redução acoplada ao motor e  $\kappa$  o grau de eficiência do acoplamento.

As forças  $f_1$ ,  $f_2$  e  $f_3$ , mostradas na figura 4.3, são forças que se opõem aos torques gerados pelos atuadores. Essas forças aparecem quando as rodas do robô estão em

contato com o solo e quando os atuadores estão gerando os torques sobre elas. Portanto, analisando o conjunto roda-atuador, existem basicamente dois momentos sendo aplicados sobre o centro de cada roda. Um deles é o torque do atuador, o outro é o produto da força  $f_i$  pela distância  $R$  da roda(ver figura 4.2).

Relacionando  $u_i$  em função de  $\tau_i$  obtém-se:

$$u_i = L_a \frac{\dot{\tau}_i}{k_T} + R_a \frac{\tau_i}{k_T} + k_{em} \omega_{m_i} \quad (4.36)$$

Substituindo os valores de  $\tau_i$  e  $\dot{\tau}_i$  (equação 4.32 e sua derivada, respectivamente) na equação 4.36 chega-se a expressão

$$u_i = \frac{L_a J_m}{k_T} \ddot{\omega}_{m_i} + \left( \frac{R_a J_m + L_a B_{vis}}{k_T} \right) \dot{\omega}_{m_i} + \left( \frac{R_a B_{vis} + k_T k_{em}}{k_T} \right) \omega_{m_i} + \frac{R_a f_i R}{\kappa n k_T} + \frac{L_a \dot{f}_i R}{\kappa n k_T} \quad (4.37)$$

A equação acima relaciona a velocidade angular do motor com a sua tensão de entrada. Os motores, porém, são acionados por uma largura de pulso  $\delta_i$  que gera a tensão de entrada  $u_i$  do motor  $i$  em função de uma tensão média de referência  $V_{ref}$ . A relação entre  $\delta_i$  e a constante  $V_{ref}$  é expressa por

$$\delta_i = \frac{u_i}{V_{ref}} \quad (4.38)$$

Substituindo 4.38 em 4.37, a dinâmica do atuador terá como entrada a largura de pulso. Esta dinâmica é representada em 4.39

$$\begin{aligned} \delta_i = & \frac{L_a J_m}{k_T V_{ref}} \ddot{\omega}_{m_i} + \left( \frac{R_a J_m + L_a B_{vis}}{k_T V_{ref}} \right) \dot{\omega}_{m_i} + \left( \frac{R_a B_{vis} + k_T k_{em}}{k_T V_{ref}} \right) \omega_{m_i} \\ & + \frac{R_a f_i R}{n \kappa k_T V_{ref}} + \frac{L_a \dot{f}_i R}{n \kappa k_T V_{ref}} \end{aligned} \quad (4.39)$$

### 4.3.3 Modelo dinâmico da base móvel com inclusão dos atuadores

Isolando as forças  $f_i$  das equações 4.28 a 4.30 são obtidas as equações

$$f_1 = \frac{2mla_{xR} + J_r \ddot{\theta} + 2ml\dot{\theta} v_{yR}}{3l} \quad (4.40)$$

$$f_2 = \frac{mla_{xR} + ml\sqrt{3}a_{yR} + J_r \ddot{\theta} - ml\dot{\theta}v_{yR} + ml\sqrt{3}\dot{\theta} v_{xR}}{3l} \quad (4.41)$$

$$f_3 = \frac{mla_{xR} - ml\sqrt{3}a_{yR} + J_r \ddot{\theta} - ml\dot{\theta}v_{yR} - ml\sqrt{3}\dot{\theta} v_{xR}}{3l} \quad (4.42)$$

Para a formulação das equações da dinâmica inversa da base móvel, representadas em 4.43 a 4.45, os seguintes passos foram executados:

- Foram substituídos os valores de  $f_i$  de 4.40, 4.41 e 4.42 e as suas respectivas derivadas,  $\dot{f}_i$ , em 4.39
- Das equações resultantes, da execução do passo anterior, foram substituídos os valores de  $v_{xR}$  e  $v_{yR}$  pelos da equação 4.14.
- Ainda das equações resultantes, da execução do primeiro passo, foram substituídos os valores de  $a_{xR}$  e  $a_{yR}$  pelos valores da primeira derivada, no tempo, da equação 4.14

$$\begin{aligned} \delta_1 = & \beta_{i2}\ddot{\omega}_{m1} + \beta_{i3}\ddot{\omega}_{m2} + \beta_{i3}\ddot{\omega}_{m3} + (\beta_{i4} + \beta_{i5}\omega_{m2} - \beta_{i5}\omega_{m3})\dot{\omega}_{m1} + (\beta_{i6} + 2\beta_{i5}\omega_{m2} \\ & + \beta_{i5}\omega_{m1})\dot{\omega}_{m2} + (\beta_{i6} - \beta_{i5}\omega_{m1} - 2\beta_{i5}\omega_{m3})\dot{\omega}_{m3} + (\beta_{i7} + \beta_{i8}\omega_{m2} - \beta_{i8}\omega_{m3})\omega_{m1} \\ & + (\beta_{i8}\omega_{m2})\omega_{m2} + (-\beta_{i8}\omega_{m3})\omega_{m3} \end{aligned} \quad (4.43)$$

$$\begin{aligned} \delta_2 = & \beta_{i3}\ddot{\omega}_{m1} + \beta_{i2}\ddot{\omega}_{m2} + \beta_{i3}\ddot{\omega}_{m3} + (\beta_{i6} - 2\beta_{i5}\omega_{m1} + \beta_{i5}\omega_{m2})\dot{\omega}_{m1} + (\beta_{i4} - \beta_{i5}\omega_{m1} \\ & + \beta_{i5}\omega_{m2})\dot{\omega}_{m2} + (\beta_{i6} + \beta_{i5}\omega_{m2} + 2\beta_{i5}\omega_{m3})\dot{\omega}_{m3} + (-\beta_{i8}\omega_{m1} - \beta_{i8}\omega_{m2})\omega_{m1} \\ & + (\beta_{i7} + \beta_{i8}\omega_{m3})\omega_{m2} + (\beta_{i8}\omega_{m3})\omega_{m3} \end{aligned} \quad (4.44)$$

$$\begin{aligned}
 \delta_3 = & \beta_{i3}\ddot{\omega}_{m_1} + \beta_{i3}\ddot{\omega}_{m_2} + \beta_{i2}\ddot{\omega}_{m_3} + (\beta_{i6} - 2\beta_{i5}\omega_{m_1} + \beta_{i5}\omega_{m_3})\dot{\omega}_{m_1} + (\beta_{i6} - 2\beta_{i5}\omega_{m_2} \\
 & - \beta_{i5}\omega_{m_3})\dot{\omega}_{m_2} + (\beta_{i4} + \beta_{i5}\omega_{m_1} - \beta_{i5}\omega_{m_2})\dot{\omega}_{m_3} + (\beta_{i8}\omega_{m_1} + \beta_{i8}\omega_{m_3})\omega_{m_1} \\
 & + (-\beta_{i8}\omega_{m_2} - \beta_{i8}\omega_{m_3})\omega_{m_2} + \beta_{i7}\omega_{m_3}
 \end{aligned} \tag{4.45}$$

Sendo:

$$\begin{aligned}
 \beta_{i1} &= 1/(27n^3l^2\kappa k_T V_{ref}) \\
 \beta_{i2} &= (3L_a R^2 J_r n + 12L_a R^2 ml^2 n + 27L_a J_m l^2 \kappa n^3)\beta_{i1} \\
 \beta_{i3} &= (-6L_a R^2 ml^2 n + 3L_a R^2 J_r n)\beta_{i1} \\
 \beta_{i4} &= (12R_a R^2 ml^2 n + 27l^2 \kappa n^3 L_a B_{vis} + 27l^2 \kappa n^3 R_a J_m + 3R_a R^2 J_r n)\beta_{i1} \\
 \beta_{i5} &= (2L_a R^3 \sqrt{3}ml)\beta_{i1} \\
 \beta_{i6} &= (-6R_a R^2 ml^2 n + 3R_a R^2 J_r n)\beta_{i1} \\
 \beta_{i7} &= (27l^2 \kappa n^3 k_{em} + 27l^2 \kappa n^3 R_a B_{vis})\beta_{i1} \\
 \beta_{i8} &= (2R_a R^3 \sqrt{3}ml)\beta_{i1}
 \end{aligned}$$

Isolando as componentes da derivada da aceleração angular do motor, das equações 4.43 a 4.45, encontram-se as equações da dinâmica da base móvel do robô com a inclusão da dinâmica dos atuadores, representadas por

$$\begin{aligned}
 \ddot{\omega}_{m_1} = & \beta_{d2}\delta_1 + \beta_{d3}\delta_2 + \beta_{d3}\delta_3 + (\beta_{d4} + \beta_{d5}\omega_{m_2} - \beta_{d5}\omega_{m_3})\dot{\omega}_{m_1} + (\beta_{d6} + 2\beta_{d5}\omega_{m_2} \\
 & + \beta_{d5}\omega_{m_1})\dot{\omega}_{m_2} + (\beta_{d6} - \beta_{d5}\omega_{m_1} - 2\beta_{d5}\omega_{m_3})\dot{\omega}_{m_3} + (\beta_{d7} + \beta_{d8}\omega_{m_2} - \beta_{d8}\omega_{m_3})\omega_{m_1} \\
 & + (\beta_{d9} + \beta_{d8}\omega_{m_2})\omega_{m_2} + (\beta_{d9} - \beta_{d8}\omega_{m_3})\omega_{m_3}
 \end{aligned} \tag{4.46}$$

$$\begin{aligned}
 \ddot{\omega}_{m_2} = & \beta_{d3}\delta_1 + \beta_{d2}\delta_2 + \beta_{d3}\delta_3 + (\beta_{d6} - 2\beta_{d5}\omega_{m_1} + \beta_{d5}\omega_{m_2})\dot{\omega}_{m_1} + (\beta_{d4} - \beta_{d5}\omega_{m_1} \\
 & + \beta_{d5}\omega_{m_2})\dot{\omega}_{m_2} + (\beta_{d6} + \beta_{d5}\omega_{m_2} + 2\beta_{d5}\omega_{m_3})\dot{\omega}_{m_3} + (\beta_{d9} - \beta_{d8}\omega_{m_1} - \beta_{d8}\omega_{m_2})\omega_{m_1} \\
 & + (\beta_{d7} + \beta_{d8}\omega_{m_3})\omega_{m_2} + (\beta_{d9} + \beta_{d8}\omega_{m_3})\omega_{m_3}
 \end{aligned} \tag{4.47}$$

$$\begin{aligned}
 \ddot{\omega}_{m_3} = & \beta_{d3}\delta_1 + \beta_{d3}\delta_2 + \beta_{d2}\delta_3 + (\beta_{d6} - 2\beta_{d5}\omega_{m_1} + \beta_{d5}\omega_{m_3})\dot{\omega}_{m_1} + (\beta_{d6} - 2\beta_{d5}\omega_{m_2} \\
 & - \beta_{d5}\omega_{m_3})\dot{\omega}_{m_2} + (\beta_{d4} + \beta_{d5}\omega_{m_1} - \beta_{d5}\omega_{m_2})\dot{\omega}_{m_3} + (\beta_{d9} + \beta_{d8}\omega_{m_1} + \beta_{d8}\omega_{m_3})\omega_{m_1} \\
 & + (\beta_{d9} - \beta_{d8}\omega_{m_2} - \beta_{d8}\omega_{m_3})\omega_{m_2} + \beta_{d7}\omega_{m_3}
 \end{aligned} \tag{4.48}$$

Sendo:

$$\begin{aligned}
 \beta_{d1} &= -1/9/l/n/(9J_m^2 l^2 \kappa^2 n^4 + 3R^2 J_r J_m \kappa n^2 + 6R^2 m l^2 J_m \kappa n^2 + 2R^4 J_r m)/L_a \\
 \beta_{d2} &= (-81V_{ref} k_T \kappa^2 n^5 l^3 J_m - 18V_{ref} k_T \kappa n^3 R^2 l^3 m - 18V_{ref} k_T \kappa n^3 R^2 l J_r) \beta_{d1} \\
 \beta_{d3} &= (9V_{ref} k_T \kappa n^3 l R^2 J_r - 18V_{ref} k_T \kappa n^3 l^3 R^2 m) \beta_{d1} \\
 \beta_{d4} &= (27J_m l \kappa n^3 R_a R^2 J_r + 54J_m l^3 \kappa n^3 R_a R^2 m + 81J_m l^3 \kappa^2 n^5 L_a B_{vis} + 81J_m^2 l^3 \kappa^2 n^5 R_a \\
 & + 18R^4 J_r R_a m l n + 18R^2 m l^3 \kappa n^3 L_a B_{vis} + 18R^2 J_r l \kappa n^3 L_a B_{vis}) \beta_{d1} \\
 \beta_{d5} &= (2R^5 J_r L_a \sqrt{3} m + 6J_m l^2 \kappa n^2 L_a R^3 \sqrt{3} m) \beta_{d1} \\
 \beta_{d6} &= (18R^2 m l^3 \kappa n^3 L_a B_{vis} - 9R^2 J_r l \kappa n^3 L_a B_{vis}) \beta_{d1} \\
 \beta_{d7} &= (81J_m l^3 \kappa^2 n^5 R_a B_{vis} + 81J_m l^3 \kappa^2 n^5 k_{em} k_T + 18R^2 J_r l \kappa n^3 k_{em} k_T \\
 & + 18R^2 J_r l \kappa n^3 R_a B_{vis} + 18R^2 m l^3 \kappa n^3 k_{em} k_T + 18R^2 m l^3 \kappa n^3 R_a B_{vis}) \beta_{d1} \\
 \beta_{d8} &= (2R^5 J_r R_a \sqrt{3} m + 6J_m l^2 \kappa n^2 R_a R^3 \sqrt{3} m) \beta_{d1} \\
 \beta_{d9} &= (18R^2 m l^3 \kappa n^3 R_a B_{vis} + 18R^2 m l^3 \kappa n^3 k_{em} k_T - 9R^2 J_r l \kappa n^3 k_{em} k_T \\
 & - 9R^2 J_r l \kappa n^3 R_a B_{vis}) \beta_{d1}
 \end{aligned}$$

Observando que, nas expressões 4.43 a 4.45, os valores das velocidades angulares das rodas,  $\dot{\varphi}_{ix}$ , foram substituídos pelos valores das velocidades angulares dos motores,  $\omega_{m_i}$ , conhecida a relação  $\dot{\varphi}_{ix} = \frac{\omega_{m_i}}{n}$ .

Pode-se concluir, em relação ao modelo do Axebot, que:

- O sistema é acoplado, dificultando a utilização de controladores SISO.
- O sistema de equações em 4.46 a 4.48 é determinado e admite uma única solução, facilitando encontrar uma lei de controle por realimentação baseada na inversa

do modelo dinâmico.

- Ainda como consequência do item anterior, pode-se controlar a orientação do robô, independente do controle de sua posição.
- O sistema se encontra na forma companheira, como será visto no próximo capítulo, facilitando encontrar um lei de controle por linearização entrada-saída, sem a necessidade de derivar a equação de saída do sistema.

As leis de controle baseadas nos modelos cinemáticos e dinâmicos do robô Axebot são mostradas no próximo capítulo.

#### 4.4 CONCLUSÃO

O desenvolvimento dos modelos cinemático e dinâmico do robô AxeBot foi apresentado neste capítulo. Estes modelos servirão de base para a formulação das leis de controle baseadas na cinemática e na dinâmica do controlador de trajetória proposto, mostradas em detalhes no próximo capítulo.

Pode-se concluir que, apesar de o sistema ser acoplado, dificultando a utilização de controladores SISO, o sistema é determinado e admite uma única solução, facilitando a utilização de leis de controle por realimentação baseada na inversa do modelo dinâmico.

# ANÁLISE DE DESEMPENHO DOS CONTROLADORES PROPOSTOS. ANÁLISE BASEADA NO MODELO DO ROBÔ AXEBOT

## 5.1 INTRODUÇÃO

O objetivo deste capítulo é analisar o desempenho dos controladores de trajetória utilizando o modelo do AxeBot como referência, começando pelo controlador mais simples, baseado apenas no modelo cinemático do robô, avançando até os controladores em cascata, com compensadores de ruídos, utilizando controladores difusos do tipo PI.

A análise do desempenho dos controladores foi feita através das respostas dos mesmos a sinais de posição e velocidade de referência. O índice de desempenho utilizado foi o ISE (*Integral of Squared Error*) - Integral do quadrado do erro. A simulação feita foi baseada nos modelos cinemático e dinâmico do robô Axebot, mostrados no capítulo anterior.

## 5.2 CONTROLE BASEADO NA CINEMÁTICA

O controle baseado na cinemática gera as componentes de velocidade de referência do robô, em relação ao sistema de coordenadas de referência local, em função do erro de postura (diferença entre a postura de referência e a postura do robô).

É mostrado em Muir e Neuman (1986) que uma ação de controle proporcional é suficiente para fazer convergir o erro de postura a zero, em um controlador cinemático.

A lei de controle para o controlador cinemático pode ser representada pela equação 5.1

$$\dot{\xi}_{R_r} = (\xi_{I_r} - \xi_I)R(\theta)K \quad (5.1)$$

onde  $\dot{\xi}_{R_r}$  é a velocidade de referência do robô em relação ao sistema local de referência,  $\xi_{I_r}$  a postura de referência,  $\xi_I$  a postura do robô,  $R(\theta)$  a matriz de transformação do sistema de referência inercial para o sistema local, representada pela equação 4.8, e  $K$  uma matriz diagonal de constantes positivas.

O diagrama de blocos do controle baseado na cinemática é mostrado na figura 5.1

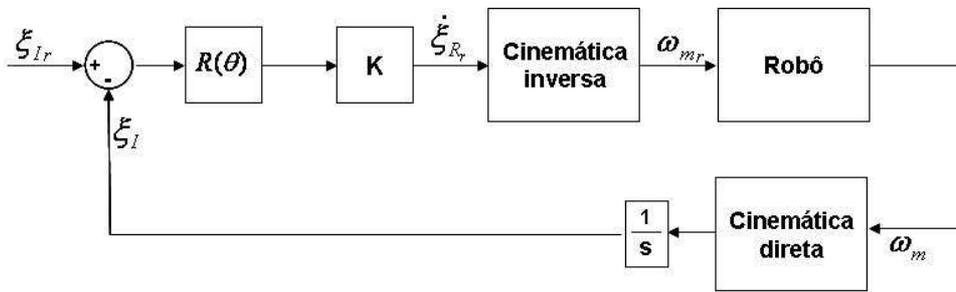


Figura 5.1. Diagrama de blocos do controle baseado na cinemática.

### 5.3 CONTROLE BASEADO NA DINÂMICA

O controlador baseado na dinâmica fecha a malha na velocidade do robô, em relação ao seu sistema de coordenadas de referência local, ou fecha a malha nas velocidades angulares dos motores. Os sinais de controle são os torques aplicados aos atuadores. A velocidade local de referência do robô é calculada pela transformação linear da derivada da postura de referência (velocidade linear e angular do robô em relação ao sistema de coordenadas de referência inercial).

Para este controlador, utilizou-se a lei de controle baseada na linearização por realimentação estática de estado. Pode-se observar que o sistema de equações diferenciais

regido pelas equações 4.46 a 4.48 se encontra na forma 3.49, sendo

$$\mathbf{x}^{(n)} = \begin{bmatrix} \ddot{\omega}_1 \\ \ddot{\omega}_2 \\ \ddot{\omega}_3 \end{bmatrix} \quad (5.2)$$

$$\mathbf{x} = \begin{bmatrix} \omega_1 \\ \dot{\omega}_1 \\ \omega_2 \\ \dot{\omega}_2 \\ \omega_3 \\ \dot{\omega}_3 \end{bmatrix} \quad (5.3)$$

$$\mathbf{u} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix} \quad (5.4)$$

A lei de controle baseada na equação 3.50 pode ser obtida pelas equações 4.43 a 4.45, substituindo  $\mathbf{x}^{(n)}$  por  $\mathbf{v}$ , de acordo com 3.52, sendo

$$\mathbf{v} = \begin{bmatrix} \ddot{\omega}_{1_r} - k_1(\omega_{1_r} - \omega_1) - k_2(\dot{\omega}_{1_r} - \dot{\omega}_1) \\ \ddot{\omega}_{2_r} - k_1(\omega_{2_r} - \omega_2) - k_2(\dot{\omega}_{2_r} - \dot{\omega}_2) \\ \ddot{\omega}_{3_r} - k_1(\omega_{3_r} - \omega_3) - k_2(\dot{\omega}_{3_r} - \dot{\omega}_3) \end{bmatrix} \quad (5.5)$$

onde  $\omega_{1_d}$ ,  $\omega_{2_d}$  e  $\omega_{3_d}$  são as velocidades angulares de referência dos motores acoplados às rodas do robô.

Os valores dos parâmetros do Axebot e dos controladores são mostrados na tabela 5.1.

Parâmetro	Valor	Unidade
$m$	3,4	$kg$
$J_r$	2,125	$kgm^2$

$l$	0,09	$m$
$r$	0,024	$m$
$L$	0,00011	H
$R$	1,71	Ohm
$n$	19:1	-
$J_m$	$3,88 \times 10^{-7}$	$kgm^2$
$K_{em}$	0,0059	Volt s/rad
$K_t$	0,0059	N m/A
$\mu_{vis}$	$2,4 \times 10^{-6}$	N m s/rad
$\eta$	0,83	-
$V_d$	6	Volt
$k_1$	300	-
$k_2$	30	-

Tabela 5.1: Valores dos parâmetros do Axebot e dos controladores

O diagrama de blocos do controle baseado na dinâmica é mostrado na figura 5.2

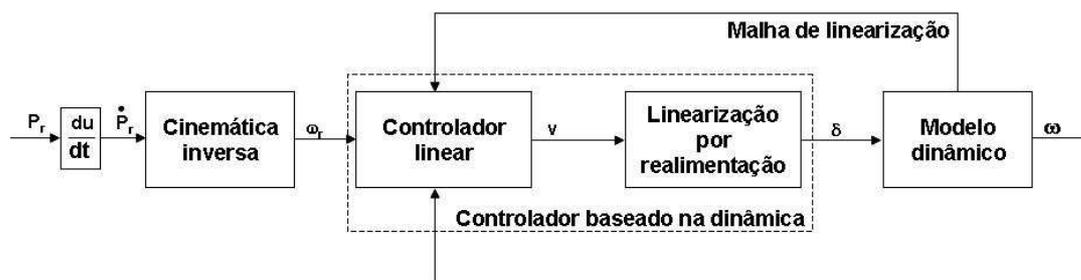
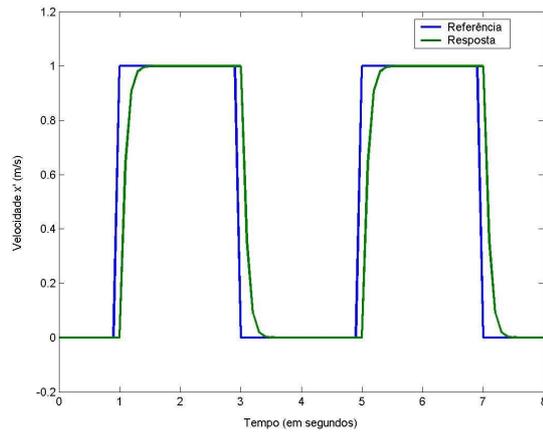
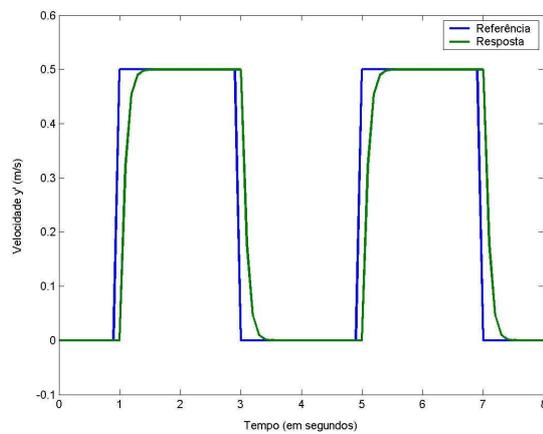


Figura 5.2. Diagrama de blocos do controle baseado na dinâmica.

A análise do desempenho do controlador foi baseada primeiramente na resposta do sistema a sinais pulsantes de velocidade linear, com a velocidade angular de referência do robô nula, considerando o sistema ausente a ruídos e incertezas dos parâmetros do modelo dinâmico. O resultado pode ser observado nas figuras 5.3 e 5.4



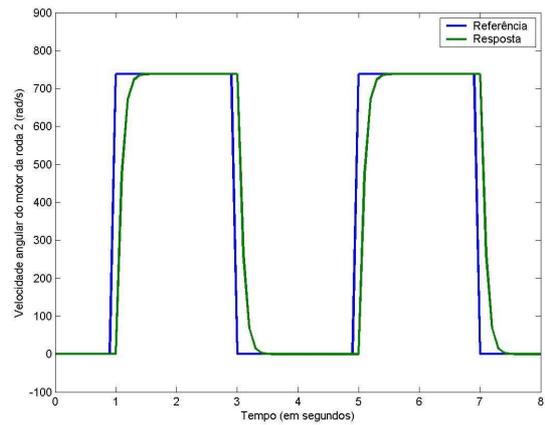
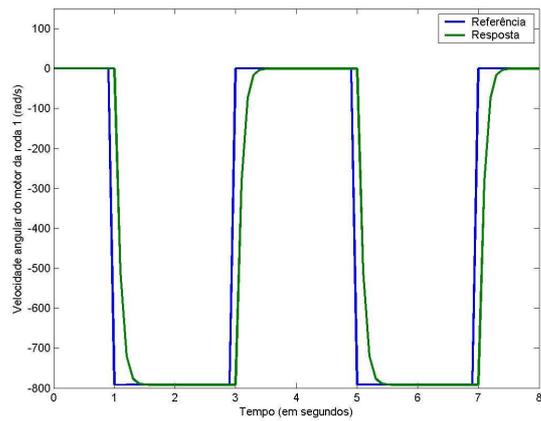
(a) Resposta à componente de velocidade x



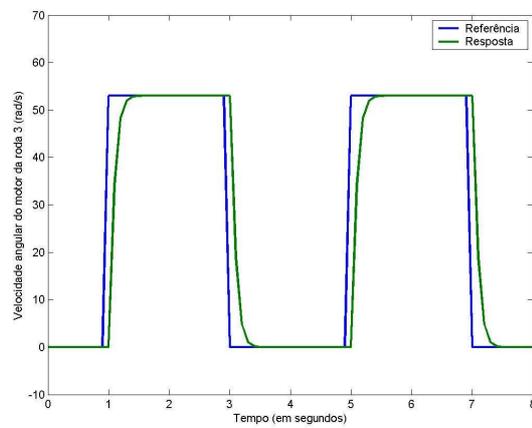
(b) Resposta à componente de velocidade y

**Figura 5.3.** Resposta do sistema às componentes de velocidade linear de referência do robô, não nula, e angular nula

### 5.3 CONTROLE BASEADO NA DINÂMICA



(a) Resposta à velocidade angular do motor da roda 1 (b) Resposta à velocidade angular do motor da roda 2



(c) Resposta à velocidade angular do motor da roda 3

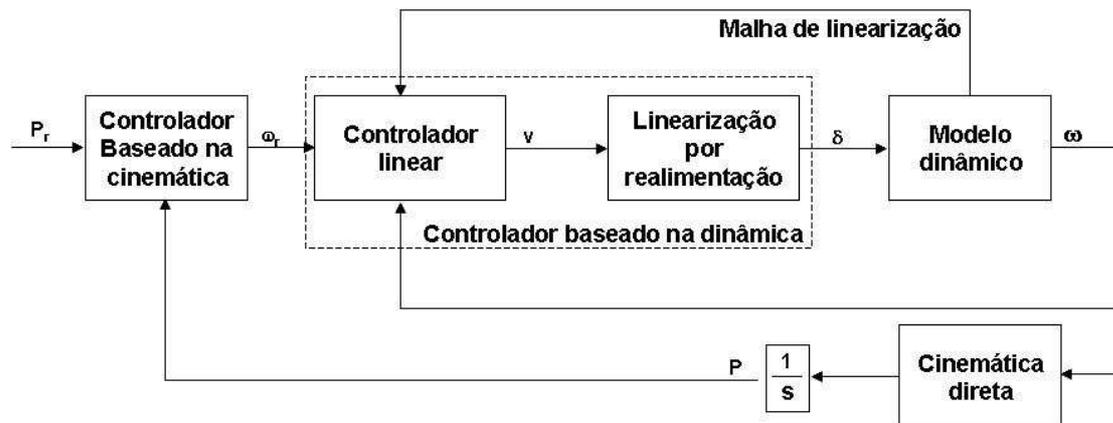
**Figura 5.4.** Resposta do sistema às velocidades angulares de referência dos motores com a velocidade angular de referência do robô nula

Quando a velocidade angular do robô é diferente de zero ou quando a sua referência de posição muda rapidamente, existe um erro em regime estacionário do sinal de postura e de derivada da postura do robô (posição, orientação e velocidade do robô em relação ao sistema de coordenadas de referência inercial).

## 5.4 CONTROLE EM CASCATA

O problema do erro em regime estacionário, quando usado o controle baseado na dinâmica, pode ser solucionado fechando a malha de controle de postura, podendo portanto ser usado um sistema de controle em cascata. Uma malha de controle externa, baseada na cinemática, gerando as velocidades angulares de referência dos motores, em função do erro de postura, e uma malha interna baseada na dinâmica.

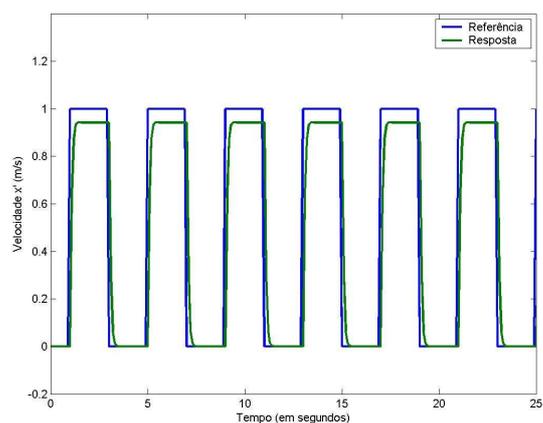
A figura 5.5 mostra o diagrama de blocos do controlador em cascata.



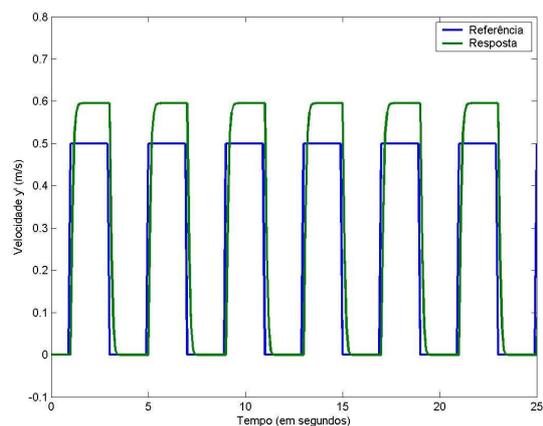
**Figura 5.5.** Diagrama de blocos do controle em cascata.

A figura 5.6 mostra as respostas do controlador baseado na dinâmica e dos controladores em cascata a sinais de velocidade linear e angular de referência, não nulas.

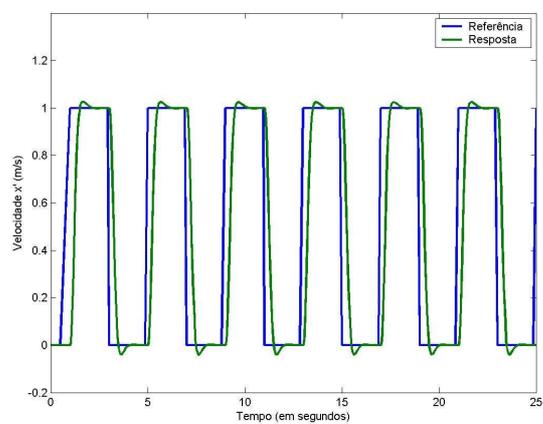
## 5.4 CONTROLE EM CASCATA



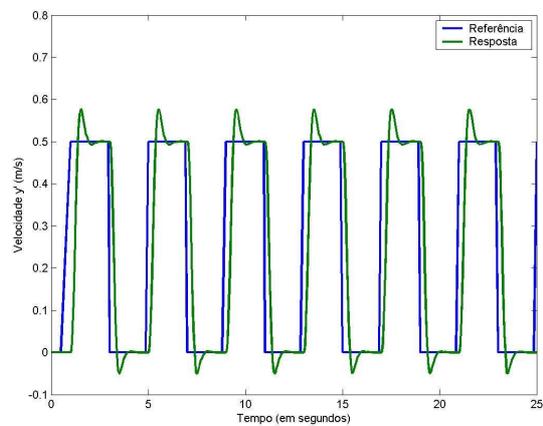
(a) Resposta à componente de velocidade  $x$ , usando o controle baseado na dinâmica



(b) Resposta à componente de velocidade  $y$ , usando o controle baseado na dinâmica



(c) Resposta à componente de velocidade  $x$ , usando o controle em cascata

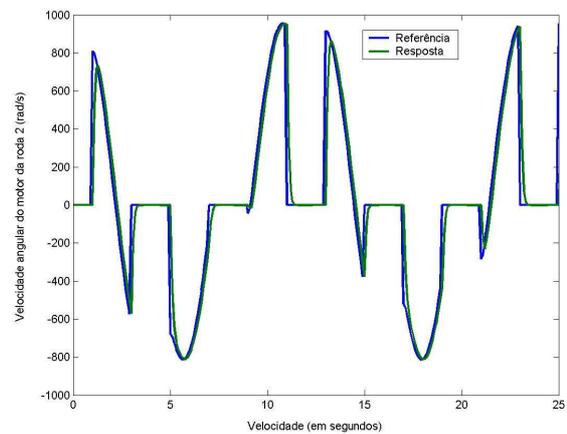
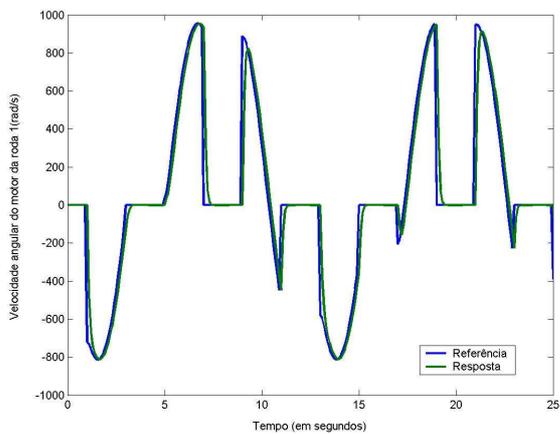


(d) Resposta à componente de velocidade  $y$ , usando o controle em cascata

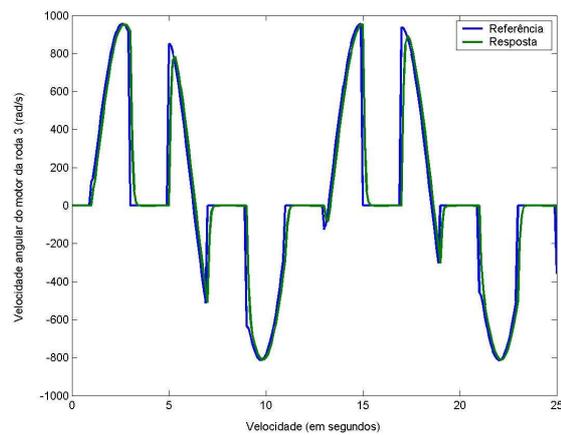
**Figura 5.6.** Resposta do sistema às componentes de velocidade linear com a velocidade angular de referência do robô não nula

## 5.4 CONTROLE EM CASCATA

As respostas do sistema às velocidades angulares de referência dos motores, geradas pelos sinais de velocidade linear e angular de referência, são mostradas nas figuras 5.7 e 5.8



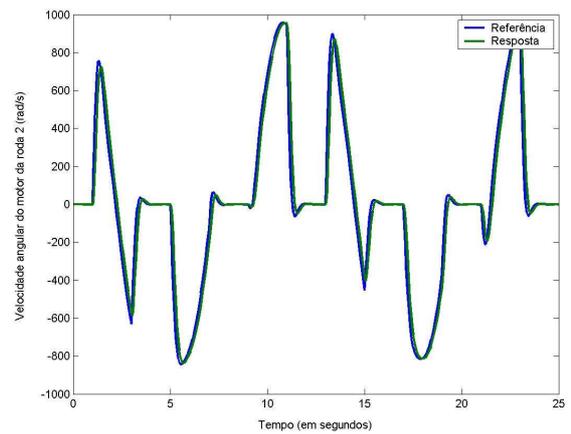
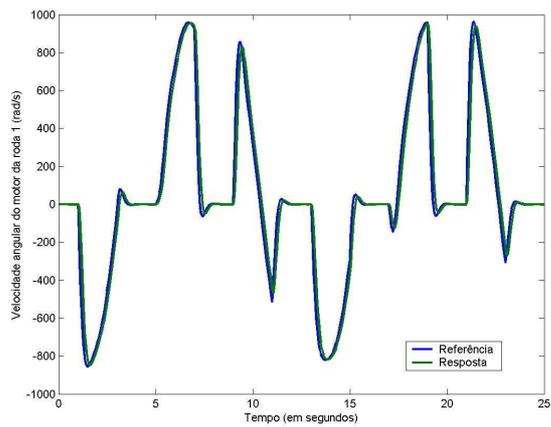
(a) Resposta à velocidade angular do motor da roda 1 (b) Resposta à velocidade angular do motor da roda 2



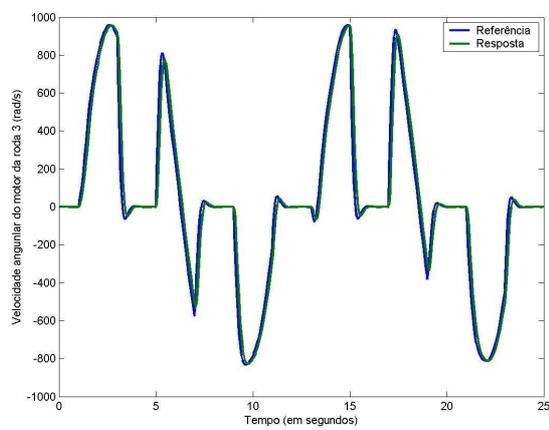
(c) Resposta à velocidade angular do motor da roda 3

**Figura 5.7.** Resposta do sistema às velocidades angulares de referência dos motores com a velocidade angular de referência do robô não nula, usando o controlador baseado na dinâmica

## 5.4 CONTROLE EM CASCATA



(a) Resposta à velocidade angular do motor da roda 1 (b) Resposta à velocidade angular do motor da roda 2

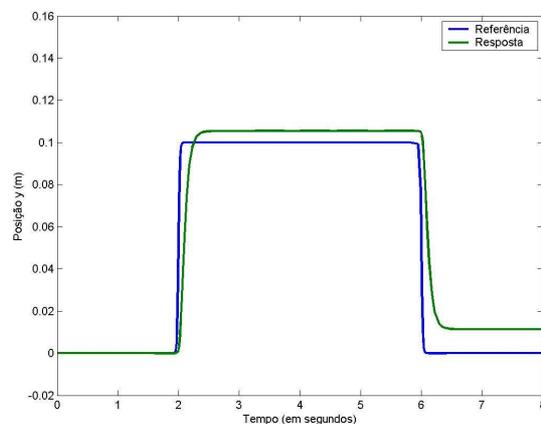
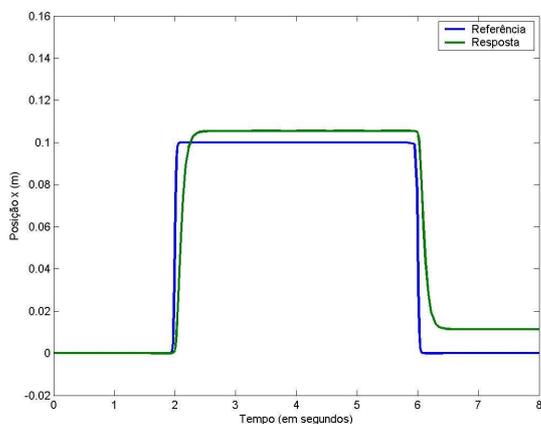


(c) Resposta à velocidade angular do motor da roda 3

**Figura 5.8.** Resposta do sistema às velocidades angulares de referência dos motores com a velocidade angular de referência do robô não nula, usando o sistema de controle em cascata

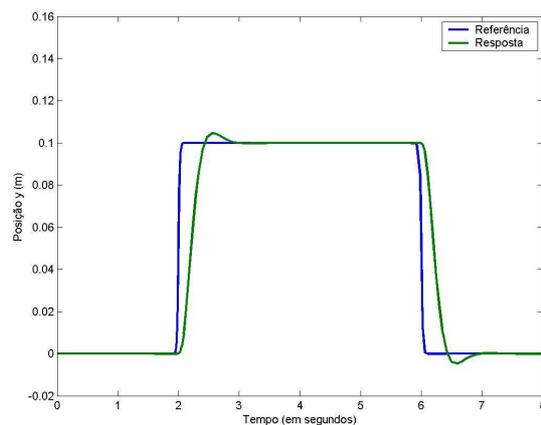
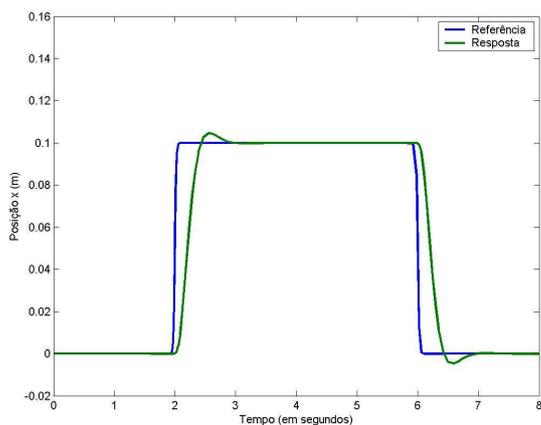
## 5.4 CONTROLE EM CASCATA

A figura 5.9 mostra as respostas do controlador baseado na dinâmica e dos controladores em cascata a mudanças bruscas de posição.



(a) Resposta à componente de posição x, usando o controle baseado na dinâmica

(b) Resposta à componente de posição y, usando o controle baseado na dinâmica



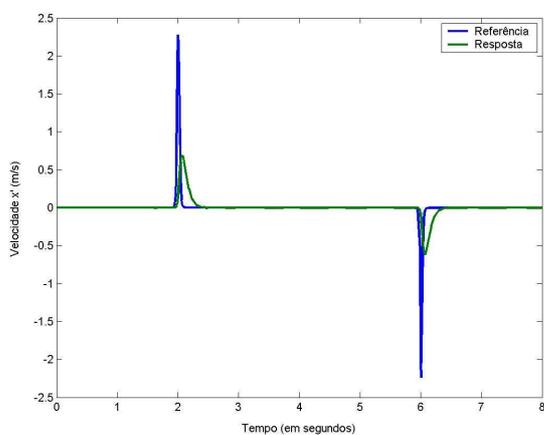
(c) Resposta à componente de posição x, usando o controle em cascata

(d) Resposta à componente de posição y, usando o controle em cascata

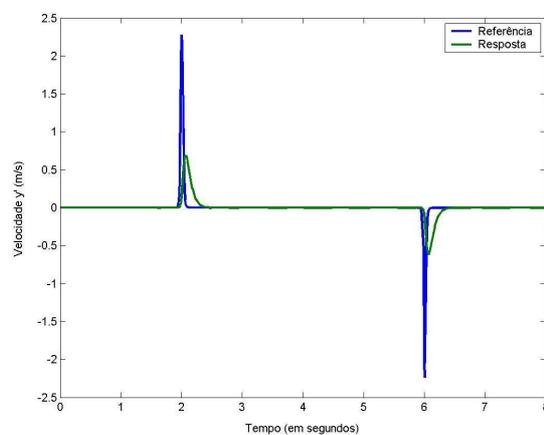
**Figura 5.9.** Resposta do sistema às mudanças bruscas de posição de referência do robô

## 5.4 CONTROLE EM CASCATA

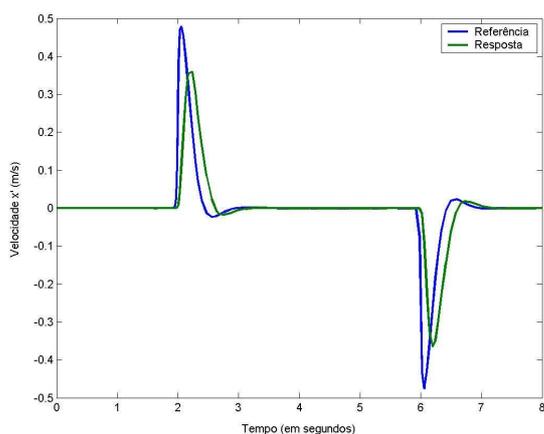
As respostas do sistema às componentes de velocidade linear de referência, geradas pelos sinais de posição de referência, são mostradas na figura 5.10.



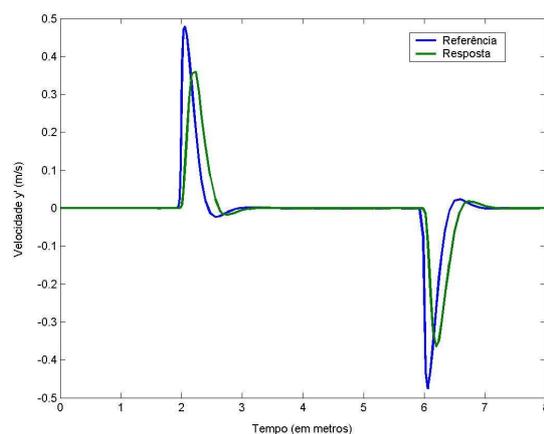
(a) Resposta à componente de velocidade x, usando o controle baseado na dinâmica



(b) Resposta à componente de velocidade y, usando o controle baseado na dinâmica



(c) Resposta à componente de velocidade x, usando o controle em cascata

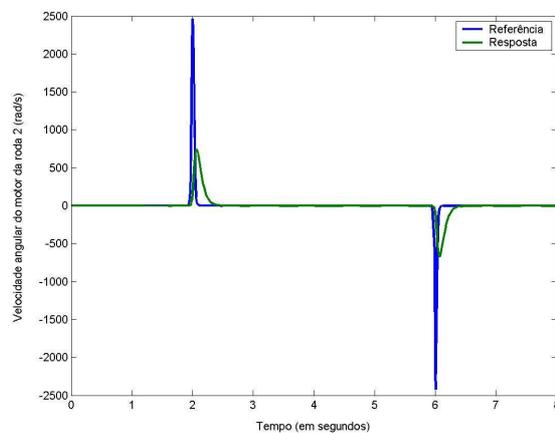
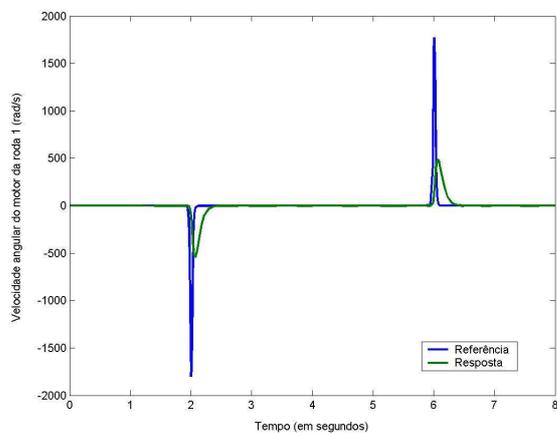


(d) Resposta à componente de velocidade y, usando o controle em cascata

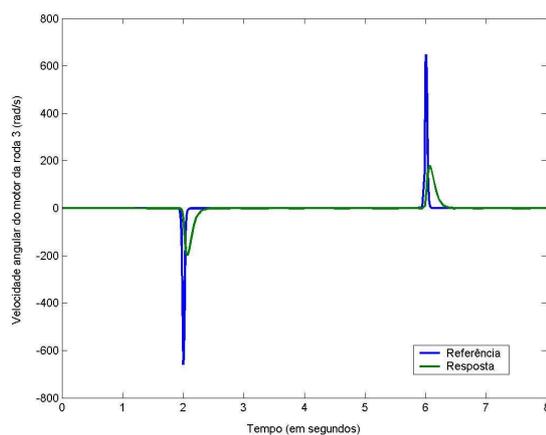
**Figura 5.10.** Resposta do sistema às componentes de velocidade linear de referência do robô, geradas por mudanças bruscas de posição

## 5.4 CONTROLE EM CASCATA

As figuras 5.11 e 5.12 mostram as respostas do sistema às velocidades angulares de referência dos motores, geradas pelos sinais de posição de referência.



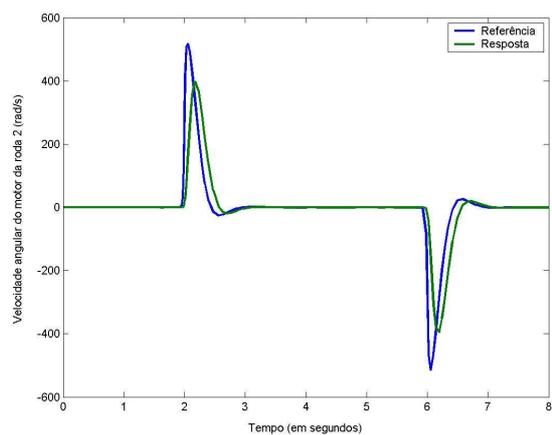
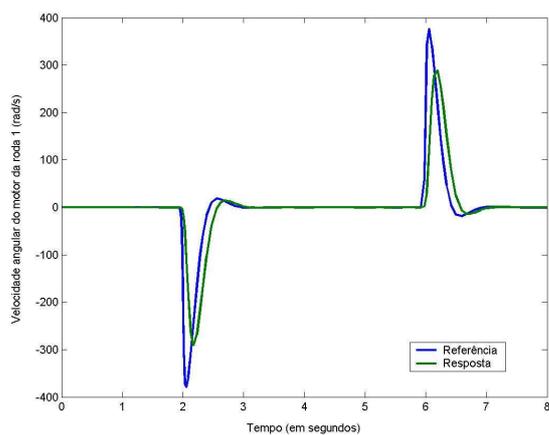
(a) Resposta à velocidade angular do motor da roda 1 (b) Resposta à velocidade angular do motor da roda 2



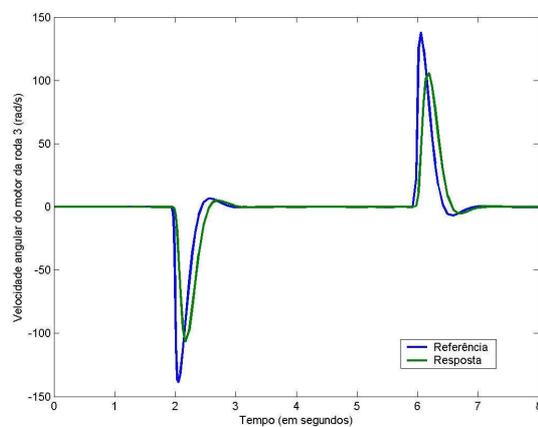
(c) Resposta à velocidade angular do motor da roda 3

**Figura 5.11.** Resposta do sistema às velocidades angulares de referência dos motores, geradas por mudanças bruscas de posição, usando o controle baseado na dinâmica.

## 5.4 CONTROLE EM CASCATA



(a) Resposta à velocidade angular do motor da roda 1 (b) Resposta à velocidade angular do motor da roda 2



(c) Resposta à velocidade angular do motor da roda 3

**Figura 5.12.** Resposta do sistema às velocidades angulares de referência dos motores, geradas por mudanças bruscas de posição, usando o controle em cascata

Nota-se, nas figuras 5.6 e 5.9, que o erro em regime estacionário é eliminado pelos controladores em cascata.

### **5.5 CONTROLE EM CASCATA COM COMPENSADORES DIFUSOS DO TIPO PI**

O controle em cascata, abordado na seção anterior, apresenta um desempenho satisfatório quando o sistema não está sujeito a ruídos e quando os parâmetros do seu modelo dinâmico não variam com o tempo, possuindo um grau elevado de certeza nos seus valores.

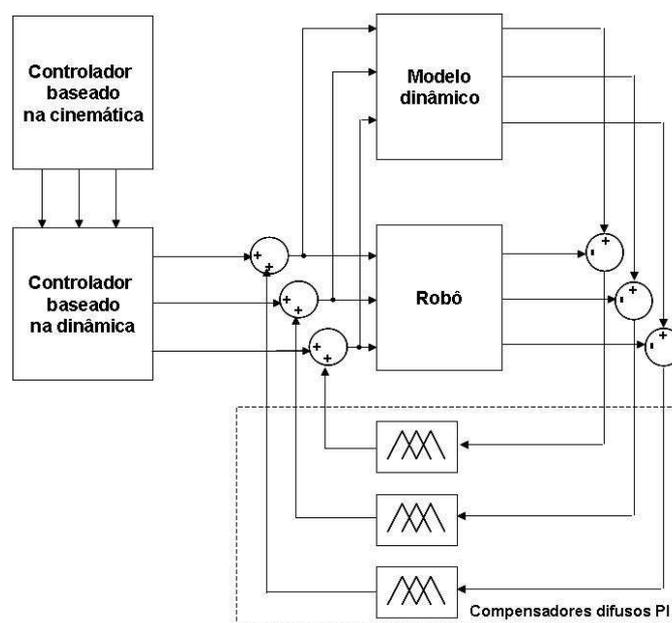
Não linearidades, como atrito estático, leitura incorreta dos sensores, variação e incertezas dos parâmetros do modelo, entre outras, às quais o robô está sujeito, degradam o desempenho dos controladores, podendo estes até se tornarem instáveis. Alguns trabalhos na literatura utilizam técnicas de controle adaptativo para minimizar os efeitos das não linearidades não modeladas do sistema (REIS, 2006; GOMES, 2006; FUKAO; NAKAGAWA; ADACHI, 2000; SUGISAKA, 2003; ALBAGUL; MARTONO, 2004).

O uso de controladores baseados em lógica difusa mostra-se eficiente devido a sua capacidade de lidar com imprecisões e não linearidades (SANDRI; CORREA, 1999). Um outra característica é a sua fácil implementação. Kodagoda *et al.*(2002) usa controladores difusos para o controle de trajetória de um veículo não-holonômico. Um controlador adaptativo baseado em lógica difusa pode ser encontrado em Sugisaka e Dai (2003).

Uma solução proposta neste trabalho, para minimizar os efeitos dos ruídos no desempenho dos controladores em cascata, foi a de utilizar compensadores de ruídos baseados em controladores difusos do tipo PI, adicionando um ganho nos sinais de controle em função do erro entre os valores de saída do sistema (velocidade angular dos motores obtidos pelos tacômetros) e os valores de saída calculados (velocidade

angular dos motores obtidos pelo modelo dinâmico).

Foram implementados três compensadores, um para cada sinal de controle. O diagrama de blocos dos controladores difusos e sua conexão ao sistema de controle em cascata é mostrado na figura 5.13



**Figura 5.13.** Diagrama de blocos dos compensadores difusos e sua conexão com os controladores.

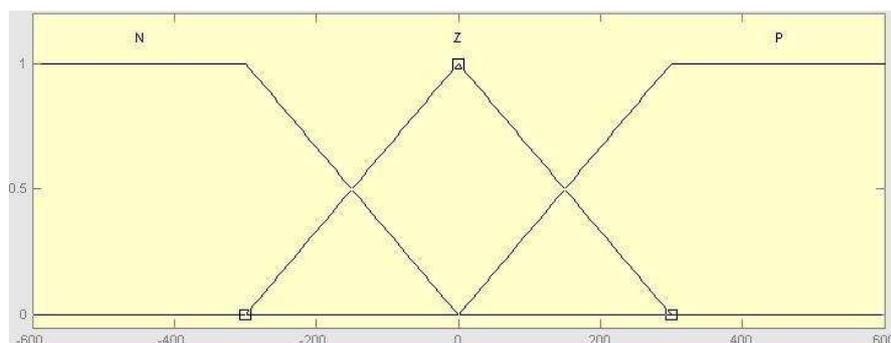
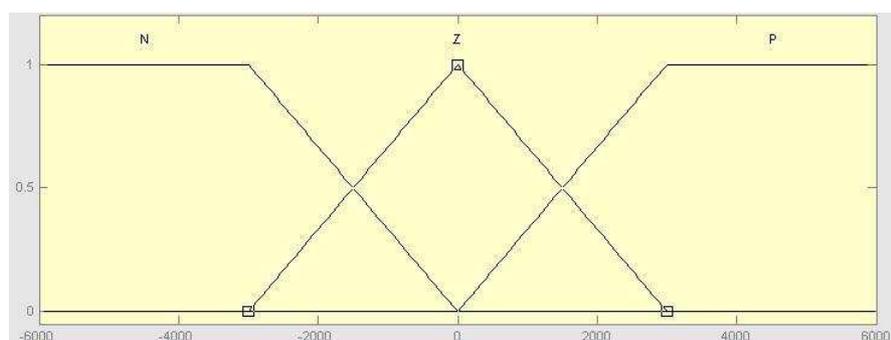
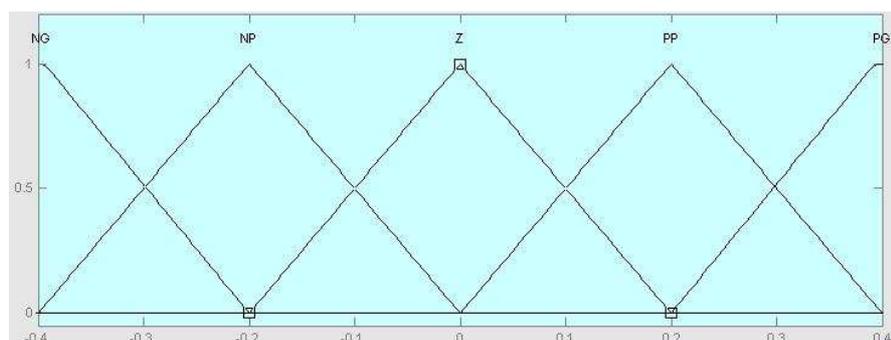
A base de regras dos compensadores é mostrada na tabela abaixo, sendo  $e$  os

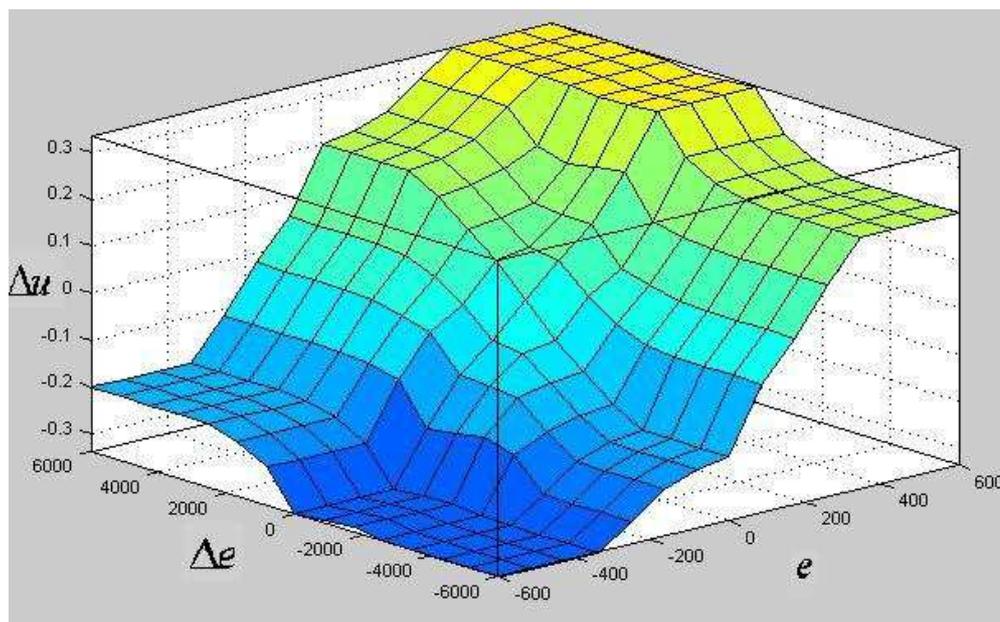
<b>Se</b> ( $e$ é $N$ ) e ( $\Delta e$ é $N$ )	<b>Então</b> $\Delta u$ é $NG$
<b>Se</b> ( $e$ é $N$ ) e ( $\Delta e$ é $Z$ )	<b>Então</b> $\Delta u$ é $NG$
<b>Se</b> ( $e$ é $N$ ) e ( $\Delta e$ é $P$ )	<b>Então</b> $\Delta u$ é $NP$
<b>Se</b> ( $e$ é $Z$ ) e ( $\Delta e$ é $N$ )	<b>Então</b> $\Delta u$ é $NP$
<b>Se</b> ( $e$ é $Z$ ) e ( $\Delta e$ é $Z$ )	<b>Então</b> $\Delta u$ é $Z$
<b>Se</b> ( $e$ é $Z$ ) e ( $\Delta e$ é $P$ )	<b>Então</b> $\Delta u$ é $PP$
<b>Se</b> ( $e$ é $P$ ) e ( $\Delta e$ é $N$ )	<b>Então</b> $\Delta u$ é $PP$
<b>Se</b> ( $e$ é $P$ ) e ( $\Delta e$ é $Z$ )	<b>Então</b> $\Delta u$ é $PG$
<b>Se</b> ( $e$ é $P$ ) e ( $\Delta e$ é $P$ )	<b>Então</b> $\Delta u$ é $PG$

**Tabela 5.2.** Base de regras dos compensadores

erros obtidos pelas diferenças entre os valores medidos e calculados, das velocidades angulares dos motores,  $\Delta e$  as derivadas dos erros e  $\Delta u$  os ganhos aplicados aos sinais

de controle. As nomenclaturas dadas as funções de pertinência  $NG$ ,  $NP$ ,  $Z$ ,  $PP$  e  $PG$ , são abreviações de *Negativo Grande*, *Negativo Pequeno*, *Zero*, *Positivo Pequeno* e *Positivo Grande*, respectivamente. As funções de pertinência das variáveis difusas são mostradas na figura 5.14 e a superfície de soluções é mostrada na figura 5.15

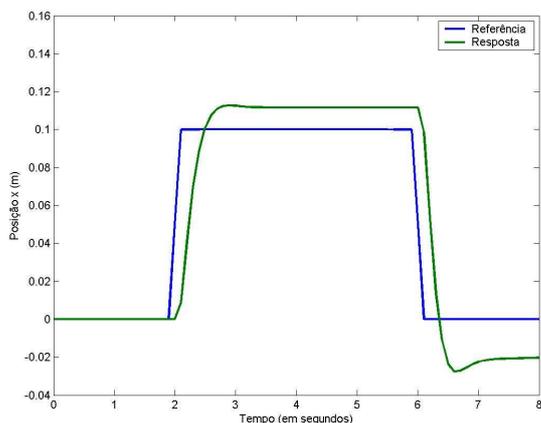
(a) Funções de pertinência da variável  $e$ (b) Funções de pertinência da variável  $\Delta e$ (c) Funções de pertinência da variável  $\Delta u$ **Figura 5.14.** Funções de pertinência das variáveis difusas



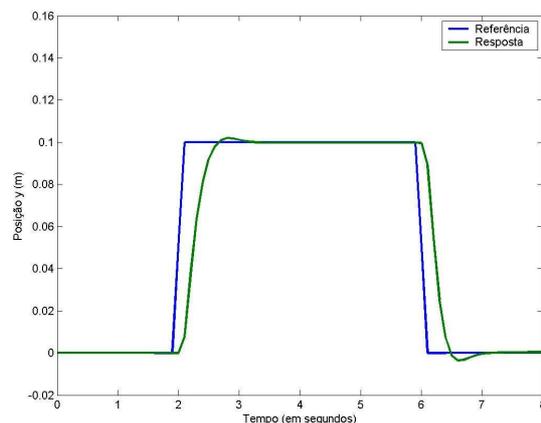
**Figura 5.15.** Superfície de soluções.

Para analisar o desempenho dos compensadores foram inicialmente inseridas, na simulação, não linearidades de entrada do tipo zona morta, sendo uma das causas deste efeito, o atrito estático. Foi utilizada uma faixa de 30% do sinal de controle, na qual o atuador não responde. Depois foram inseridas não linearidades de saída. Foi utilizado para isto um sinal de ruído senoide com uma frequência igual a 0,5 rad/s e amplitude igual a 30% dos valores de saída. Este ruído pode representar uma possível oscilação da velocidade angular da roda devido, por exemplo, a irregularidades do terreno.

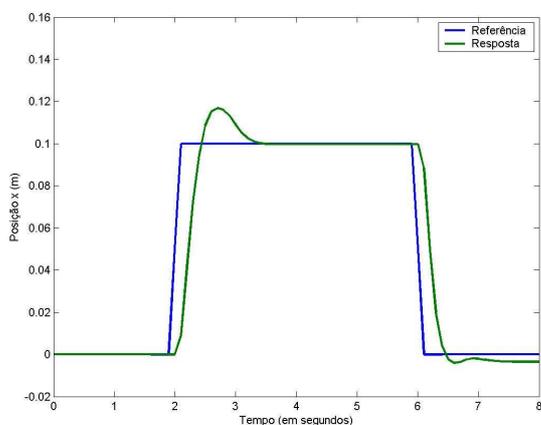
As figuras 5.16 e 5.17 mostram as respostas dos controladores em cascata, a sinais de posição e velocidade do robô, sob o efeito da zona morta, com e sem o uso dos compensadores.



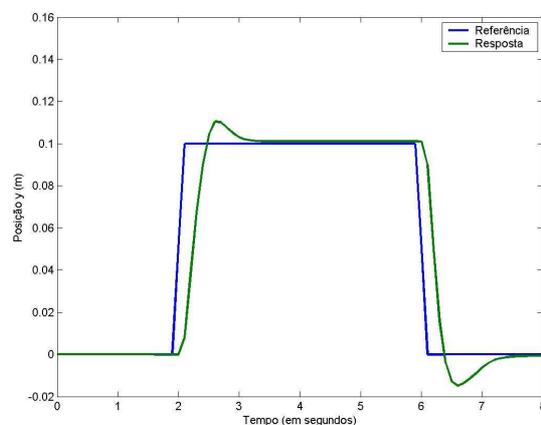
(a) Resposta à componente de posição x, sem os compensadores



(b) Resposta à componente de posição y, sem os compensadores

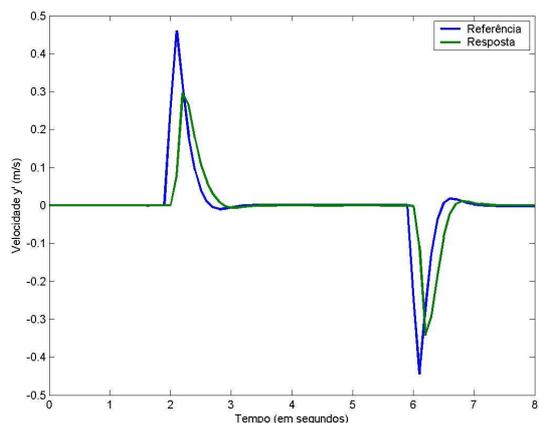
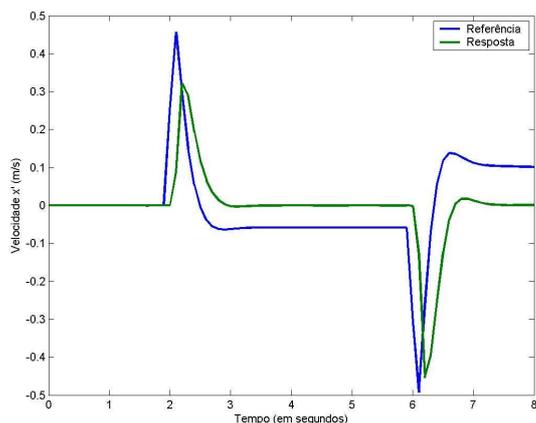


(c) Resposta à componente de posição x, com os compensadores



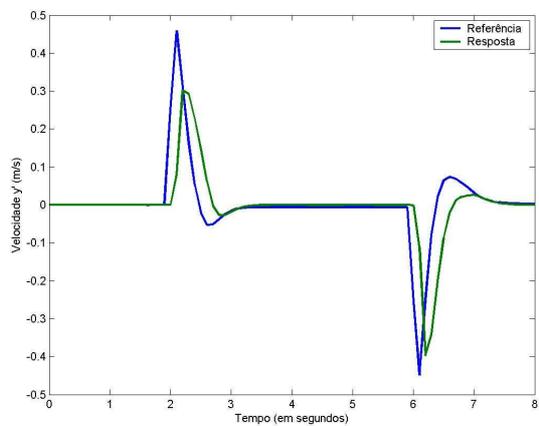
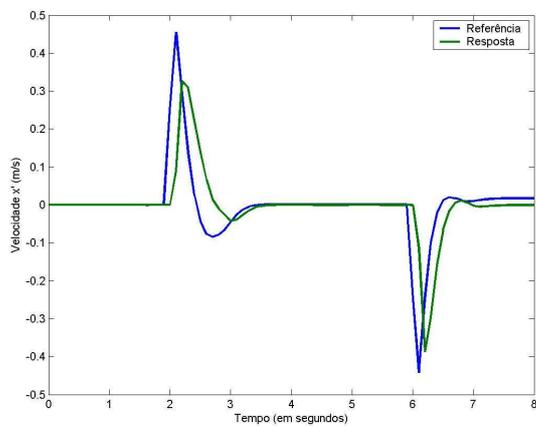
(d) Resposta à componente de posição y, com os compensadores

**Figura 5.16.** Resposta do sistema, sujeito a ruídos de entrada, à posição de referência do robô, usando os controladores com e sem os compensadores de ruídos



(a) Resposta à componente de velocidade x, sem os compensadores

(b) Resposta à componente de velocidade y, sem os compensadores



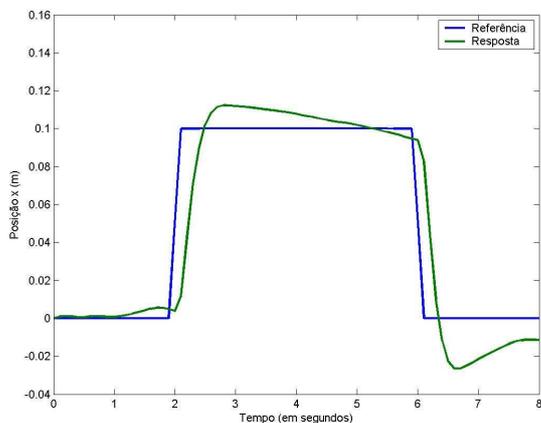
(c) Resposta à componente de velocidade x, com os compensadores

(d) Resposta à componente de velocidade y, com os compensadores

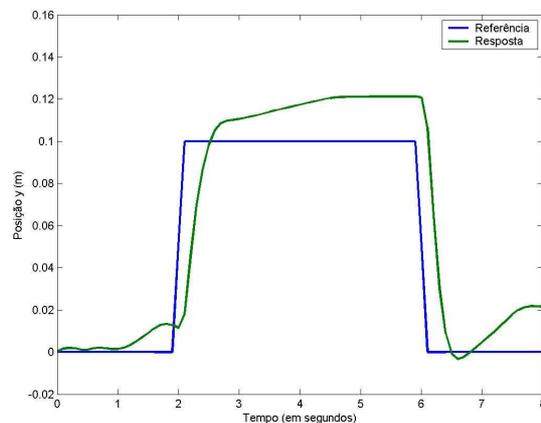
**Figura 5.17.** Resposta do sistema, sujeito a ruídos de entrada, à velocidade de referência do robô, usando os controladores com e sem os compensadores de ruídos

## 5.5 CONTROLE EM CASCATA COM COMPENSADORES DIFUSOS DO TIPO PI

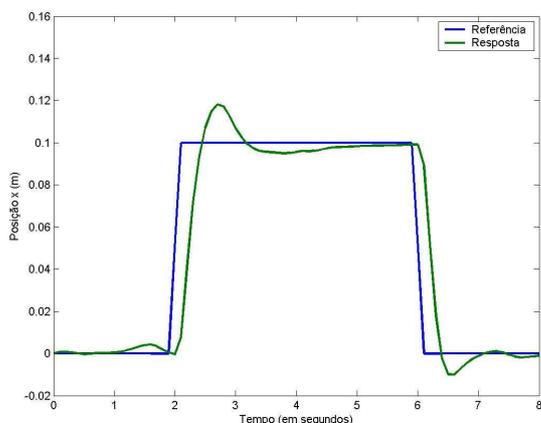
Nas figuras 5.18 e 5.19 são mostradas as respostas dos controladores em cascata, a sinais de posição e velocidade do robô, sob o efeito da zona morta e das não linearidades de saída, com e sem o uso dos compensadores.



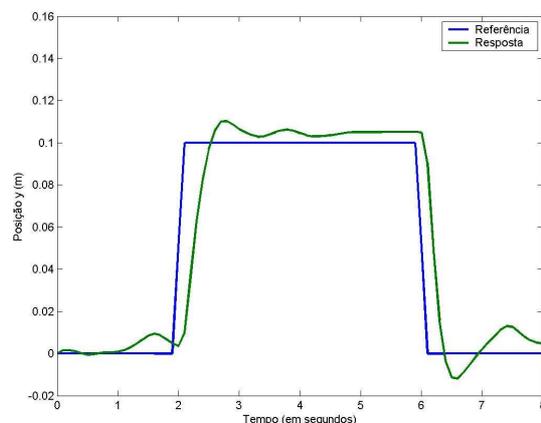
(a) Resposta à componente de posição x, sem os compensadores



(b) Resposta à componente de posição y, sem os compensadores

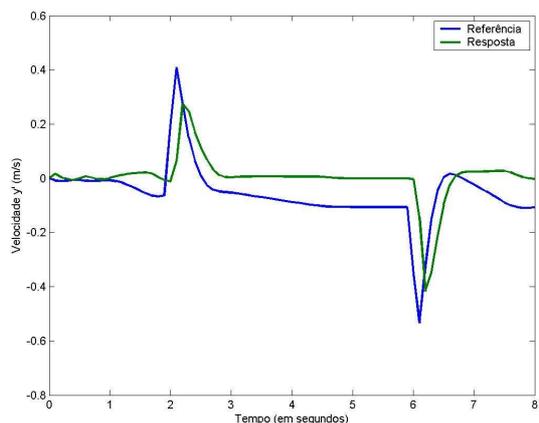
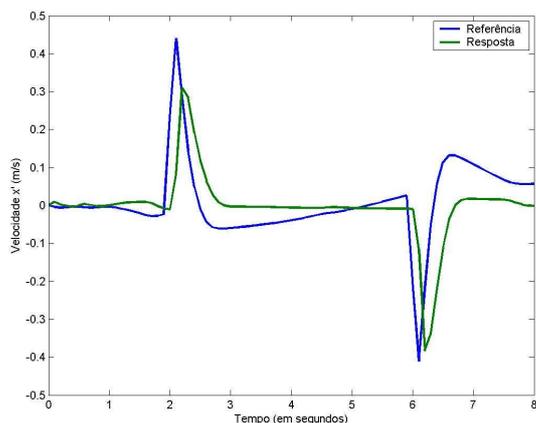


(c) Resposta à componente de posição x, com os compensadores



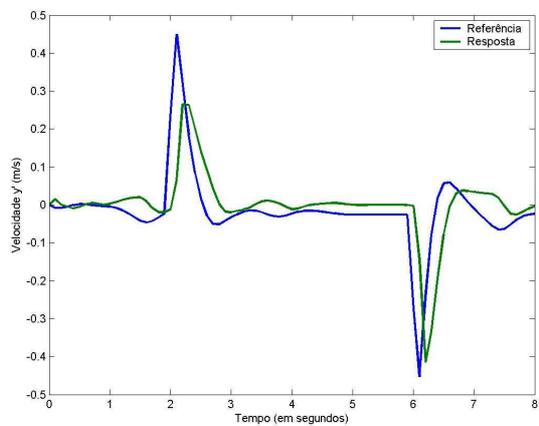
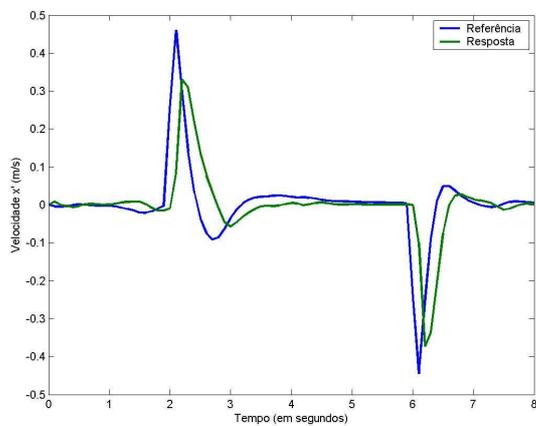
(d) Resposta à componente de posição y, com os compensadores

**Figura 5.18.** Resposta do sistema, sujeito a ruídos de entrada e saída, à posição de referência do robô, usando os controladores com e sem os compensadores de ruídos



(a) Resposta à componente de velocidade x, sem os compensadores

(b) Resposta à componente de velocidade y, sem os compensadores

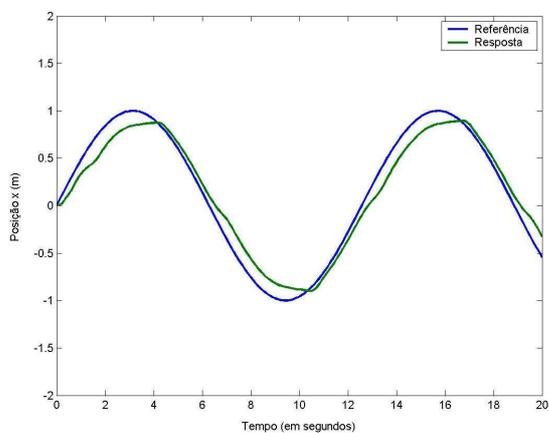


(c) Resposta à componente de velocidade x, com os compensadores

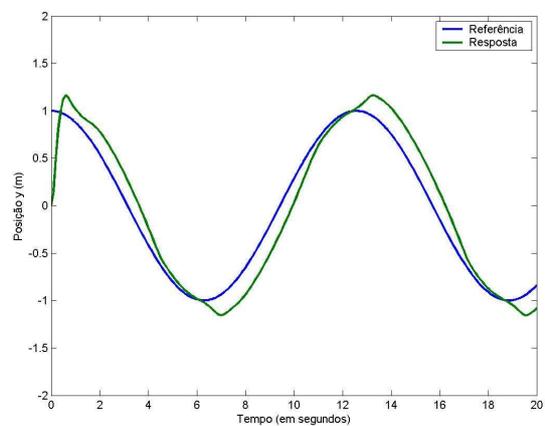
(d) Resposta à componente de velocidade y, com os compensadores

**Figura 5.19.** Resposta do sistema, sujeito a ruídos de entrada e saída, à velocidade de referência do robô, usando os controladores com e sem os compensadores de ruídos

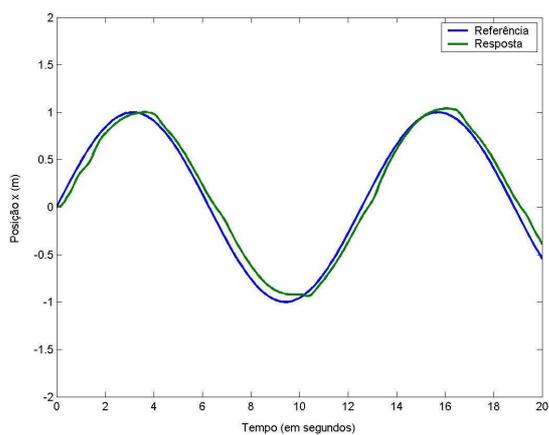
A figura 5.20 mostra a resposta do sistema a sinais de posição senoidais, sob efeito das não linearidades de entrada e saída, com e sem o uso dos compensadores.



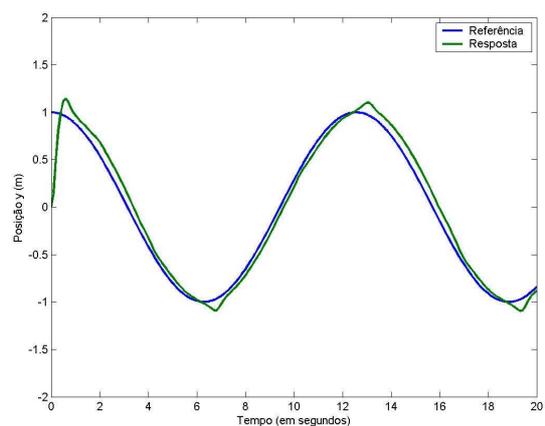
(a) Resposta à componente de posição x, sem os compensadores



(b) Resposta à componente de posição y, sem os compensadores



(c) Resposta à componente de posição x, com os compensadores



(d) Resposta à componente de posição y, com os compensadores

**Figura 5.20.** Resposta do sistema, sujeito a ruídos de entrada e saída, a sinais de posição senoidais, com e sem o uso dos compensadores de ruídos

## 5.6 ANÁLISE DE DESEMPENHO DOS CONTROLADORES

Para a análise de desempenho dos controladores foi utilizado como parâmetro a integral do quadrado do erro. A análise foi feita tendo o sistema sob as mesmas condições apresentadas nas seções anteriores. A equação 5.6 representa o valor do índice utilizado como parâmetro de análise.

$$I = \int_{t=0}^N \mathbf{e}^T \mathbf{e} dt \quad (5.6)$$

sendo  $I$  o valor do índice,  $N$  o período de integração e  $\mathbf{e}$  o vetor de erro.

As tabela 5.3 mostra os valores dos índices calculados para as respostas do sistema sob as condições e sinais de referência mostradas nas seções anteriores. As condições são listadas abaixo:

1. Velocidade linear de referência do robô não nula e angular nula.
2. Velocidade linear e angular de referência do robô não nula.
3. Mudanças bruscas no sinal de posição de referência.
4. Sistema sob condições de ruído de entrada.
5. Sistema sob condições de ruído de entrada e saída.
6. Sistema sob condições de ruído de entrada e saída com sinais senoidais de entrada.

Índice	Período	Referência	Controle	Condição
18, 2896	25 s	Velocidade do robô	Dinâmico	1
19, 4686	25 s	Velocidade do robô	Dinâmico	2
18, 5795	25 s	Velocidade do robô	Cascata	2
0, 1504	8 s	Posição	Dinâmico	3
50, 4257	8 s	Velocidade do robô	Dinâmico	3

## 5.6 ANÁLISE DE DESEMPENHO DOS CONTROLADORES

---

0,1425	8 s	Posição	Cascata	3
2,3570	8 s	Velocidade do robô	Cascata	3
0,1731	8 s	Posição	Cascata	4
1,8716	8 s	Velocidade do robô	Cascata	4
0,1597	8 s	Posição	Cascata com compensadores	4
1,3465	8 s	Velocidade do robô	Cascata com compensadores	4
0,2015	8 s	Posição	Cascata	5
1,9103	8 s	Velocidade do robô	Cascata	5
0,1743	8 s	Posição	Cascata com compensadores	5
1,5926	8 s	Velocidade do robô	Cascata com compensadores	5
1,8824	100 s	Posição	Cascata	6
0,8965	100 s	Posição	Cascata com compensadores	6

Tabela 5.3: Análise dos controladores sob determinadas condições

De acordo com a tabela 5.3 pode-se concluir que:

- Na situação 2 e 3, houve uma queda do índice do controlador em cascata em relação ao controlador dinâmico, significando uma melhora no desempenho.
- O controle em cascata com os compensadores teve um melhor desempenho em relação ao controle em cascata sem os compensadores, nas situações 4, 5 e 6.
- Na situação 6, uma trajetória circular de referência é rastreada com um desempenho 53% melhor, quando usados os compensadores no controle em cascata.

### 5.7 CONCLUSÃO

A aplicação de controladores baseados na cinemática e na dinâmica do AxeBot e compensadores de ruídos baseados em controladores difusos foram apresentados neste capítulo. Foi mostrado que controladores baseados apenas na dinâmica podem, em algumas situações, gerar erros em regime estacionário da postura de referência do robô, necessitando fechar a malha da postura através de um controlador baseado na cinemática. Foi analisado o desempenho dos controladores a sinais de posição e velocidade, utilizando o índice de desempenho baseado na integral do quadrado do erro. Esta análise mostrou a eficiência da utilização dos compensadores de ruídos nos controladores em cascata.

# MÓDULO DE GERAÇÃO E CONTROLE DE TRAJETÓRIA PROPOSTO PARA O ROBÔ AXEBOT, SUA SIMULAÇÃO E RESULTADOS

## 6.1 INTRODUÇÃO

Este capítulo apresenta o módulo de geração e controle de trajetória proposto, mostrando sua simulação e resultados, através da resposta dos controladores à trajetórias de referência, geradas pelo gerador de trajetória.

O módulo de geração e controle de trajetória foi simulado no Matlab *Simulink*<sup>®</sup>. Os algoritmos de geração de trajetória foram desenvolvidos na linguagem nativa do Matlab, enquanto que as equações do modelo dinâmico e cinemático e o controlador de trajetória foram construídos no *Simulink*. Foi inserido na simulação um bloco de realidade virtual no intuito de observar o comportamento do robô através de uma animação em três dimensões.

A seção 6.2 dá uma visão geral do módulo de geração e controle de trajetória proposto. A seção 6.3 mostra os tipos de trajetórias geradas pelo gerador de trajetória e o seu funcionamento. A seção 6.4 descreve o controlador de trajetória proposto e a seção 6.5 mostra a simulação e os resultados obtidos, tendo algumas trajetórias como referência.

## 6.2 ESTRUTURA DO MÓDULO DE GERAÇÃO E CONTROLE DE TRAJETÓRIA

A estrutura do módulo de geração e controle de trajetória proposto para o robô Axebot é mostrado na figura 6.1

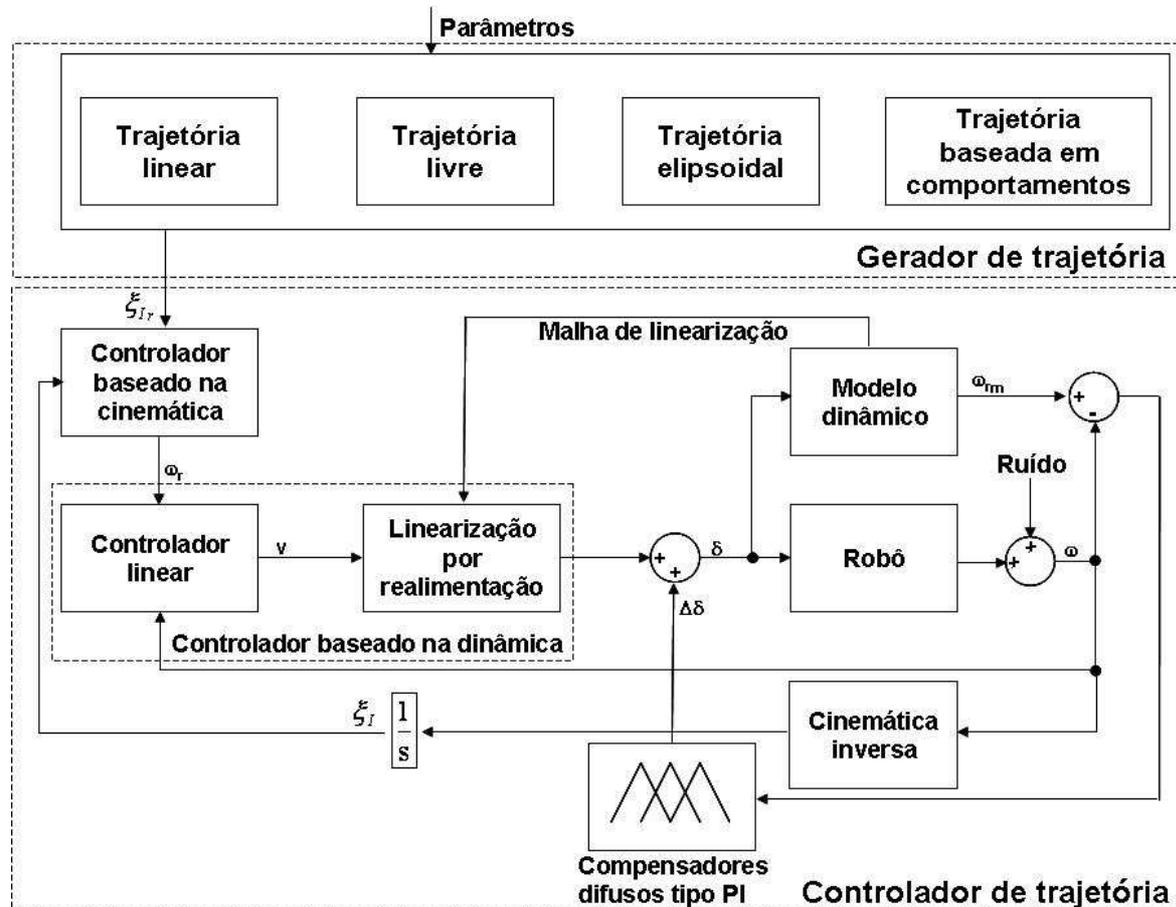


Figura 6.1. Estrutura do gerador e controlador de trajetória.

O gerador de trajetória está configurado para receber parâmetros do planejamento de trajetória. As suas saídas são as componentes de posições de referência no plano ( $\xi_{I_r}$ ), sendo as entradas para o controle de trajetória. O controle de trajetória é composto por controladores em cascata, com compensadores de ruídos, como mostrados nas seções 5.4 e 5.5.

### 6.3 GERADOR DE TRAJETÓRIA PROPOSTO

Pode-se destacar alguns trabalhos, encontrados na literatura, sobre técnicas de geração de trajetória, como: Andrade *et al.*(2001), Garrido *et al.*(2002) e Almeida (2005), que usam técnicas de inteligência artificial para a geração de trajetórias; Ramos (1986) que utiliza técnicas de geração de trajetórias contínuas; Dias *et al.*(2000), onde é desenvolvida uma metodologia para a geração de trajetórias de manipuladores; Paromtchik e Rembold (1994) e Paromtchik e Asama (2000), onde é utilizada a interpolação polinomial, por funções cúbicas *spline*.

Existem basicamente dois tipos de técnicas de geração de trajetórias, como mostrado na literatura. Um deles é por funções contínuas, o outro por técnicas de interpolação polinomial. A Geração de trajetórias por funções contínuas é simples de ser planejada, devido a necessidade de um conjunto reduzido de parâmetros para a sua geração, porém necessita que a forma da trajetória tenha um comportamento conhecido. A geração por interpolação é mais difícil de ser planejada, pois necessita de todos os pontos por onde a função interpoladora tem que passar. Quanto mais precisa e maior for a trajetória, maior será o número de parâmetros a ser gerado pelo planejamento. A geração por esta técnica, porém, gera trajetórias com qualquer formato.

O gerador de trajetória proposto utiliza uma técnica híbrida de geração de trajetórias através de funções contínuas, denominadas de comportamentos, e uma técnica de interpolação por funções *spline*, eliminando as mudanças bruscas de posição na interseção de dois comportamentos. Cada comportamento define um trecho de trajetória. As trajetórias divididas em comportamentos requerem um menor número de parâmetros para o seu planejamento, simplificando o interfaceamento entre o módulo de geração e controle de trajetória e o planejamento de trajetória. Trajetórias por técnicas de interpolação, do tipo linear, elipsoidal e de caminho livre, também são geradas pelo gerador de trajetória.

O gerador de trajetória proposto gera quatro tipos de trajetórias no plano: Tra-

jetórias do tipo linear, trajetórias de caminho livre, trajetórias elipsoidais e trajetórias baseadas em comportamentos elipsoidais e lineares.

No capítulo 3 foram mostradas técnicas de geração de polinômios entre dois ou mais pontos com condições de continuidade nos polinômios e em algumas de suas derivadas. Foi mostrado que essas condições são fundamentais para que não haja mudanças bruscas de velocidade e aceleração do robô, durante o rastreamento da trajetória.

Nesta seção é apresentado o uso de técnicas abordadas no capítulo 3, para a construção dos tipos de trajetórias geradas pelo gerador de trajetória proposto.

### 6.3.1 Trajetórias lineares

Para gerar uma trajetória plana é necessário definir as ordenadas e as abscissas dos pontos sobre os quais os polinômios vão interpolar e o tempo de execução da trajetória. São necessários, portanto, dois conjuntos de polinômios interpoladores, onde um deles define a evolução do percurso no tempo, em relação ao eixo das ordenadas, enquanto que o outro é responsável pela evolução no eixo das abscissas.

Para a geração de trajetórias lineares foi usada a técnica de interpolação por polinômios de grau cinco. Os dois polinômios necessários para a geração da trajetória são regidos pela equação 3.1, sendo  $r(\tau)$  representada pela equação 3.14. Esta técnica garante que a velocidade e a aceleração do robô sejam nulas no início e no final do percurso, garantindo a continuidade da velocidade e da aceleração na interseção de duas trajetórias. É importante notar que, apesar da geometria do caminho ser linear, o comportamento, no tempo, dos polinômios que formam a trajetória, é não-linear.

O exemplo a seguir mostra a geração de duas trajetórias lineares usando a técnica de interpolação por polinômios de grau cinco. Para a primeira trajetória foi considerado o ponto  $P(1;0)$ , como ponto inicial e o ponto  $P(3;5)$ , como ponto final. Na segunda, os pontos  $P(3;5)$  e  $P(4;3)$ , foram considerados como ponto inicial e final,

respectivamente.

De acordo com a equação 3.1 e com a equação 3.14, considerando o tempo de execução da trajetória igual a 50 segundos (valor de  $t_f = 50$ ), são formulados os seguintes polinômios

$$s_{1_x}(t) = 1 + 20\left(\frac{t}{50}\right)^3 - 30\left(\frac{t}{50}\right)^4 + 12\left(\frac{t}{50}\right)^5 \quad (6.1)$$

$$s_{1_y}(t) = 50\left(\frac{t}{50}\right)^3 - 75\left(\frac{t}{50}\right)^4 + 30\left(\frac{t}{50}\right)^5 \quad (6.2)$$

$$s_{2_x}(t) = 3 + 10\left(\frac{t}{50}\right)^3 - 15\left(\frac{t}{50}\right)^4 + 6\left(\frac{t}{50}\right)^5 \quad (6.3)$$

$$s_{2_y}(t) = 5 - 20\left(\frac{t}{50}\right)^3 + 30\left(\frac{t}{50}\right)^4 - 12\left(\frac{t}{50}\right)^5 \quad (6.4)$$

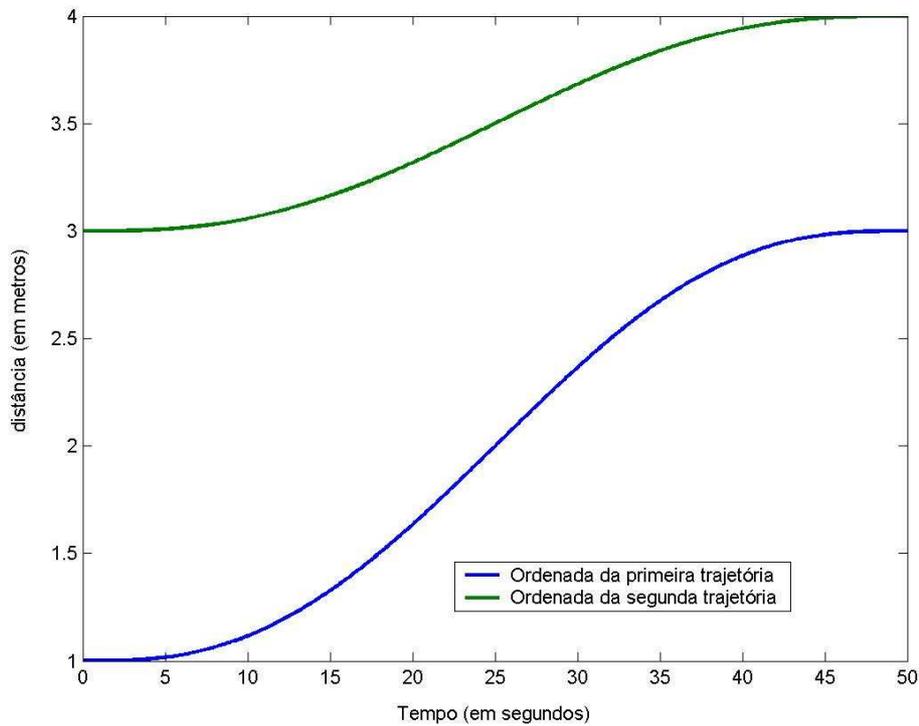
sendo  $s_{1_x}(t)$ ,  $s_{1_y}(t)$ ,  $s_{2_x}(t)$  e  $s_{2_y}(t)$  os polinômios referentes aos eixos das ordenadas e abscissas da primeira e segunda trajetória, respectivamente.

A figura 6.2 e 6.3 mostram, respectivamente, a evolução das duas trajetórias no tempo, nos eixos das ordenadas e das abscissas. A figura 6.4 mostra o percurso feito pelas duas trajetórias.

### 6.3.2 Trajetórias de caminho livre

Para a geração de trajetórias de caminho livre foi usada a técnica de interpolação polinomial por funções cúbicas *spline* mostradas na seção 3.2.3.

O conjunto de polinômios cúbicos  $s_k(t)$ , da equação 3.20, representam o comportamento da trajetória em uma das coordenadas de um sistema cartesiano, em relação ao tempo. Pode-se dizer que os polinômios  $s_k(t)$  mostram a evolução da projeção do robô sobre um dos eixos de coordenadas, durante o seu percurso. São necessários, portanto, dois conjuntos de polinômios interpoladores, ou simplesmente duas funções *spline* para a geração da trajetória.

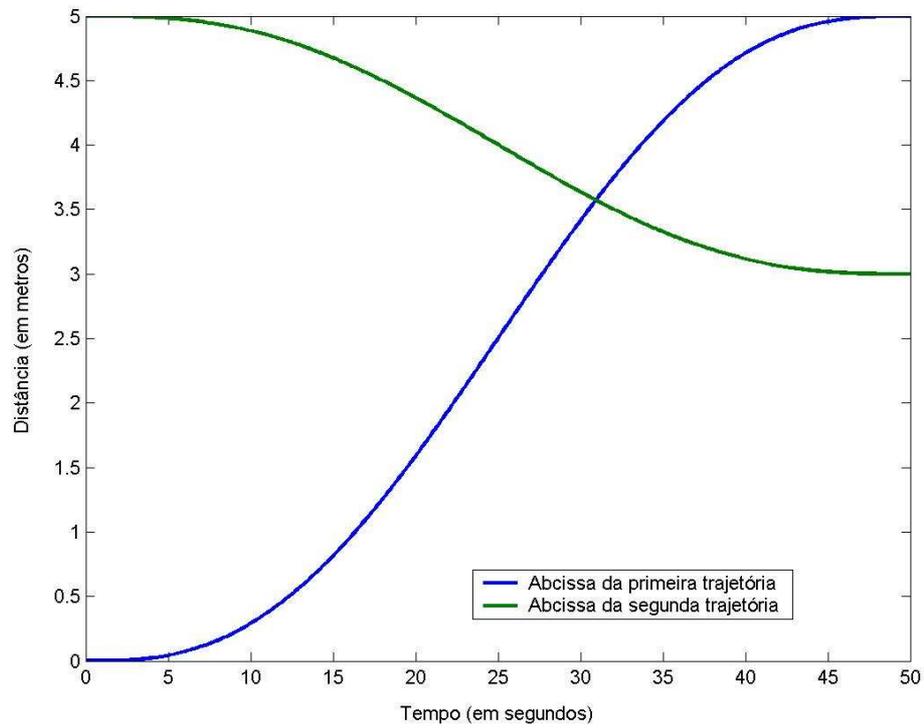


**Figura 6.2.** Comportamento das trajetórias no eixo das ordenadas.

O exemplo a seguir mostra a geração de uma trajetória plana pelo método de funções cúbicas *spline*.

Os pontos considerados para a geração da trajetória foram:  $P_1(1;3)$ ,  $P_2(2;2)$ ,  $P_3(2;4)$ ,  $P_4(0;6)$ ,  $P_5(3;8)$ ,  $P_6(3;6)$ ,  $P_7(6;5)$ ,  $P_8(7;2)$ ,  $P_9(5;3)$ ,  $P_{10}(5,5;4)$  e  $P_{11}(4;3)$ . O valor de  $t_f$  considerado foi 100. As expressões 6.5 e 6.6 mostram, respectivamente, os valores das matrizes  $\mathbf{A}$  e  $\mathbf{C}$ , calculadas de acordo com os dados acima e de acordo

### 6.3 GERADOR DE TRAJETÓRIA PROPOSTO



**Figura 6.3.** Comportamento das trajetórias no eixo das abscissas.

com as expressões 3.37 e 3.36.

$$\mathbf{A} = \begin{bmatrix}
 40 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 10 & 40 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 10 & 40 & 10 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 10 & 40 & 10 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 10 & 40 & 10 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 10 & 40 & 10 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 10 & 40 & 10 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 10 & 40 & 10 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 40
 \end{bmatrix} \quad (6.5)$$

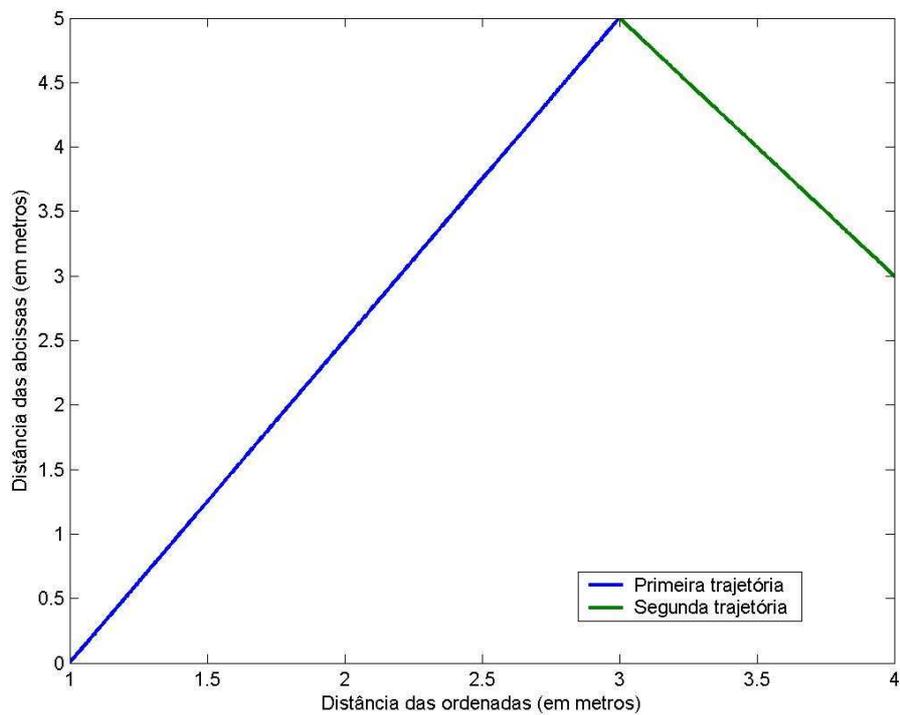


Figura 6.4. Primeira e segunda trajetória.

$$\mathbf{C} = \begin{bmatrix} 0.1 & -0.2 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & -0.2 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & -0.2 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & -0.2 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & -0.2 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & -0.2 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & -0.2 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & -0.2 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & -0.2 & 0.1 \end{bmatrix} \quad (6.6)$$

Os valores dos coeficientes  $A_k$ ,  $B_k$ ,  $C_k$  e  $D_k$ , de acordo com as equações 3.28, 3.23, 3.30 e 3.25, para o conjunto de polinômios que interpolam as ordenadas dos pontos, são representados respectivamente pelos vetores  $\mathbf{a}_x$ ,  $\mathbf{b}_x$ ,  $\mathbf{c}_x$  e  $\mathbf{d}_x$ , mostrados

nas expressões 6.7, 6.8, 6.9 e 6.10. Os valores da segunda derivada deste conjunto de polinômios, calculados através da equação 3.34 são representados pelo vetor  $\ddot{\mathbf{s}}_{\mathbf{x}}$ , mostrado na expressão 6.11.

$$\mathbf{a}_{\mathbf{x}} = \begin{bmatrix} -0.0019 & 0.0024 & 0.0008 & -0.0011 & 0.0007 & -0.0017 & 0.0021 \\ -0.0015 & -0.0002 & 0.0004 & & & & \end{bmatrix} \quad (6.7)$$

$$\mathbf{b}_{\mathbf{x}} = \begin{bmatrix} 0 & 0.0248 & -0.0092 & 0.0121 & -0.0393 & 0.0251 & -0.0311 \\ 0.0394 & -0.0065 & -0.0134 & & & & \end{bmatrix} \quad (6.8)$$

$$\mathbf{c}_{\mathbf{x}} = \begin{bmatrix} -0.1827 & 0.0654 & 0.2211 & 0.2501 & -0.0216 & -0.1636 & -0.2238 \\ -0.1410 & 0.1880 & -0.0108 & & & & \end{bmatrix} \quad (6.9)$$

$$\mathbf{d}_{\mathbf{x}} = \begin{bmatrix} 3 & 2 & 4 & 6 & 8 & 6 & 5 & 2 & 3 & 4 \end{bmatrix} \quad (6.10)$$

$$\ddot{\mathbf{s}}_{\mathbf{x}} = \begin{bmatrix} -0 & 0.0005 & 0.0581 & 0.1127 & -0.0929 & -0.0789 & -0.0427 \\ -0.0282 & 0.0055 & -0.0061 & 0 & & & \end{bmatrix} \quad (6.11)$$

A figura 6.5 mostra a função *spline* que interpola as ordenadas dos pontos.

Os valores dos coeficientes  $A_k$ ,  $B_k$ ,  $C_k$  e  $D_k$ , para o conjunto de polinômios que interpolam a abscissa dos pontos são representados respectivamente pelos vetores  $\mathbf{a}_{\mathbf{y}}$ ,  $\mathbf{b}_{\mathbf{y}}$ ,  $\mathbf{c}_{\mathbf{y}}$  e  $\mathbf{d}_{\mathbf{y}}$ , mostrados nas expressões 6.12, 6.13, 6.14 e 6.15. Os valores da segunda derivada deste conjunto de polinômios são representados pelo vetor  $\ddot{\mathbf{s}}_{\mathbf{y}}$ , mostrado na expressão 6.16.

$$\mathbf{a}_{\mathbf{y}} = \begin{bmatrix} -0 & -0.0010 & 0.0028 & -0.0034 & 0.0029 & -0.0020 & 0.0002 \\ 0.0006 & 0.0000 & -0.0001 & & & & \end{bmatrix} \quad (6.12)$$

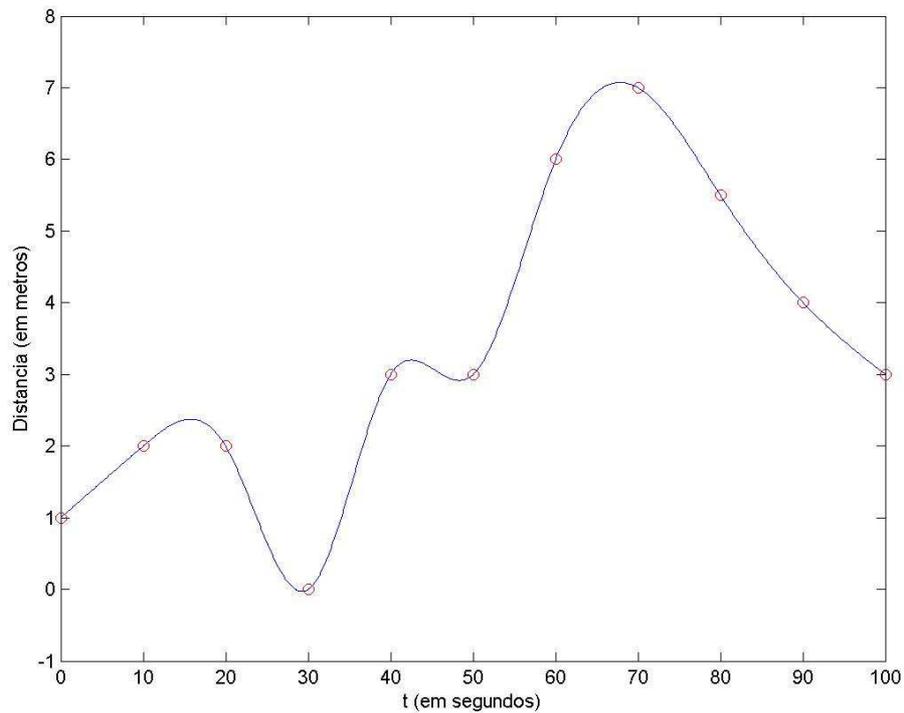


Figura 6.5. Função *spline* cúbica natural interpolando a ordenada dos onze pontos.

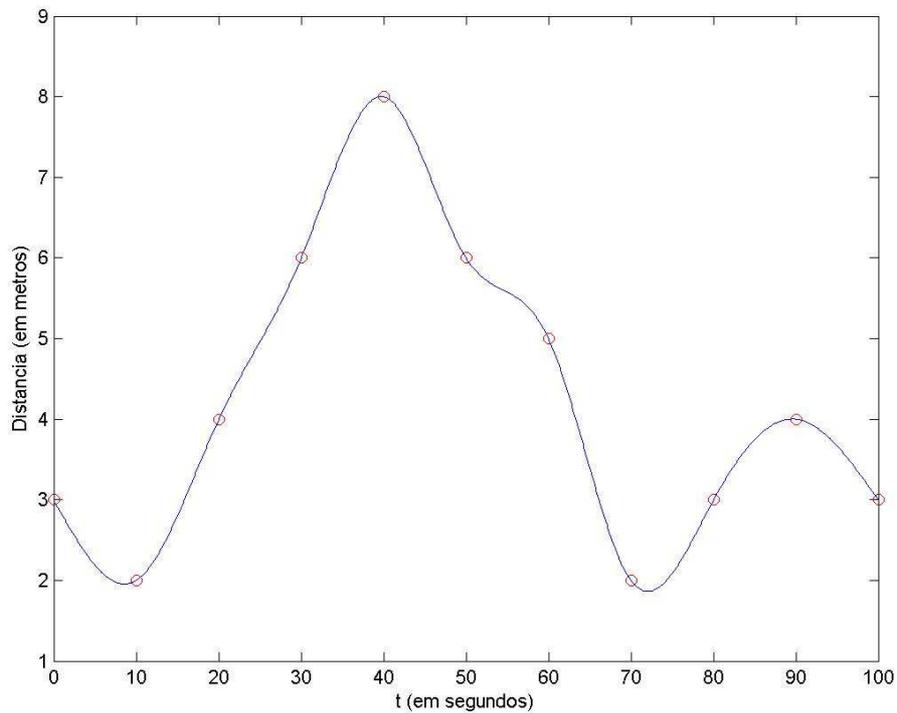
$$\mathbf{b}_y = \begin{bmatrix} 0 & -0.0002 & -0.0290 & 0.0564 & -0.0465 & 0.0394 & -0.0213 \\ -0.0141 & 0.0028 & 0.0031 & & & & \end{bmatrix} \quad (6.13)$$

$$\mathbf{c}_y = \begin{bmatrix} 0.1008 & 0.0984 & -0.1944 & 0.0790 & 0.1782 & 0.1081 & 0.2893 \\ -0.0652 & -0.1786 & -0.1204 & & & & \end{bmatrix} \quad (6.14)$$

$$\mathbf{d}_y = \begin{bmatrix} 1 & 2 & 2 & 0 & 3 & 3 & 6 & 7 & 5.5 & 4 \end{bmatrix} \quad (6.15)$$

$$\ddot{\mathbf{s}}_y = \begin{bmatrix} -0 & 0.0496 & 0.0185 & 0.0243 & -0.0786 & -0.0502 & -0.0623 \\ -0.0788 & 0.0130 & -0.0267 & 0 & & & \end{bmatrix} \quad (6.16)$$

A figura 6.6 mostra a função *spline* que interpola as abscissas dos pontos.



**Figura 6.6.** Função *spline* cúbica natural interpolando a abscissa dos onze pontos.

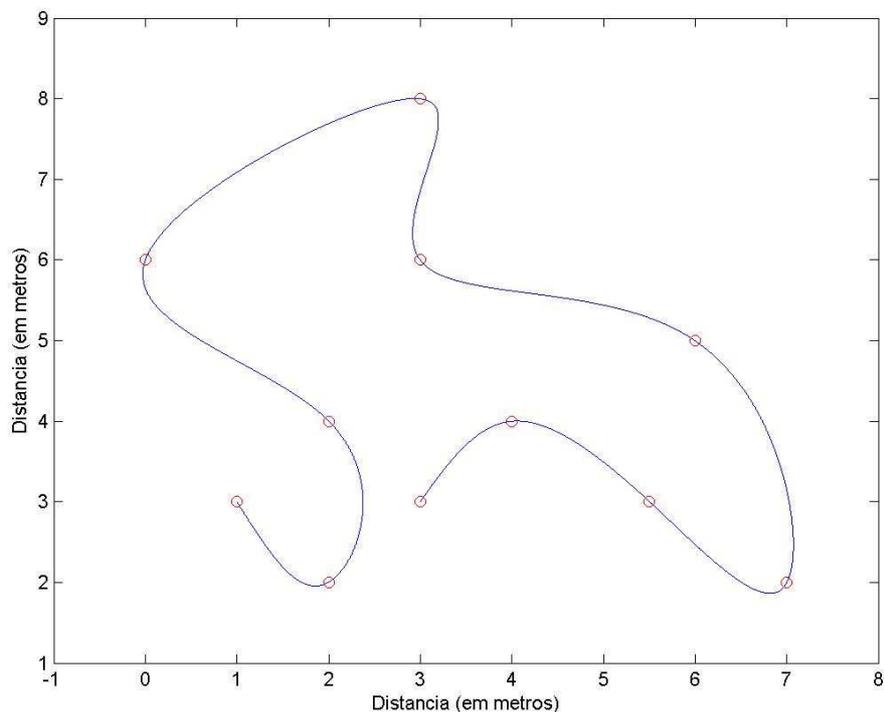
Unindo a ordenada e a abscissa dos pontos é formada a trajetória plana desejada. Esta trajetória é mostrada na figura 6.7

### 6.3.3 Trajetórias elipsoidais

Trajetoórias em forma de elipse são alternativas de trajetórias planas e podem ser geradas por um conjunto reduzido de parâmetros. A geração de uma trajetória elipsoidal através de uma função *spline* não é uma alternativa atraente, pois é necessário conhecer as coordenadas de alguns pontos sobre a elipse a ser gerada.

Uma trajetória elipsoidal completa ou uma trajetória em forma de arco de elipse pode ser construída através das equações:

$$x(t) = a \cos(\Delta\alpha r(\tau) + \alpha) \quad (6.17)$$



**Figura 6.7.** Trajetória plana gerada pelos onze pontos.

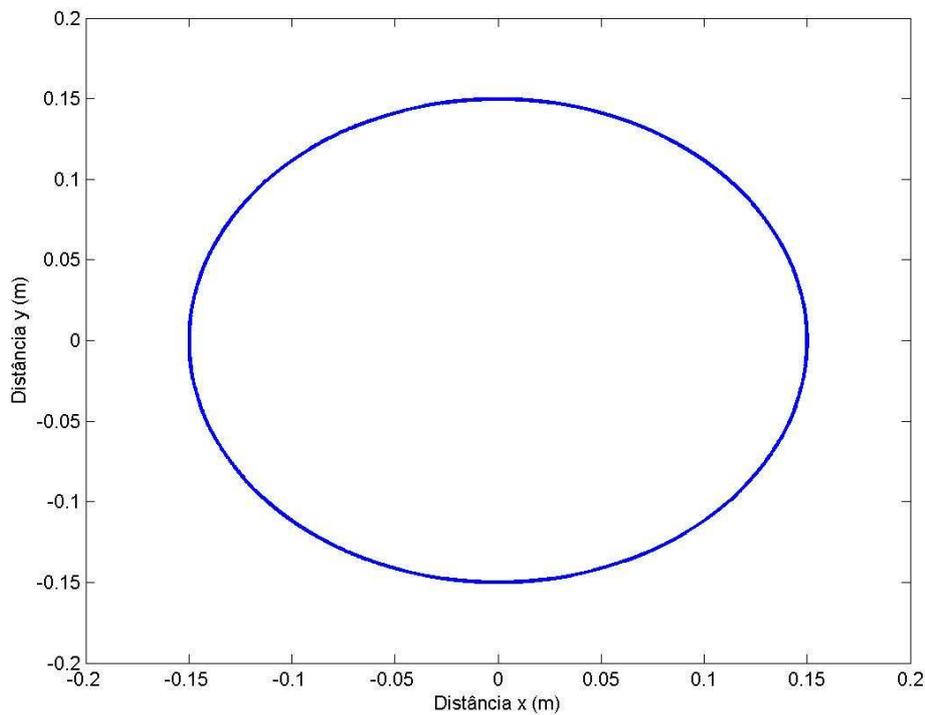
$$y(t) = b \sin(\Delta\alpha r(\tau) + \alpha) \quad (6.18)$$

onde  $x(t)$  e  $y(t)$  são, respectivamente, os comportamentos da trajetória nos eixos das ordenadas e das abscissas,  $a$  e  $b$ , respectivamente, os raios horizontais e verticais da elipse,  $r(\tau)$  o polinômio interpolador normalizado,  $\alpha$  o ângulo inicial do arco de elipse e  $\Delta\alpha$  o valor a ser incrementado a  $\alpha$ , definindo o ângulo final do percurso.

Para que o comportamento da trajetória seja suave no tempo, ou seja, os polinômios que compõem a trajetória e suas primeiras e segundas derivadas sejam contínuas, é necessário que  $r(\tau)$  seja representado pela equação 3.14.

Se  $\Delta\alpha$  for positivo, o seu valor será acrescido ao ângulo inicial  $\alpha$  no sentido anti-horário, caso contrário, este valor será acrescido no sentido horário.

A figura 6.8 mostra uma trajetória elipsoidal completa, formada pelo gerador de trajetória. O comportamento desta trajetória no tempo é mostrado na figura 6.9.



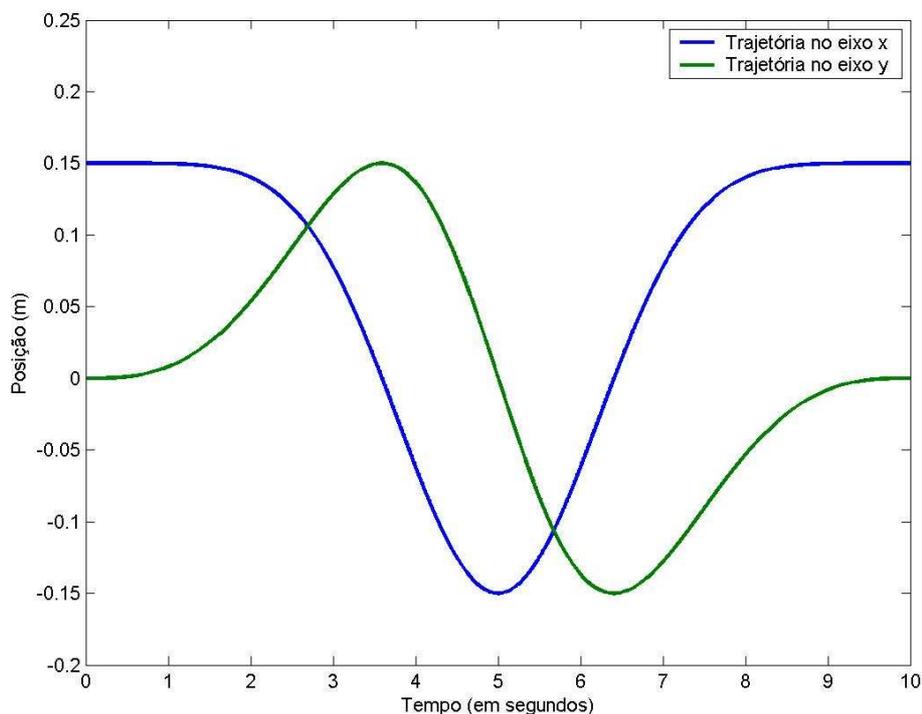
**Figura 6.8.** Trajetória elipsoidal.

#### 6.3.4 Trajetórias baseadas em comportamentos

Dois tipos de comportamentos foram utilizados para definir os trechos de uma trajetória a ser planejada. Um comportamento do tipo linear e um comportamento por arcos de elipse. Com estes dois tipos de comportamentos podem ser construídas trajetórias com formas variadas.

A figura 6.10 mostra uma trajetória gerada por oito comportamentos, sendo três lineares e cinco arcos de elipse.

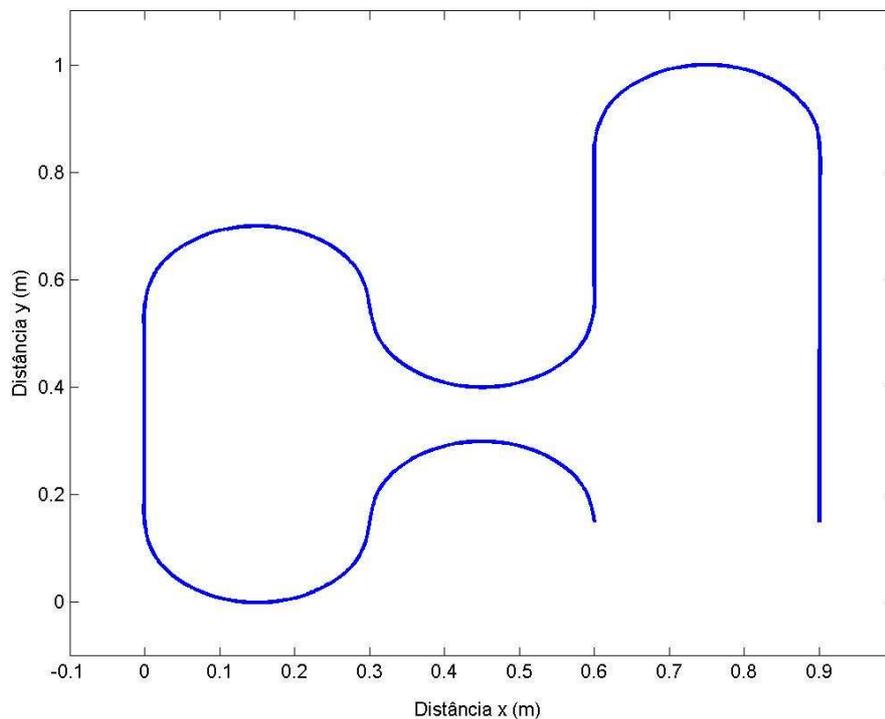
Esta trajetória foi planejada tomando o cuidado de não haver mudanças bruscas de posição, nos trechos que interceptam dois comportamentos. Mudanças bruscas de posição, podem acarretar mudanças bruscas de velocidade e aceleração, do robô, se não houver uma queda suave de velocidade a zero, nos pontos que interceptam dois trechos de trajetória.



**Figura 6.9.** Comportamento da trajetória elipsoidal no tempo.

Dois problemas irão surgir quando uma trajetória for composta por um conjunto grande de comportamentos, com velocidade e aceleração nulas nos seus extremos. Primeiro, a velocidade do robô irá oscilar muito durante o rastreamento da trajetória, potencializando o efeito do atrito estático e reduzindo, portanto, o desempenho do controlador de trajetória. Segundo, o esforço de controle poderá ser grande, pois o robô terá que compensar as velocidades nulas nos extremos de cada comportamento, com altas acelerações, para completar a trajetória em um tempo especificado. Torna-se inviável, portanto, o uso das técnicas de geração de trajetória lineares e elipsoidais para a construção dos comportamentos.

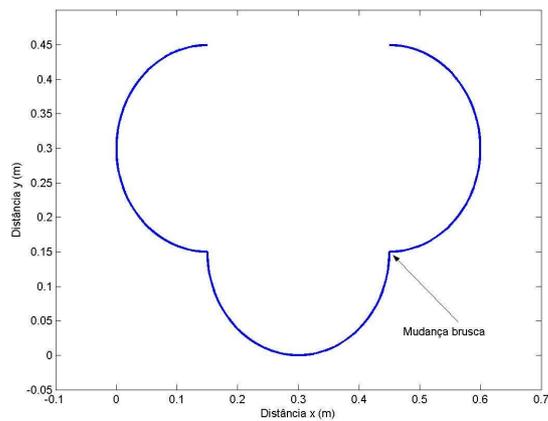
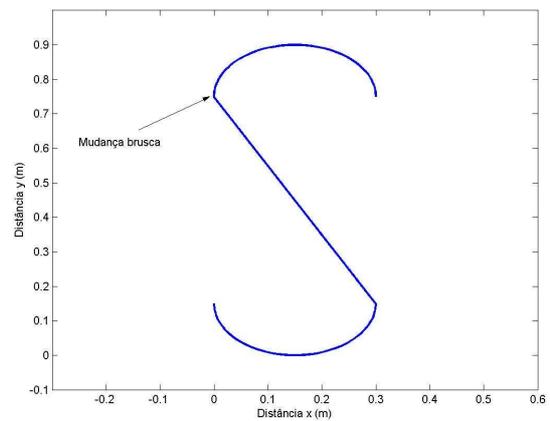
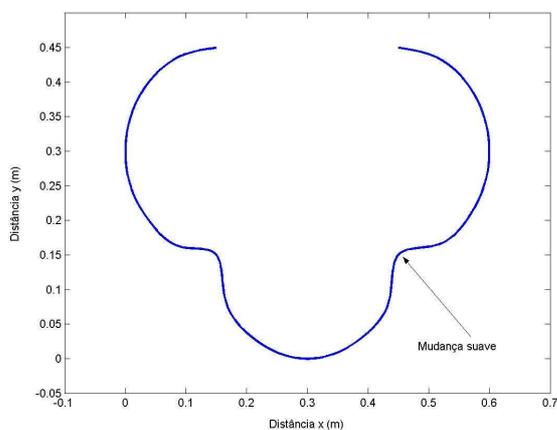
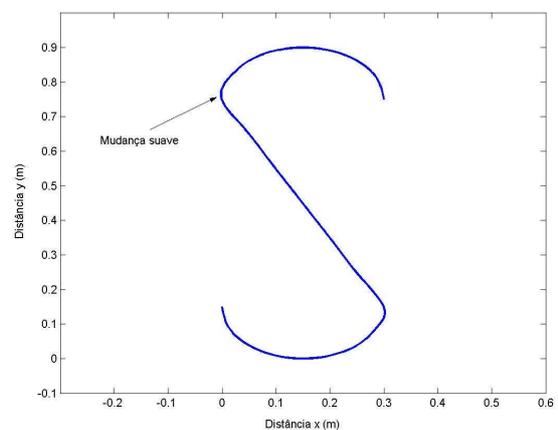
Para solucionar os problemas apontados acima, a trajetória é gerada em duas etapas. Na primeira etapa, os comportamentos constroem a trajetória sem a preocupação da ocorrência de mudanças bruscas de posição e sem a necessidade de gerar trechos de trajetórias por funções suaves. As técnicas de implementação dos comportamentos



**Figura 6.10.** Trajetória baseada em comportamentos.

são semelhante às técnicas de geração de trajetórias lineares e elipsoidais, exceto que o polinômio  $r(\tau)$  é linear. Na segunda etapa, a trajetória gerada pela primeira etapa é discretizada em um conjunto de pontos que são interpolados por uma função *spline*. Esta técnica suaviza as variações bruscas de posição, eliminando a oscilação da velocidade do robô, durante o percurso. A geração da trajetória final, pela segunda etapa, é semelhante à técnica de geração de trajetórias de caminho livre.

A figura 6.11 mostra a diferença entre trajetórias geradas, com e sem o uso da técnica de interpolação polinomial por funções *spline*. Observa-se que as mudanças bruscas são suavizadas pela *spline*.

(a) Sem o uso de funções *spline*(b) Sem o uso de funções *spline*(c) Com o uso de funções *spline*(d) Com o uso de funções *spline*

**Figura 6.11.** Geração de trajetórias baseadas em comportamentos com e sem o uso de funções *spline*

### 6.3.5 Implementação do gerador de trajetória

O módulo de geração de trajetória foi construído na linguagem nativa do Matlab, com uma interface no *Simulink*, ligando este módulo ao módulo de controle. A estrutura do módulo de geração de trajetória é mostrada na figura 6.12.

O bloco *Gera trajetória* deve ser chamado pela etapa de planejamento. Este bloco tem a função de gerar um caminho, podendo ser de forma livre, em forma de elipse, em forma de segmentos de reta ou baseada em comportamentos, a depender do tipo de

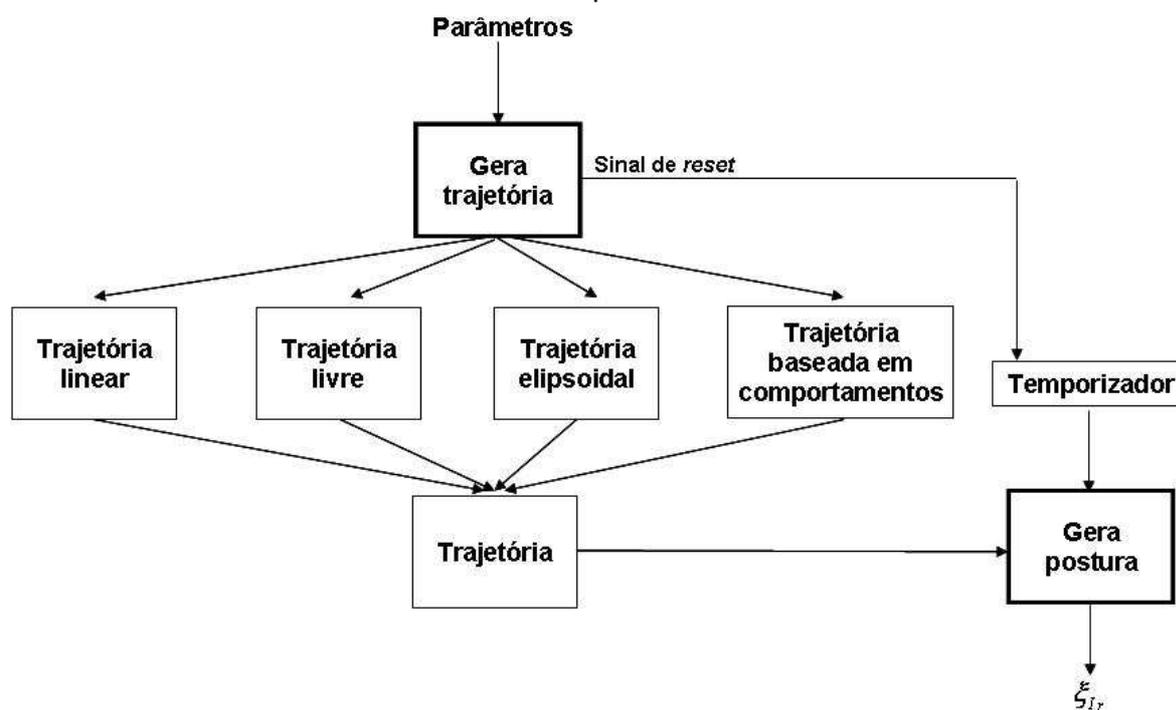


Figura 6.12. Estrutura do módulo de geração de trajetória.

trajetória selecionada. O caminho gerado é armazenado na estrutura *trajetoria*. Esta estrutura armazena o comportamento do percurso, em relação ao tempo, nos eixos das ordenadas e das abscissas. O comportamento no eixo das ordenadas é armazenado no campo  $x$  da estrutura. Este campo é um vetor, onde cada posição é o valor da ordenada de um ponto da trajetória interpolada. O comportamento no eixo das abscissas é armazenado no campo  $y$  da estrutura e segue o mesmo padrão do campo  $x$ . Existem valores de tempo associado a cada ponto  $(x, y)$ . Estes valores são armazenados no campo  $t$  da estrutura.

O bloco *Gera postura* é chamado, a cada instante de tempo, pelo gerador de trajetória. A estrutura *trajetoria* é passada como parâmetro, juntamente com um valor de instante de tempo gerado pelo temporizador. A posição no vetor  $t$ , da estrutura, do valor mais próximo do instante de tempo, é encontrada. Esta posição será a mesma posição dos valores nos campos  $x$  e  $y$ , correspondente a postura  $\xi_{I_r}$ , retornada pelo

bloco *Gera postura*.

O bloco *Gera trajetória* recebe cinco parâmetros. Os dois primeiros parâmetros são vetores e a interpretação do seu conteúdo depende do tipo da trajetória. Se o tipo for linear, os elementos do primeiro parâmetro correspondem, respectivamente, às ordenadas dos pontos inicial e final do percurso, enquanto que os elementos do segundo parâmetro, às abscissas destes mesmos pontos. Se o tipo for elipsoidal, os primeiros elementos do primeiro e segundo parâmetros correspondem, respectivamente, aos valores  $a$  e  $b$  das equações 6.17 e 6.18. Os valores de  $\alpha$  e  $\Delta\alpha$ , destas mesmas equações, são dados, respectivamente, pelo o segundo elemento do primeiro e segundo parâmetro. Os terceiros elementos do primeiro e segundo parâmetro informam o centro da elipse. Se o tipo for caminho livre, o primeiro parâmetro contém as ordenadas dos pontos a serem interpolados e o segundo parâmetro, as abscissas destes mesmos pontos. Por fim, se o tipo da trajetória for baseada em comportamentos, cada quatro elementos dos dois primeiros parâmetros correspondem aos parâmetros do comportamento. O primeiro dos quatro elementos informa o tipo do comportamento, os outros elementos são os parâmetros para a geração da trajetória linear ou elipsoidal, sendo o último elemento igual a zero, quando o comportamento for linear.

O terceiro parâmetro de entrada do bloco *Gera trajetória* corresponde ao tempo de geração da trajetória, o quarto parâmetro, o seu tipo e o quinto e último parâmetro, um flag indicando se a trajetória é nova ou é uma alteração da trajetória em execução. A cada nova trajetória gerada um sinal de *reset* é enviado para o temporizador iniciar a contagem do tempo.

As chamadas ao módulo *Gera trajetória*, pela etapa de planejamento, devem ser feitas de maneira assíncrona, ou seja, enquanto o gerador de trajetória faz chamadas ao módulo *Gera postura*, a cada instante de tempo, o planejamento de trajetória pode interromper a trajetória corrente, gerando uma trajetória nova, ou alterando a trajetória em execução, através de chamadas ao módulo *Gera trajetória*.

Uma chamada ao módulo *Gera trajetória* pode ocorrer em qualquer instante de

tempo. Isto é necessário porque, as condições dinâmicas do ambiente podem fazer com que novos planejamentos de trajetórias tenham que ser feitos antes do término da trajetória em execução. Um exemplo seria a geração de caminhos alternativos para desvio de obstáculos móveis.

Os algoritmos de geração de trajetória, em português estruturado, são mostrados no apêndice A. Estes algoritmos podem ser facilmente migrados para uma linguagem de programação, possibilitando a sua implementação em um robô real.

### 6.4 CONTROLADOR DE TRAJETÓRIA PROPOSTO

O sistema de controle de trajetória proposto é constituído de duas malhas de controle, em cascata. Uma malha externa, com um controlador baseado na cinemática, como mostrado na seção 5.2 e uma malha interna, com um controlador baseado na dinâmica, como mostrado na seção 5.3. São utilizados três compensadores de ruídos, um para cada sinal de controle, como mostrado na seção 5.5. O diagrama de blocos do sistema de controle de trajetória proposto é mostrado na figura 6.1

Para a obtenção dos resultados, o módulo de controle de trajetória foi construído no *Simulink*. Detalhes deste módulo no *Simulink* são mostrados no apêndice B.

É importante ressaltar que, para a implementação dos controladores em um robô real, é necessário fazer-se o uso de um método de discretização das leis de controle propostas no domínio contínuo.

### 6.5 SIMULAÇÃO E RESULTADOS

No capítulo anterior foram mostradas as respostas de alguns tipos de controladores a sinais de posição e velocidade, com o sistema sob determinadas condições.

O objetivo desta seção é mostrar o desempenho do controlador de trajetória pro-

posto, no rastreamento de trajetórias, geradas pelo gerador de trajetória, com presença e ausência de ruídos de entrada e saída e, com e sem o uso dos compensadores de ruídos. É mostrada a importância da geração de trajetórias por funções suaves, comparando o esforço de controle de trajetórias lineares com mudança brusca e suaves de velocidade e trajetórias baseadas em comportamentos, com e sem o uso da interpolação por função *spline*.

Foram geradas duas trajetórias de caminho livre, duas trajetórias baseadas em comportamentos, uma trajetória elipsoidal e um conjunto de trajetórias lineares, como referências para os controladores. Todas as trajetórias foram geradas em dez segundos, dentro de uma área de um metro quadrado. Na trajetória de caminho livre fechada, o robô atingiu a velocidade máxima de 0,5 m/s. Esta velocidade chegou a 0,8 m/s, na trajetória de caminho livre aberta, 0,4 m/s, na elipsoidal, 1,1 m/s no conjunto de trajetórias lineares, 0,3 m/s nas trajetórias baseadas em comportamentos.

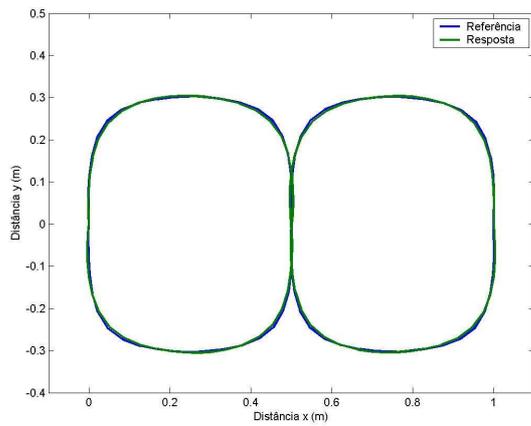
As figuras 6.13 e 6.14 mostram, respectivamente, as respostas dos controladores no rastreamento de uma trajetória de caminho livre fechada e os esforços de controle. As figuras 6.15 e 6.16 mostram, respectivamente, as respostas dos controladores a uma trajetória de caminho livre aberta e uma trajetória elipsoidal. Observa-se que na presença de ruídos, os compensadores difusos melhoram o desempenho dos controladores.

A figura 6.17 mostra o rastreamento de trajetórias lineares com mudanças bruscas e suaves de velocidade, nos pontos de interseção entre os segmentos de retas, e os respectivos esforços de controle. As trajetórias lineares com mudanças bruscas não foram geradas pelo gerador de trajetória. Nota-se a importância da geração de trajetórias com comportamentos suaves no tempo, quando os esforços de controle destas são comparados aos esforços de controle de trajetórias com comportamentos de mudanças bruscas. A figura 6.18 mostra o comportamento das trajetórias lineares no tempo.

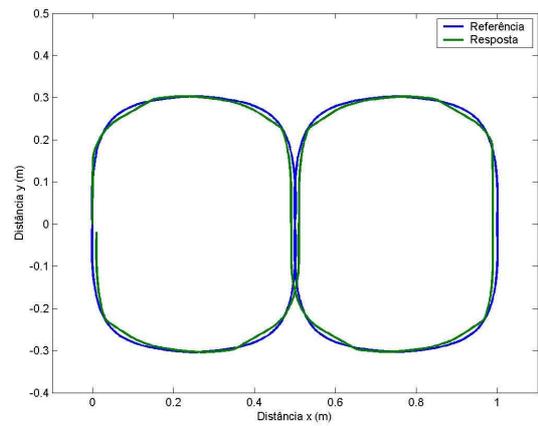
O rastreamento de trajetórias baseadas em comportamentos, com e sem o uso da etapa de interpolação por funções *spline* e a comparação dos esforços de controle,

são mostrados nas figuras 6.19 e 6.20, respectivamente. As trajetórias com mudanças bruscas de posição foram geradas apenas pela primeira etapa da geração de trajetórias baseadas em comportamentos. Observa-se que, a suavidade na mudança de posição, entre um comportamento e outro, diminui o esforço de controle, não sendo portanto necessário reduzir a velocidade e aceleração a zero, nestes pontos.

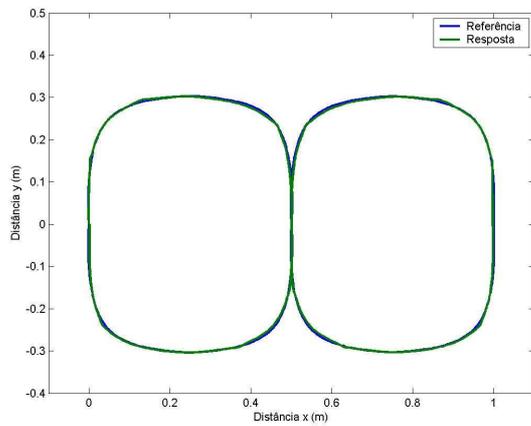
## 6.5 SIMULAÇÃO E RESULTADOS



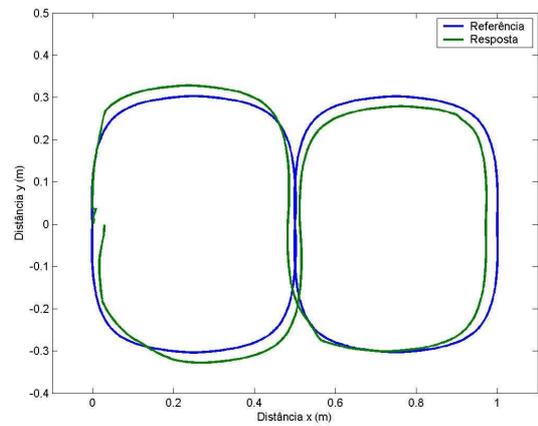
(a) Ausência de ruídos



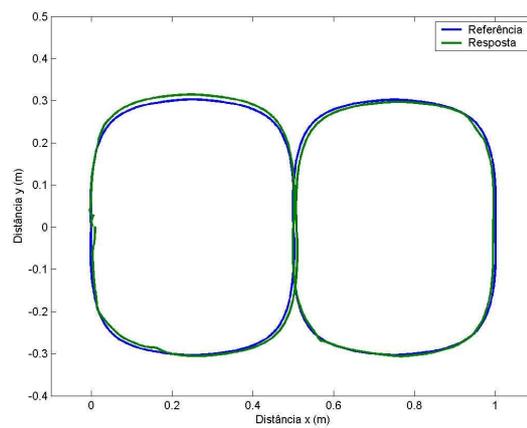
(b) Com ruídos de entrada, sem os compensadores



(c) Com ruídos de entrada e com os compensadores



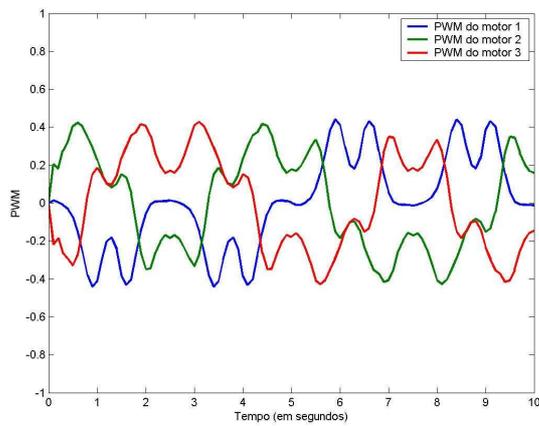
(d) Com ruídos de entrada e saída, sem os compensadores



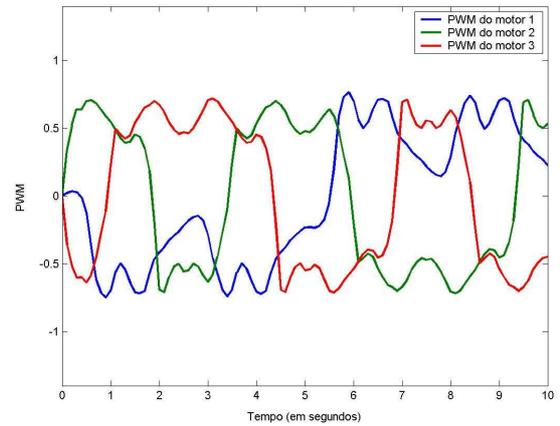
(e) Com ruídos de entrada e saída e com os compensadores

**Figura 6.13.** Resposta do sistema à trajetória de caminho livre fechada

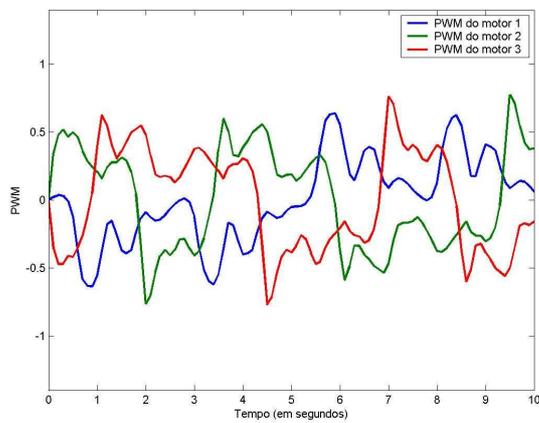
## 6.5 SIMULAÇÃO E RESULTADOS



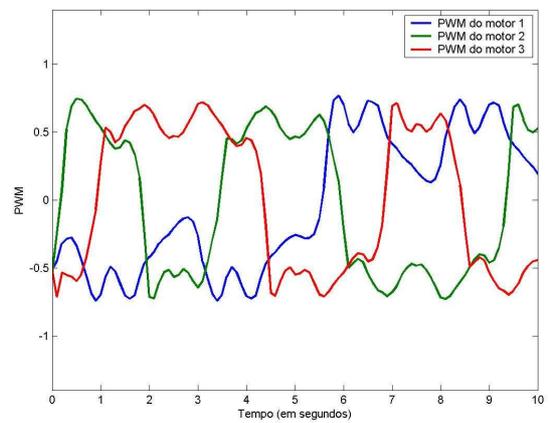
(a) Ausência de ruídos



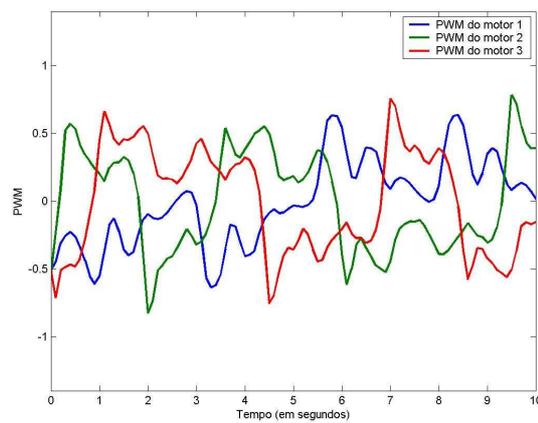
(b) Com ruídos de entrada, sem os compensadores



(c) Com ruídos de entrada e com os compensadores



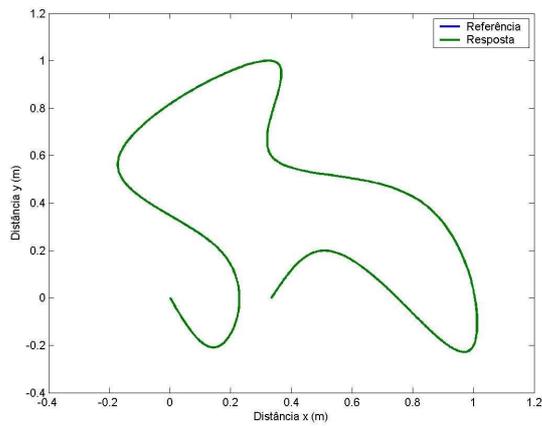
(d) Com ruídos de entrada e saída, sem os compensadores



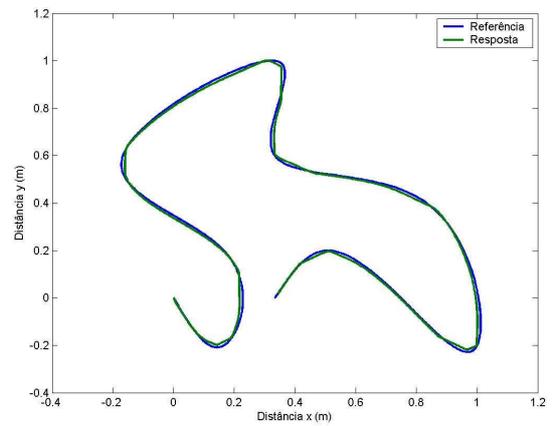
(e) Com ruídos de entrada e saída e com os compensadores

**Figura 6.14.** Esforço de controle do sistema à trajetória de caminho livre fechada

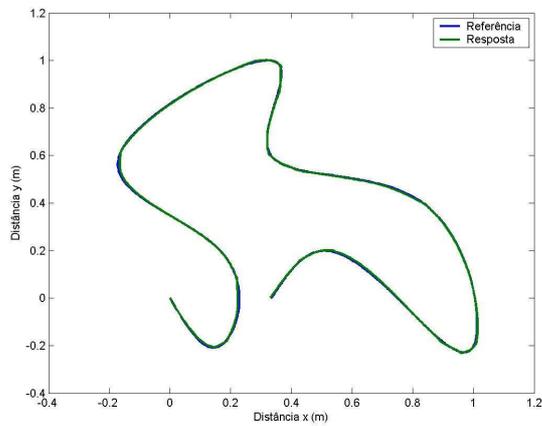
## 6.5 SIMULAÇÃO E RESULTADOS



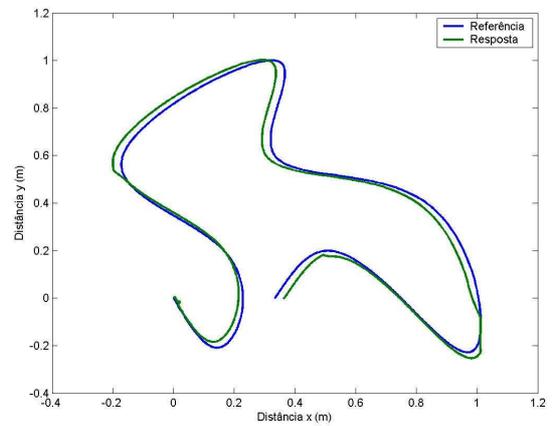
(a) Ausência de ruídos



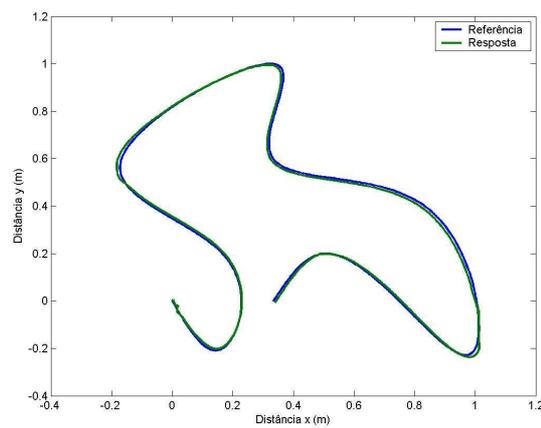
(b) Com ruídos de entrada, sem os compensadores



(c) Com ruídos de entrada e com os compensadores



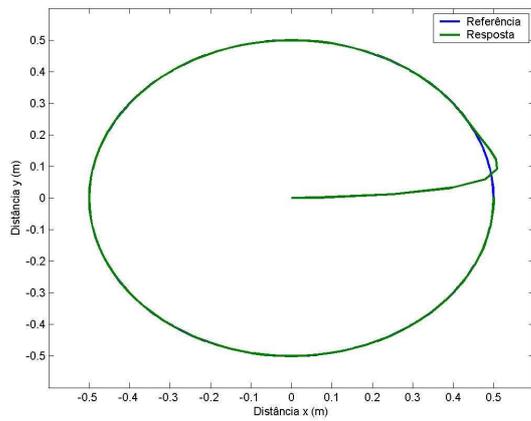
(d) Com ruídos de entrada e saída, sem os compensadores



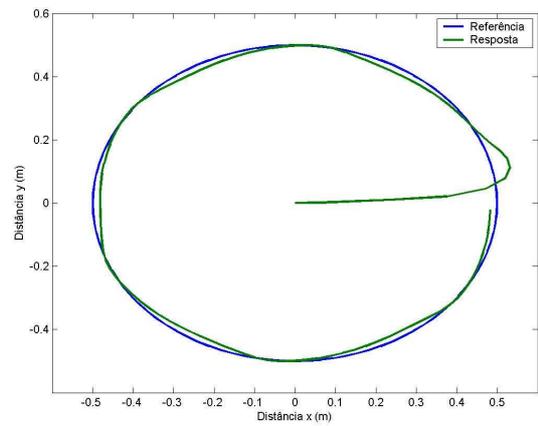
(e) Com ruídos de entrada e saída e com os compensadores

**Figura 6.15.** Resposta do sistema à trajetória de caminho livre aberta

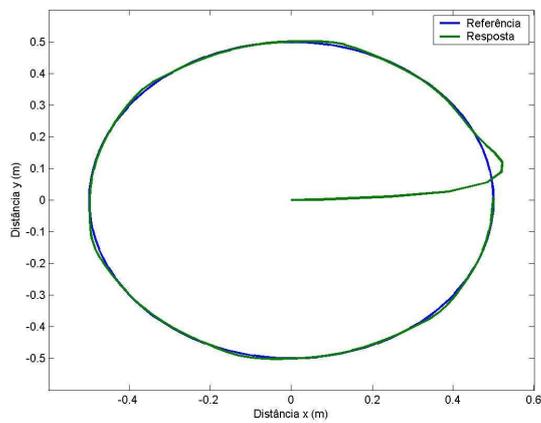
## 6.5 SIMULAÇÃO E RESULTADOS



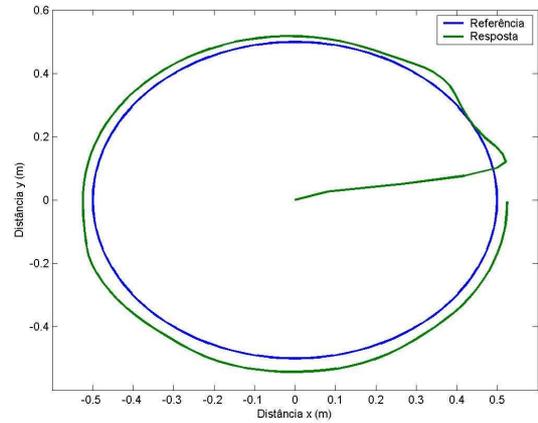
(a) Ausência de ruídos



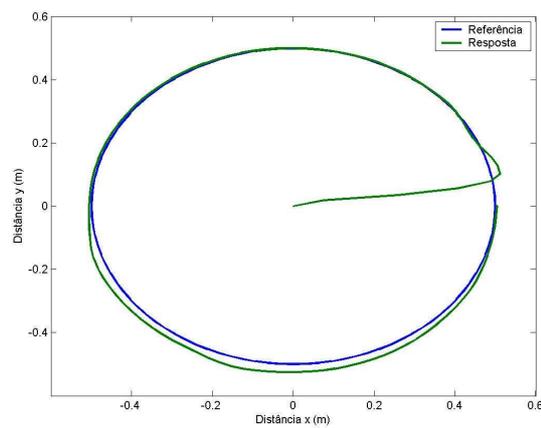
(b) Com ruídos de entrada, sem os compensadores



(c) Com ruídos de entrada e com os compensadores

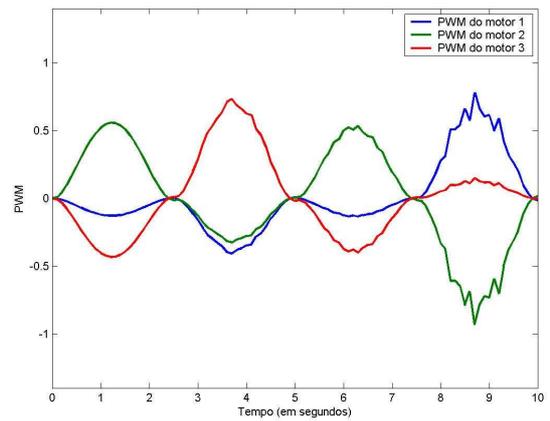
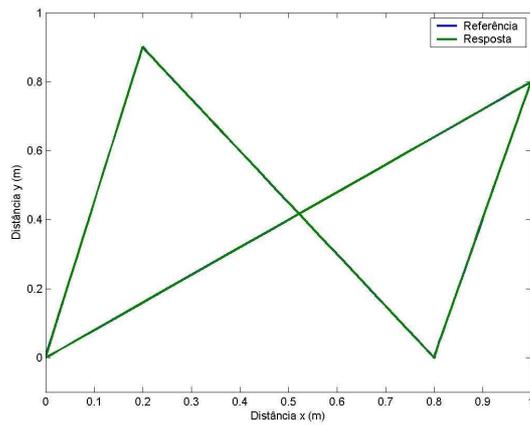


(d) Com ruídos de entrada e saída, sem os compensadores

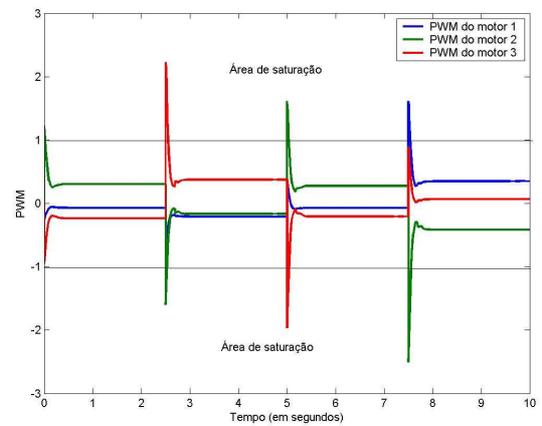
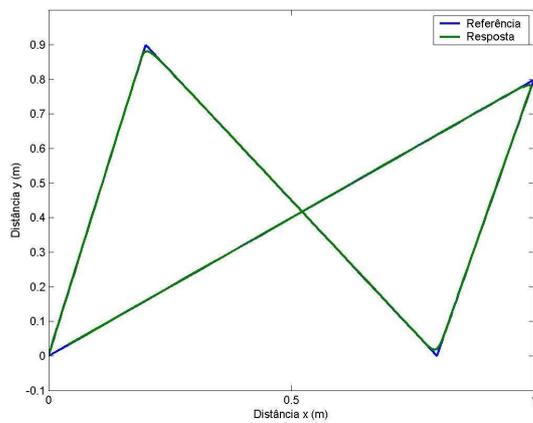


(e) Com ruídos de entrada e saída e com os compensadores

**Figura 6.16.** Resposta do sistema à trajetória elipsoidal

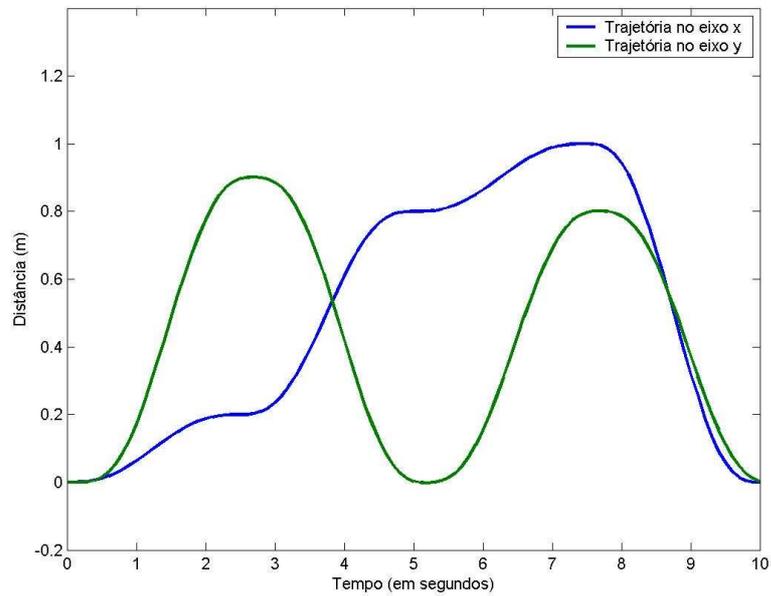


(a) Trajetória com mudanças suaves de velocidade (b) Esforço de controle com mudanças suaves de velocidade

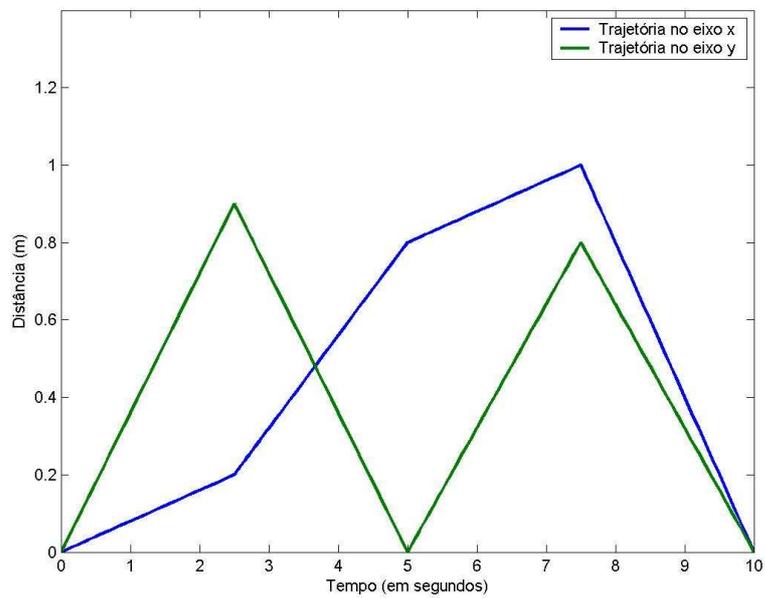


(c) Trajetória com mudanças bruscas de velocidade (d) Esforço de controle com mudanças bruscas de velocidade

**Figura 6.17.** Resposta do sistema à trajetórias lineares com mudanças suaves e bruscas de velocidade e os esforços de controle

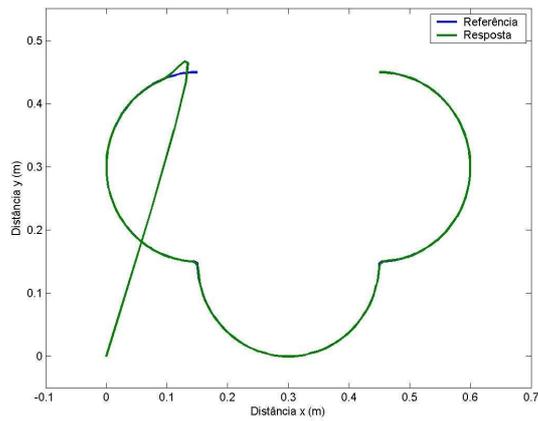


(a) Com mudanças suaves de velocidade

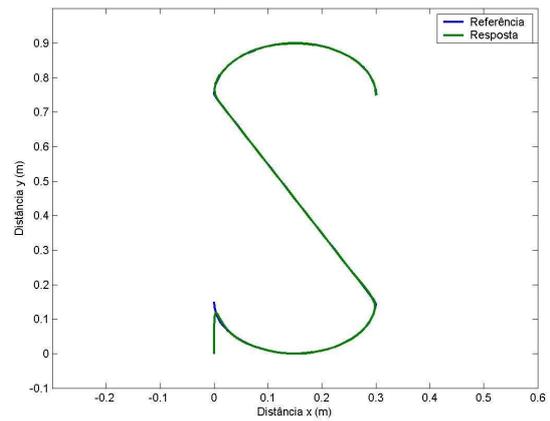


(b) Com mudanças bruscas de velocidade

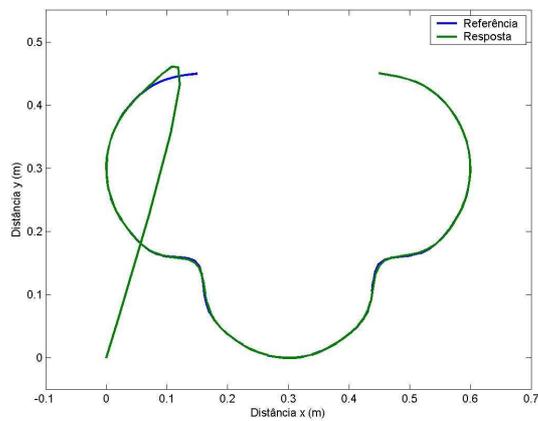
**Figura 6.18.** Comportamento das trajetórias lineares, com mudanças bruscas e suaves, no tempo, sem a presença de ruídos



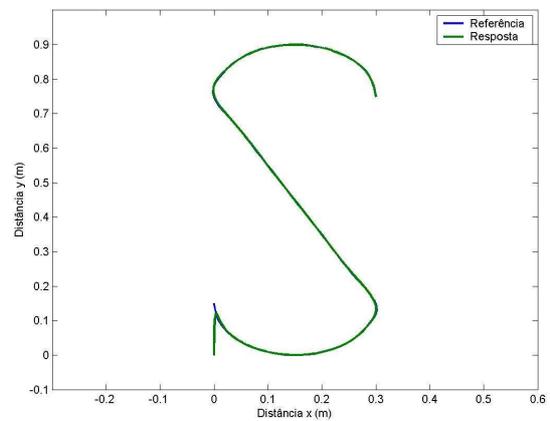
(a) Trajetória 1. Sem o uso da interpolação por funções *spline*



(b) Trajetória 2. Sem o uso da interpolação por funções *spline*

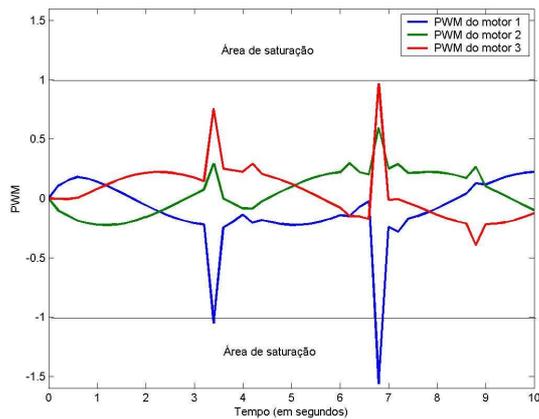


(c) Trajetória 1. Com o uso da interpolação por funções *spline*

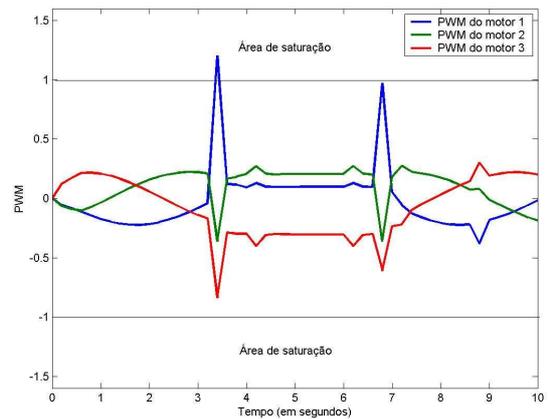


(d) Trajetória 2. Com o uso da interpolação por funções *spline*

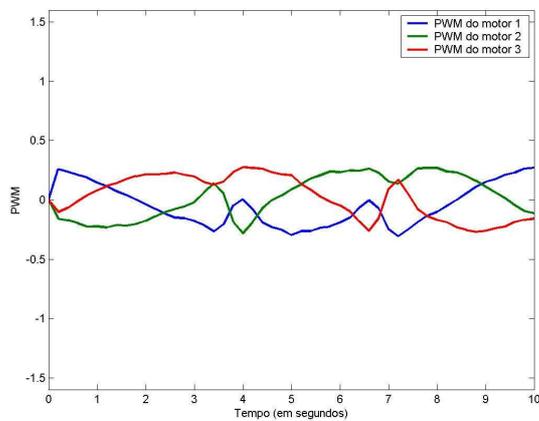
**Figura 6.19.** Resposta do sistema às trajetórias baseadas em comportamentos, com e sem o uso da interpolação por funções *spline*



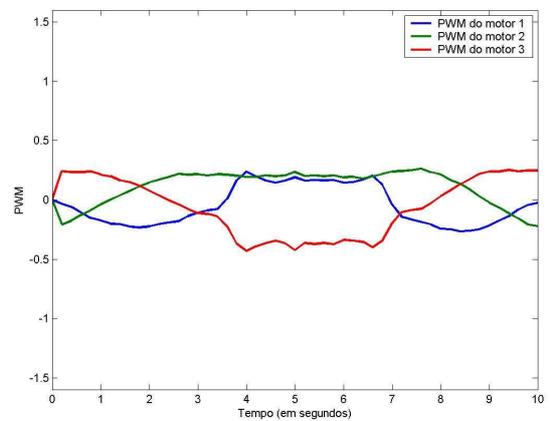
(a) Trajetória 1. Sem o uso da interpolação por funções *spline*



(b) Trajetória 2. Sem o uso da interpolação por funções *spline*



(c) Trajetória 1. Com o uso da interpolação por funções *spline*



(d) Trajetória 2. Com o uso da interpolação por funções *spline*

**Figura 6.20.** Esforços de controle do sistema às trajetórias baseadas em comportamentos, com e sem o uso da interpolação por funções *spline*

### 6.6 CONCLUSÃO

Este capítulo apresentou o gerador e o controlador de trajetória proposto, mostrando o desempenho do controlador de trajetória a trajetórias de referências. Foi mostrada a eficiência dos compensadores difusos em resposta a ruídos de entrada e saída. Observou-se que trajetórias com mudanças bruscas de posição, sem mudanças suaves de velocidade, geram um alto esforço de controle nos atuadores. Foi mostrada, também, a importância do uso da técnica de geração de trajetória por um conjunto de funções contínuas, denominadas de comportamentos. O uso desta técnica foi combinado com a técnica de *spline* para eliminar as mudanças bruscas de posição nos pontos de interseção entre dois comportamentos.

## CONCLUSÃO E TRABALHOS FUTUROS

A tarefa da navegação autônoma de robôs, o mote principal da robótica móvel, deve ser cumprida de maneira eficiente, caso contrário, um objetivo pré-definido poderá não ser atingido. Diversos fatores influenciam em uma eficiente navegação. Este trabalho tratou questões relacionadas com a geração e controle de trajetória de robôs móveis omnidirecionais. Aspectos relacionados à implementação de um módulo de geração e controle de trajetória, ainda que não tenham sido, de fato, implementadas em um robô real, mostraram uma proposta para a criação de uma camada de software que oferece os serviços dos dois níveis mais baixo da navegação de robôs móveis. Esta camada poderá servir de base para futuros trabalhos, oferecendo uma interface para que algoritmos de planejamento de trajetória sejam testados, e que, de maneira transparente, a trajetória planejada seja gerada e rastreada pelo robô.

Foi mostrada, também, a importância da geração de trajetórias suaves, evitando um alto esforço de controle dos atuadores do robô, o que poderia acarretar em problemas como deslizamento das rodas sobre o solo, ou a operação dos atuadores na zona de saturação. Problemas como atrito estático, ruído na leitura dos sensores e parâmetros incertos do modelo, foram atenuados por compensadores baseado em controladores difusos do tipo PI.

Os problemas do deslizamento das rodas do robô sobre o solo e a operação dos atuadores na zona de saturação, apesar de serem amenizados, pela geração de trajetórias suaves, não são tratados, caso estes venham a ocorrer. Estes problemas podem ser solucionados com controladores sob restrições e tolerantes a deslizamento, sendo uma proposta para trabalhos futuros.

## REFERÊNCIAS

ADÔRNO, B. V.; AGUIAR, C. S. R.; BORGES, G. A. Planejamento de trajetória para o robô omni utilizando o algoritmo mapa de rotas probabilísticos. *VII Simpósio Brasileiro de Automação Inteligente / II IEEE Latin-American Robotics Symposium*, p. 1–8, Setembro 2005.

ALBAGUL, A.; MARTONO, W. Dynamic modelling and adaptive traction control for mobile robots. *International Journal of Advanced Robotic Systems*, v. 1, n. 3, p. 149–154, 2004.

ALMEIDA, A. D. V. de. *B-Spline:CAEP - Algoritmos Culturais Para a Geração de Trajetórias B-Spline de Robôs Móveis*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Paraná, Curitiba, Outubro 2005.

ANDRADE, J. F. A.; MENDELECK, A.; ZAMPIERI, D. E. Geração de trajetórias para robôs móveis autônomos usando redes neurais artificiais. In: *Anais do V Congresso Brasileiro de Redes Neurais*. Rio de Janeiro: [s.n.], 2001. v. 1, p. 571–576.

ANGELES, J. *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*. 2. ed. New York: Springer-Verlag New York, Inc., 2003.

BORENSTEIN, J. Control and kinematic design of multi-degree-of-freedom mobile robots with compliant linkage. *IEEE Transactions on Robotics and Automation*, v. 11, n. 1, p. 21–35, Fevereiro 1995.

BORESI, A. P.; SCHMIDT, R. J. *Dynamics*. Pacific Grove, CA: Brooks/Cole, 2001.

BRAUNL, T. *Embedded Robotics*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003. ISBN 3540034366.

CAMPION, G.; BASTIN, G.; DANDREA-NOVEL, B. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, v. 12, p. 47–62, Fevereiro 1996.

CARTER, B. et al. Mechanical design and modeling of an omni-directional robocup player. In: *Proceedings RoboCup 2001 International Symposium*. Seattle, WA: [s.n.], 2001.

CHENG, M.-P.; TSAI, C.-C. Dynamic modeling and tracking control of a nonholonomic wheeled mobile manipulator with two robotic arms. In: *Proceedings of the 42nd IEEE Conference on Decision and Control Maui*. Maui, Hawaii USA: [s.n.], 2003. p. 2932–2937.

DIAS, A. et al. Uma metodologia para geração de trajetórias de manipuladores no espaço de trabalho. In: *Congresso Nacional de Engenharia Mecânica*. Natal / RN: [s.n.], 2000.

FIERRO, R.; LEWIS, F. L. Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics. In: *Proceedings of the IEEE Conference on Decision and Control*. New Orleans, LA: [s.n.], 1995. p. 3805–3810.

FIERRO, R.; LEWIS, F. L. Control of a nonholonomic mobile robot using neural networks. *IEEE Transactions on Neural Networks*, v. 9, n. 4, p. 589–600, 1998.

FRANCO, A. C. S.; COSTA, A. C. L. Modelagem e simulação do robô omnidirecional axebot. *Anais do XVI Congresso Brasileiro de Automática*, p. 2844–2849, Outubro 2006.

FUKAO, T.; NAKAGAWA, H.; ADACHI, N. Adaptive tracking control of a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, v. 16, n. 5, p. 609–615, Outubro 2000.

GARRIDO, B. M. et al. Aplicação de redes neurais na geração de trajetórias para robôs móveis. In: *Anais do II Congresso Nacional de Engenharia Mecânica*. João Pessoa: [s.n.], 2002. v. 1, p. 1–8.

GOMES, G. K. *Controle Preditivo em Tempo-Real Para Seguimento de Trajetória de Veículos Autônomos*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, Março 2006.

GU, D.; HU, H. Neural predictive control for a car-like mobile robot. *International Journal of Robotics and Autonomous Systems*, v. 39, n. 2, p. 1279–1284, Maio 2002.

ISIDORI, A. *Nonlinear control systems: an introduction (2nd ed.)*. New York, NY, USA: Springer-Verlag New York, Inc., 1989. ISBN 0-387-50601-2.

KANAYAMA, Y. et al. A stable tracking control method for an autonomous mobile robot. In: *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 1990. p. 384–389.

KHALIL, W.; DOMBRE, E. *Modeling, Identification and Control of Robots*. Bristol, PA, USA: Taylor & Francis, Inc., 2002. ISBN 1560329831.

KIM, B.; NECSULESCU, D.; SASIADEK, J. Model predictive control of an autonomous vehicle. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Como, Italy: [s.n.], 2001. v. 2, p. 73–86. ISBN 0-7803-6736-7.

KODAGODA, K. R. S.; WIJESOMA, W. S.; TEOH, E. K. Fuzzy speed and steering control of an agv. *IEEE Transactions on Control Systems Technology*, v. 10, n. 1, p. 112–120, Janeiro 2002.

LAGES, W. F. *Controle e Estimação de Posição e Orientação de Robôs Móveis*. Dissertação (Mestrado) — Instituto Tecnológico de Aeronáutica, São José dos Campos, 1998.

LAURA, T. L. *Modelagem Dinâmica da Estrutura da Base Móvel de Robôs Manipuladores com Inclusão das Não Linearidades de Entrada*. Dissertação (Mestrado) — Universidade Federal da Bahia, Salvador, 2006.

LIU, Y. et al. Omni-directional mobile robot controller design by trajectory linearization. In: *Proceedings American Control Conference*. Denver, CO: [s.n.], 2003.

MAMDANI, E. H. Advances in the linguistic synthesis of fuzzy controllers. *International Journal of Man-Machine Studies*, v. 8, p. 669–678, 1976.

MCNEILL, F. M.; THRO, E. *Fuzzy Logic: A Practical Approach*. [S.l.]: Morgan Kaufmann, 1994.

MUIR, P. F.; NEUMAN, C. P. *Kinematics Modeling of Wheeled Mobile Robots*. Pittsburgh, PA, Junho 1986.

NELSON, W. L. Continuous steering-function control of robot carts. *IEEE Transactions on Industrial Electronics*, v. 36, n. 3, p. 330–337, Agosto 1989.

NELSON, W. L. An intelligent predictive control approach to path tracking problem of autonomous mobile robot. *IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, CA, USA, v. 4, p. 3301–3306, Outubro 1998.

OGATA, K. *Engenharia de Controle Moderno*. 4. ed. São Paulo: Prentice Hall, 2003.

OLIVEIRA, V. M. de. *Técnicas de Controle de Robôs Móveis*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, fevereiro 2001.

OUBBATI, M. *Neural Dynamics for Mobile Robot Adaptive Control*. Tese (Doutorado) — Institut für Parallele und Verteilte Systeme der Universität Stuttgart, Tag der mündlichen Prüfung, Junho 2006.

PAROMTCHIK, I. E.; ASAMA, H. A motion generation approach for an omnidirectional vehicle. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. San Francisco, CA, USA: [s.n.], 2000. p. 1213–1218. ISBN 0-7803-5889-9.

PAROMTCHIK, I. E.; REMBOLD, U. A practical approach to motion generation and control for an omnidirectional mobile robot. In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. San Diego, CA, USA: [s.n.], 1994. p. 2790–2795. ISBN 0-8186-5330-2.

PIRES, E. J. S.; MACHADO, J. A. T.; OLIVEIRA, P. B. de M. Robot trajectory planning using multi-objective genetic algorithm optimization. In: *Genetic and Evolutionary Computation Conference, Proceedings, Part I*. Seattle, WA, USA: Springer, 2004. (Lecture Notes in Computer Science, v. 3102), p. 615–626. ISBN 3-540-22344-4.

RAFFO, G. V. *Algoritmos de controle preditivo para seguimento de trajetória de veículos autônomos*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, 2005.

RAMOS, J. J. G. *Geração de Trajetórias Contínuas para Robôs: Aspectos Cinemáticos e Computacionais*. Dissertação (Mestrado) — FEE/UNICAMP, Campinas, SP, Junho 1986.

REIS, G. A. dos. *Controle  $H_\infty$  não linear de robôs móveis com rodas*. Dissertação (Mestrado) — Universidade de São Paulo, São Carlos, 2006.

SANDRI, S.; CORREA, C. Lógica nebulosa. *V Escola de Redes Neurais, Promoção: Conselho Nacional de Redes Neurais*, ITA, São José dos Campos, SP, p. c073–c090, julho 1999.

SCHNITMAN, L.; YONEYAMA, T. A method for learning membership functions in mamdani fuzzy models. In: *Congresso Brasileiro de Automática*. Florianópolis: Proceedings of the XIII CBA, 2000.

SCHROEDER, G. N. et al. Simulador gráfico para controle de robôs móveis omnidirecionais. *INFOCOMP Journal of Computer Science*, Universidade Federal de Lavras, Brasil, v. 4, n. 4, p. 38–47, Junho 2005.

SIEGWART, R.; NOURBAKHSI, I. R. *Introduction to Autonomous Mobile Robots*. London, England: [s.n.], 2004.

SLOTINE, J. J. E.; LI, J. W. *Applied nonlinear control*. [S.l.]: Prentice Hall, 1991.

SUGISAKA, F. D. M. Adaptive fuzzy control based on neural network for a mobile vehicle. In: *Transactions of the Institute of Electrical Engineers of Japan*. [S.l.: s.n.], 2003. v. 123, p. 786–795.

TSOUKALAS, L. H.; UHRIG, R. E. *Fuzzy and Neural Approaches in Engineering*. New York, NY, USA: John Wiley & Sons, Inc., 1996. ISBN 0471160032.

VAZ, J. M.; FABRO, J. A. Snnap - sistema neural de navegação em ambientes pré-mapeados. In: *Proceedings of the IV Brazilian Conference on Neural Networks - IV Congresso Brasileiro de Redes Neurais*. [S.l.: s.n.], 1999. p. 118–123.

WATANABE, K. Control of an omnidirectional mobile robot. In: *Second International Conference on Knowledge-Based Intelligent Electronic System*. Adelaide, Australia: [s.n.], 1998. p. 51–60.

WATANABE, K.; IZUMI, K.; HAN, F. Stochastic fuzzy servo control using multiple linear dynamic models. In: *Second International Conference on Knowledge-Based Intelligent Electronic System*. Adelaide, Australia: [s.n.], 1998. p. 474–482.

WATANABE, K. et al. Feedback control of an omnidirectional autonomous platform for mobile service robots. *Journal of Intelligent and Robotic Systems*, Kluwer Academic Publishers, Hingham, MA, USA, v. 22, n. 3-4, p. 315–330, 1998. ISSN 0921-0296.

WILLIAMS, R. L. et al. Dynamic model with slip for wheeled omnidirectional robots. *IEEE Transactions on Robotics and Automation*, v. 18, n. 3, p. 285–293, 2002.

WILSON, L. et al. Design and modeling of a redundant omni-directional robotcup goalie. In: *Proceedings RoboCup 2001 International Symposium*. Seattle, WA: [s.n.], 2001.

WIT, C. C. D. et al. Nonlinear control design for mobile robots. In: *World Scientific Series in Robotics and Intelligent Systems*. [S.l.]: World Scientific Publisher, 1993. p. 121–157.

YAMADA, T. et al. Dynamic model and control for a holonomic omnidirectional mobile robot. *Autonomous Robots*, Kluwer Academic Publishers, Hingham, MA, USA, v. 11, n. 2, p. 173–189, 2001. ISSN 0929-5593.

ZADEH, L. A. Fuzzy sets. *Fuzzy Sets, Information and Control*, v. 8, p. 338–353, 1965.

## ALGORITMOS DE GERAÇÃO DA TRAJETÓRIA

São descritos neste apêndice os algoritmos, em português estruturado, de geração de trajetória e de postura referentes aos blocos *Gera trajetoria* e *Gera postura* da figura 6.12.

O algoritmo de geração da trajetória, mostrado abaixo, tem como entradas as variáveis  $X$ ,  $Y$ ,  $tf$ ,  $tipo$  e  $novo$ , e como saída a estrutura *trajetoria*. Os valores dos dois primeiros parâmetros  $X$  e  $Y$  dependem do tipo da trajetória a ser gerada, como mencionado na seção 6.3.5. O parâmetro  $tf$  especifica o tempo de execução do percurso,  $tipo$  pode assumir os valores: 0, para linear; 1, para caminho livre; 2, para elipsoidal e 3 para trajetória baseada em comportamentos. Quando o valor do parâmetro  $novo$  for 1, um sinal de *reset* é enviado ao temporizador, zerando a variável de entrada *tempo* do módulo *Gera postura*.

---

### Algoritmo: Gera trajetória

---

```

01: entradas:  $X, Y, tf, tipo, novo$ 
02: saída: trajetoria
03: inicio
04:     se ( $tipo = 0$ ) então
05:          $trajetoria \leftarrow trajetoria\_linear(X, Y, tf)$ 
06:     senão se ( $tipo = 1$ )
07:          $trajetoria \leftarrow trajetoria\_livre(X, Y, tf)$ 
08:     senão se ( $tipo = 2$ )
09:          $trajetoria \leftarrow trajetoria\_elipsoidal(X, Y, tf)$ 

```

```
09: |      senão se (tipo = 3)
10: |         trajetoria ← trajetoria_comportamentos(X, Y, tf)
11: |      fim
12: |      se (novo = 1) então
13: |         Enviar sinal de reset para o temporizador
13: |      fim
14: fim
```

---

O algoritmo de geração de postura, mostrado abaixo, tem como entradas o parâmetro *tempo*, vindo do temporizador, e a estrutura *trajetoria*, gerada pelo gerador de trajetória. O parâmetro de saída é a estrutura *postura*, tendo como campos *x* e *y*, correspondentes, respectivamente, às coordenadas *x* e *y* da postura  $\xi_{I_r}$ .

---

**Algoritmo:** Gera postura

---

```
01: entradas: trajetoria, tempo
02: saída: postura
03: inicio
04: |      se (sinal de reset enviado ao temporizador) então
05: |         tempo ← 0
06: |      fim
07: |      n ← 0
08: |      enquanto (trajetoria.t[n] ≤ tempo) faça
09: |         n ← n + 1
10: |      fim
11: |      postura.x ← trajetoria.x[n]
12: |      postura.y ← trajetoria.y[n]
13: fim
```

---

As funções de geração de trajetória, chamadas pelo módulo *Gera trajetória*, são mostradas logo abaixo. Estas funções são responsáveis pela geração da estrutura

*trajetoria* que é lida pelo módulo *Gera postura*.

A função *trajetoria.linear*, cujo algoritmo é mostrado abaixo, gera um comportamento de trajetória conforme as equações 3.1 e 3.14.  $X$  é um vetor que contém as ordenadas dos pontos inicial e final do percurso e  $Y$  as abscissas destes mesmos pontos. O valor de  $tf$  indica o período de tempo entre o início e o final do percurso.

As linhas 3 a 6 declaram a estrutura da variável *trajetoria*, denominada de *RTRAJETORIA*. As linhas 12 a 15 atribuem os valores de posição dos pontos inicial e final do percurso, de acordo com os parâmetros  $X$  e  $Y$ . As linhas 19 a 27 constroem o comportamento da trajetória nos eixos das ordenadas (linha 20) e das abscissas (linha 21), em função da variável  $t$ . O valor desta variável corresponde a um instante de tempo entre 0 e o valor de  $tf$ , com o tempo de amostragem definido pela variável  $ta$ .

---

**Algoritmo:** Trajetória linear

---

```
01: função trajetoria_linear( $X$  vetor de real,  $Y$  vetor de real,  $tf$  real)
02: tipos
03:     registro RTRAJETORIA
04:     inicio
05:     |      $x, y, t$  :vetor de real
06:     fim
07: variáveis
08:      $x\_ini, y\_ini, x\_fim, y\_fim, t, ta$ : real
09:      $n$ : inteiro
10:      $trajetoria$ : RTRAJETORIA
11: inicio
12: |      $x\_ini \leftarrow X[1]$ 
13: |      $x\_fim \leftarrow X[2]$ 
14: |      $y\_ini \leftarrow Y[1]$ 
15: |      $y\_fim \leftarrow Y[2]$ 
16: |      $t \leftarrow 0$ 
```

```
17:    $ta \leftarrow 0.01$ 
18:    $n \leftarrow 1$ 
19:   enquanto ( $t \leq tf$ ) faça
20:      $x \leftarrow x_{ini} + (10 * (t/TF)^3 - 15 * (t/TF)^4 + 6 * (t/TF)^5) * (x_{fim} - x_{ini})$ 
21:      $y \leftarrow y_{ini} + (10 * (t/TF)^3 - 15 * (t/TF)^4 + 6 * (t/TF)^5) * (y_{fim} - y_{ini})$ 
22:      $t \leftarrow t + ta$ 
23:      $n \leftarrow n + 1$ 
24:      $trajetoria.x[n] \leftarrow x$ 
25:      $trajetoria.y[n] \leftarrow y$ 
26:      $trajetoria.t[n] \leftarrow t$ 
27:   fim
28:   retorne( $trajetoria$ )
29: fim
```

---

A função *trajetoria\_elipsoidal*, cujo algoritmo é mostrado abaixo, gera um comportamento de trajetória conforme as equações 6.17 e 6.18. Os parâmetros  $X$  e  $Y$  são vetores de três posições. Os valores da primeira posição de  $X$  e  $Y$  correspondem, respectivamente, as constantes  $a$  e  $b$  das equações 6.17 e 6.18. Os valores de  $\alpha$  e  $\Delta\alpha$ , destas mesmas equações, são dados, respectivamente, pelos valores da segunda posição dos parâmetros  $X$  e  $Y$ . Os valores da terceira posição de  $X$  e  $Y$  correspondem ao centro da elipse.

As linhas 20 a 29 constroem o comportamento da trajetória nos eixos das ordenadas (linha 22) e das abscissas (linha 23), em função da variável  $t$ .

---

**Algoritmo:** Trajetória elipsoidal

---

```
01: função trajetoria_elipsoidal( $X$  vetor de real,  $Y$  vetor de real,  $tf$  real)
02: tipos
03:   registro RTRAJETORIA
04:   inicio
05:      $x, y, t$  : vetor de real
```

```
06:      fim
07: variáveis
08:      a, b, PI, t, ta: real
09:      n: inteiro
10:      trajetoria: RTRAJETORIA
11: inicio
12:      a ← X[1]
13:      b ← Y[1]
14:      alpha ← X[2]
15:      Dalpha ← Y[2]
16:      PI ← 3.14
17:      t ← 0
18:      ta ← 0.01
19:      n ← 1
20:      enquanto (t ≤ tf) faça
21:          r ← 10 * (t/tf)3 - 15 * (t/tf)4 + 6 * (t/tf)6
22:          x ← a cos(Dalpha * r + alpha)
23:          y ← b sin(Dalpha * r + alpha)
24:          trajetoria.x[n] ← x + X[3]
25:          trajetoria.y[n] ← y + Y[3]
26:          trajetoria.t[n] ← t
27:          t ← t + ta
28:          n ← n + 1
29:      fim
30:      retorne(trajetoria)
31: fim
```

---

A função *trajetoria\_livre*, cujo algoritmo é mostrado abaixo, gera um comportamento de trajetória conforme as equações 3.34 e 3.37. Os valores de *X* e *Y* representam

as coordenadas dos pontos a serem interpolados pela função *spline* natural, sendo os valores de  $X$  as ordenadas e os valores de  $Y$  as abscissas dos pontos.

As linhas 17 a 21, atribuem os valores ao vetor  $T$ . Os valores deste vetor representam os instantes de tempo referentes ao início de cada polinômio cúbico. Supondo que o valor de  $tf$  seja 100 e que existam 10 pontos a serem interpolados, os valores de  $T$  serão termos de uma progressão aritmética de razão 10, sendo o valor do primeiro termo igual a 0. As linhas 23 a 41, preenchem as matrizes  $A$  e  $C$  da equação 3.34. As linhas 42, 43, 65 e 66 calculam os valores da segunda derivada da trajetória, considerando que exista uma função *inversa* e uma função *transposta* que retornam, respectivamente, a inversa e a transposta de uma matriz passada como parâmetro e, considerando que o operador  $*$ , nestas linhas, sejam operadores de multiplicação de matrizes. Os operadores  $\cup$  das linhas 43 e 66 são utilizados para representar uma operação de inserção do valor 0, no início e no final do vetor  $d2sdt2$ , de acordo com as condições de contorno da *spline* natural. Por fim, as linhas 46 a 64 e 69 a 80, constroem os polinômios cúbicos referentes às funções *spline* das ordenadas e das abscissas da trajetória.

---

**Algoritmo:** Trajetória de caminho livre

---

```
01: função trajetoria_livre( $X$  vetor de real,  $Y$  vetor de real,  $tf$  real)
02: tipos
03:     registro RTRAJETORIA
04:     início
05:     |      $x, y, t$  : vetor de real
06:     fim
07: variaveis
08:      $n, periodo, inicio\_periodo, k, i$ : inteiro
09:      $valor\_periodo, ta$ : real
10:      $T, alfa, ds2dt2, a, b, c, d, t$ : vetor de real
11:      $A, C$ : matriz de real
12:      $trajetoria$ : RTRAJETORIA
```

```

13: inicio
14:    $n \leftarrow \text{tamanho}(X)$ 
15:    $\text{periodo} \leftarrow 0$ 
16:    $\text{valor\_periodo} \leftarrow 0$ 
17:   enquanto ( $\text{valor\_periodo} \leq tf$ ) faça
18:      $T[\text{periodo}] \leftarrow \text{valor\_periodo}$ 
19:      $\text{valor\_periodo} \leftarrow \text{valor\_periodo} + tf/(n - 1)$ 
20:      $\text{periodo} \leftarrow \text{periodo} + 1$ 
21:   fim
22:    $ta \leftarrow 0.01$ 
23:   para ( $k$  de 2 até  $n - 1$  passo 1) faça
24:      $\text{alfa}[1] \leftarrow T[k] - T[k - 1]$ 
25:      $\text{alfa}[2] \leftarrow 2 * (T[K + 1] - T[K - 1])$ 
26:      $\text{alfa}[3] \leftarrow T[k + 1] - T[k]$ 
27:     se ( $k > 2$ ) e ( $k < n - 1$ ) então
28:        $A[k - 1, k - 2] \leftarrow \text{alfa}[1]$ 
29:        $A[k - 1, k - 1] \leftarrow \text{alfa}[2]$ 
30:        $A[k - 1, k] \leftarrow \text{alfa}[3]$ 
31:     senão se ( $k = 2$ )
32:        $A[1, 1] \leftarrow \text{alfa}[2]$ 
33:        $A[1, 2] \leftarrow \text{alfa}[3]$ 
34:     senão
35:        $A[n - 2, k - 2] \leftarrow \text{alfa}[1]$ 
36:        $A[n - 2, k - 1] \leftarrow \text{alfa}[2]$ 
37:     fim
38:      $C[k - 1, k - 1] \leftarrow 1/\text{alfa}[1]$ 
39:      $C[k - 1, k] \leftarrow -1/\text{alfa}[1] - 1/\text{alfa}[3]$ 
40:      $C[k - 1, k + 1] \leftarrow 1/\text{alfa}[3]$ 
41:   fim

```

```

42:    $d2sdt2 \leftarrow inversa(A) * 6 * C * transposta(X)$ 
43:    $d2sdt2 \leftarrow \{0\} \cup d2sdt2 \cup \{0\}$ 
44:    $periodo \leftarrow 0$ 
45:    $inicio\_periodo \leftarrow 0$ 
46:   para ( $k$  de 1 até  $n - 1$  passo 1) faça
47:      $a \leftarrow 1/6 / (T[k + 1] - T[k]) * (d2sdt2[k + 1] - d2sdt2[k])$ 
48:      $b \leftarrow 1/2 * d2sdt2[k]$ 
49:      $c \leftarrow (X[k + 1] - X[k]) / (T[k + 1] - T[k])$ 
50:      $c \leftarrow c - 1/6 * (T[k + 1] - T[k]) * (d2sdt2[k + 1] + 2 * d2sdt2[k])$ 
51:      $d \leftarrow X[k]$ 
52:      $valor\_periodo \leftarrow T[k]$ 
53:     enquanto ( $valor\_periodo \leq T[k + 1]$ ) faça
54:        $t[periodo] \leftarrow valor\_periodo$ 
55:        $valor\_periodo \leftarrow valor\_periodo + ta$ 
56:        $periodo \leftarrow periodo + 1$ 
57:        $trajetoria.t[periodo] \leftarrow t[periodo]$ 
58:     fim
59:     para ( $i$  de  $inicio\_periodo$  até  $periodo$ ) faça
60:        $x[i] \leftarrow a * (t[i] - T(k))^3 + b * (t[i] - T(k))^2 + c * (t[i] - T(k)) + d$ 
61:        $trajetoria.x[i] \leftarrow x[i]$ 
62:     fim
63:      $inicio\_periodo \leftarrow periodo$ 
64:   fim
65:    $d2sdt2 \leftarrow inversa(A) * 6 * C * transposta(Y)$ 
66:    $d2sdt2 \leftarrow \{0\} \cup d2sdt2 \cup \{0\}$ 
67:    $periodo \leftarrow 0$ 
68:    $inicio\_periodo \leftarrow 0$ 
69:   para ( $k$  de 1 até  $n - 1$  passo 1) faça
70:      $a \leftarrow 1/6 / (T[k + 1] - T[k]) * (d2sdt2[k + 1] - d2sdt2[k])$ 

```

---

```

71:       $b \leftarrow 1/2 * d2sdt2[k]$ 
72:       $c \leftarrow (Y[k + 1] - Y[k]) / (T[k + 1] - T[k])$ 
73:       $c \leftarrow c - 1/6 * (T[k + 1] - T[k]) * (d2sdt2[k + 1] + 2 * d2sdt2[k])$ 
74:       $d \leftarrow Y[k]$ 
75:      para (i de inicio_periodo até periodo passo 1) faça
76:           $y[i] \leftarrow a * (t[i] - T(k))^3 + b * (t[i] - T(k))^2 + c * (t[i] - T(k)) + d$ 
77:           $trajetoria.y[i] \leftarrow y[i]$ 
78:      fim
79:       $inicio\_periodo \leftarrow periodo$ 
80:  fim
81:  retorne(trajetoria)
82: fim

```

---

A função *trajetoria\_comportamentos*, cujo algoritmo é mostrado abaixo, retorna para a estrutura *trajetoria* os dados de uma trajetória baseada em comportamentos. As linhas 17 a 46 geram os comportamentos e os discretizam em pontos, cujos valores são armazenados nos vetores *X\_c* e *Y\_c*. O número de pontos discretos, para cada comportamento, é determinado pela variável *npontos*. Os comportamentos são gerados nas linhas 25 a 29, através das funções *comportamento\_linear* e *comportamento\_elipsoidal*. A discretização dos comportamentos é feita nas linhas 37 a 45. A linha 47 gera uma trajetória de caminho livre sobre os pontos discretos.

---

**Algoritmo:** Trajetória baseada em comportamentos

---

```

01: função trajetoria_comportamentos(X vetor de real, Y vetor de real, tf real)
02: tipos
03:     registro RTRAJETORIA
04:     inicio
05:          $x, y, t$  : vetor de real
06:     fim
07: variaveis

```

```
08:   n, ncomportamentos, npontos, k, cont, parametro: inteiro
09:   tf_c, passo_c, ti, tempo: real
10:   X_p, Y_p, X_c, Y_c: vetor de real
11:   trajetoria: RTRAJETORIA
12: inicio
13:   ncomportamentos ← tamanho(X)/4
14:   npontos ← 8
15:   tf_c ← tf/ncomportamentos
16:   k ← 0
17:   para (cont de 1 até ncomportamentos passo 1) faça
18:     |   parametro ← (cont - 1) * 4 + 1
19:     |   X_p[0] ← X[parametro + 1]
20:     |   X_p[1] ← X[parametro + 2]
21:     |   X_p[2] ← X[parametro + 3]
22:     |   Y_p[0] ← Y[parametro + 1]
23:     |   Y_p[1] ← Y[parametro + 2]
24:     |   Y_p[2] ← Y[parametro + 3]
25:     |   se (parametro = 0) então
26:     |   |   trajetoria ← comportamento_linear(X_p, Y_p, tf_c)
27:     |   senão
28:     |   |   trajetoria ← comportamento_elipsoidal(X_p, Y_p, tf_c)
29:     |   fim
30:     |   se (cont = 1) então
31:     |   |   passo_c ← tf/ncomportamentos/npontos
32:     |   |   ti ← 0
33:     |   senão
34:     |   |   passo_c ← tf/ncomportamentos/(npontos - 1)
35:     |   |   ti ← passo_c
36:     |   fim
```

```
37:     para (tempo de ti até tf_c passo passo_c) faça
38:          $n \leftarrow 0$ 
39:         enquanto (trajetoria.t[n] <= tempo) faça
40:              $n \leftarrow n + 1$ 
41:         fim
42:          $X\_c[k] \leftarrow trajetoria.x[n]$ 
43:          $Y\_c[k] \leftarrow trajetoria.y[n]$ 
44:          $k \leftarrow k + 1$ 
45:     fim
46: fim
47:  $trajetoria \leftarrow trajetoria\_livre(X\_c[k], Y\_c[k], tf)$ 
48: retorne(trajetoria)
49: fim
```

---

Os algoritmos das funções *comportamento\_linear* e *comportamento\_elipsoidal* são mostrados abaixo. Note que a diferença da função *comportamento\_linear* e *trajetoria\_linear* está nas linhas 20 e 21, onde a variação no tempo, da trajetória formada pelo comportamento, é linear, representada por  $t/tf$ . O mesmo acontece com as funções *comportamento\_elipsoidal* e *trajetoria\_elipsoidal*, onde a diferença está na linha 21.

---

**Algoritmo:** Comportamento linear

---

```
01: função comportamento_linear( $X$  vetor de real,  $Y$  vetor de real,  $tf$  real)
02: tipos
03:     registro RTRAJETORIA
04:     inicio
05:          $x, y, t$  : vetor de real
06:     fim
07: variáveis
08:      $x\_ini, y\_ini, x\_fim, y\_fim, t, ta$ : real
09:      $n$ : inteiro
```

```
10:   trajetoria: RTRAJETORIA
11: inicio
12:    $x\_ini \leftarrow X[1]$ 
13:    $x\_fim \leftarrow X[2]$ 
14:    $y\_ini \leftarrow Y[1]$ 
15:    $y\_fim \leftarrow Y[2]$ 
16:    $t \leftarrow 0$ 
17:    $ta \leftarrow 0.01$ 
18:    $n \leftarrow 1$ 
19:   enquanto ( $t \leq tf$ ) faça
20:      $x \leftarrow x\_ini + (t/tf) * (x\_fim - x\_ini)$ 
21:      $y \leftarrow y\_ini + (t/tf) * (y\_fim - y\_ini)$ 
22:      $t \leftarrow t + ta$ 
23:      $n \leftarrow n + 1$ 
24:      $trajetoria.x[n] \leftarrow x$ 
25:      $trajetoria.y[n] \leftarrow y$ 
26:      $trajetoria.t[n] \leftarrow t$ 
27:   fim
28:   retorne(trajetoria)
29: fim
```

---

---

**Algoritmo:** Comportamento elipsoidal

---

```
01: função comportamento_elipsoidal( $X$  vetor de real,  $Y$  vetor de real,  $tf$  real)
02: tipos
03:   registro RTRAJETORIA
04:   inicio
05:      $x, y, t$  : vetor de real
06:   fim
07: variáveis
```

```
08:   a, b, PI, t, ta: real
09:   n: inteiro
10:   trajetoria: RTRAJETORIA
11:  inicio
12:    $a \leftarrow X[1]$ 
13:    $b \leftarrow Y[1]$ 
14:    $alpha \leftarrow X[2]$ 
15:    $Dalpha \leftarrow Y[2]$ 
16:    $PI \leftarrow 3.14$ 
17:    $t \leftarrow 0$ 
18:    $ta \leftarrow 0.01$ 
19:    $n \leftarrow 1$ 
20:   enquanto ( $t \leq tf$ ) faça
21:      $r \leftarrow t/ta$ 
22:      $x \leftarrow a \cos(Dalpha * r + alpha)$ 
23:      $y \leftarrow b \sin(Dalpha * r + alpha)$ 
24:      $trajetoria.x[n] \leftarrow x + X[3]$ 
25:      $trajetoria.y[n] \leftarrow y + Y[3]$ 
26:      $trajetoria.t[n] \leftarrow t$ 
27:      $t \leftarrow t + ta$ 
28:      $n \leftarrow n + 1$ 
29:   fim
30:   retorne(trajetoria)
31: fim
```

---

## IMPLEMENTAÇÃO DOS MODELOS CINEMÁTICO E DINÂMICO E DOS CONTROLADORES DE TRAJETÓRIA, DO AXEBOT, NO SIMULINK

Os modelos cinemático e dinâmico e os controladores de trajetória foram construídos no *Simulink*, para a simulação e obtenção dos resultados. Um módulo de realidade virtual, com a utilização do *virtual reality toolbox* do *Simulink*, foi usado para a observação do comportamento do robô através de uma animação em três dimensões. Este módulo interpreta a linguagem VRML (*Virtual Reality Markup Language*), na qual são descritos todos os componentes gráficos do Axebot.

A figura B.1 mostra os blocos dos controladores de trajetória, sua conexão com a interface de geração de trajetória e o módulo de realidade virtual, além dos modelos cinemático e dinâmico, implementados no *simulink*.

Detalhes sobre os blocos dos modelos cinemático e dinâmico são encontrados nas figuras B.2 e B.3, respectivamente. Os blocos dos controladores são mostrados em detalhes nas figuras B.4, B.5 e B.6. A figura B.7 mostra visões do Axebot, geradas pelo bloco de realidade virtual.

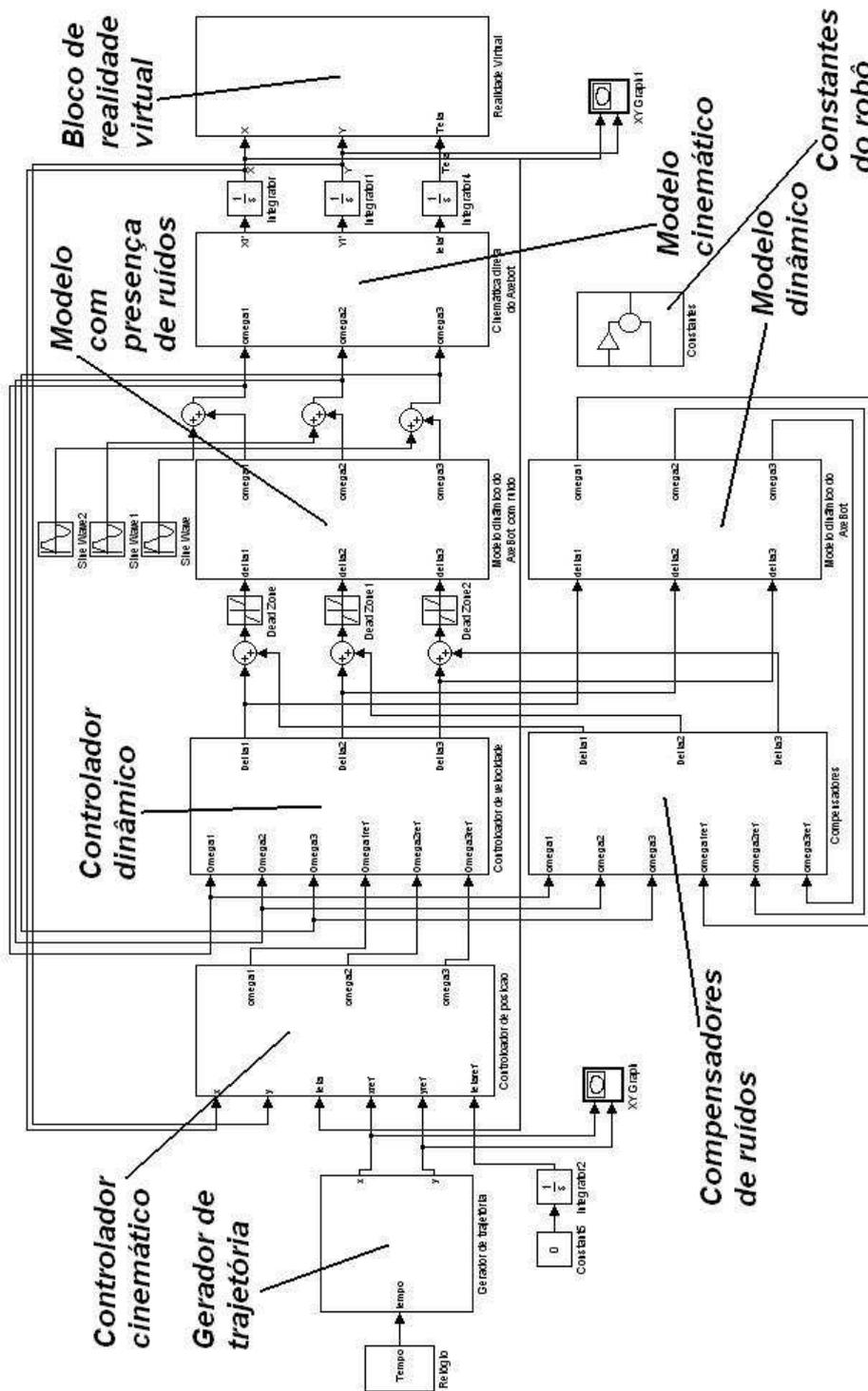


Figura B.1. Blocos, no *simulink*, da geração e controle de trajetória

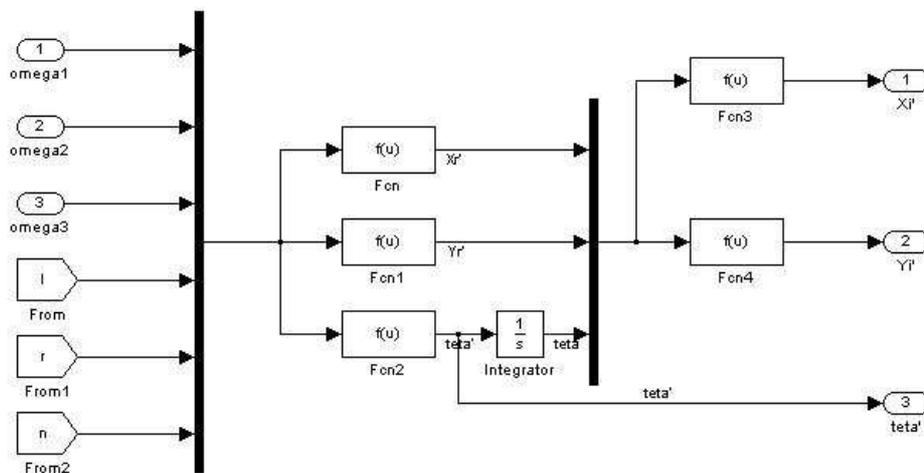


Figura B.2. Bloco do modelo cinemático em detalhes

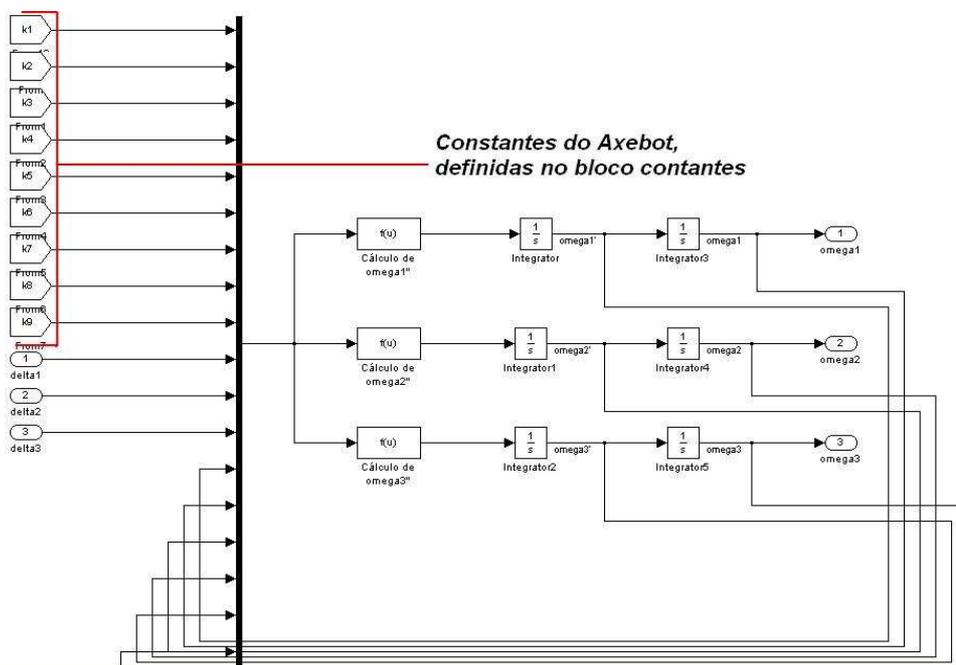


Figura B.3. Bloco do modelo dinâmico em detalhes

IMPLEMENTAÇÃO DOS MODELOS CINEMÁTICO E DINÂMICO E DOS CONTROLADORES DE TRAJETÓRIA, DO AXEBOT, NO SIMULINK

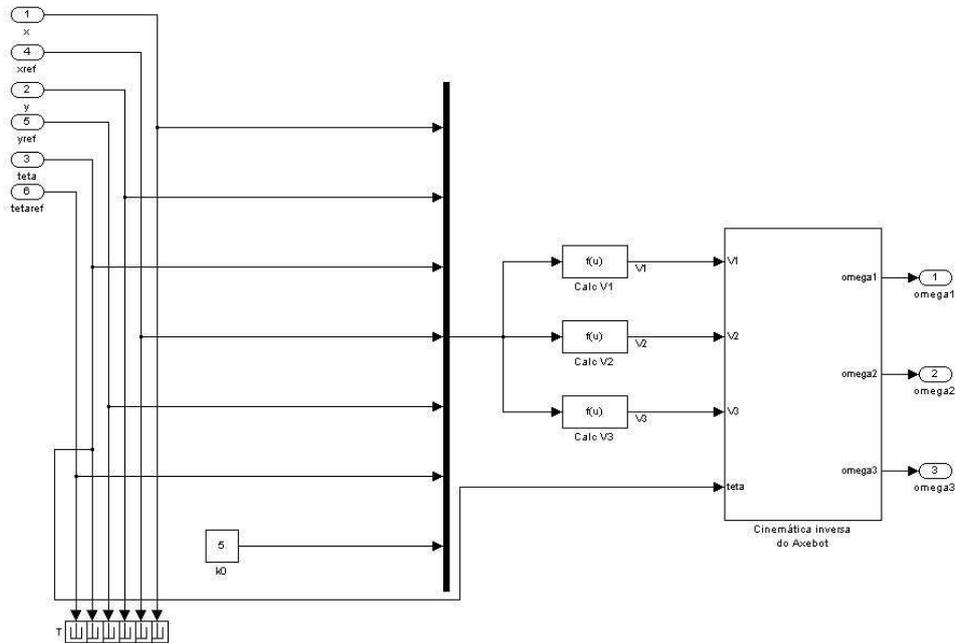


Figura B.4. Bloco do controlador cinemático em detalhes

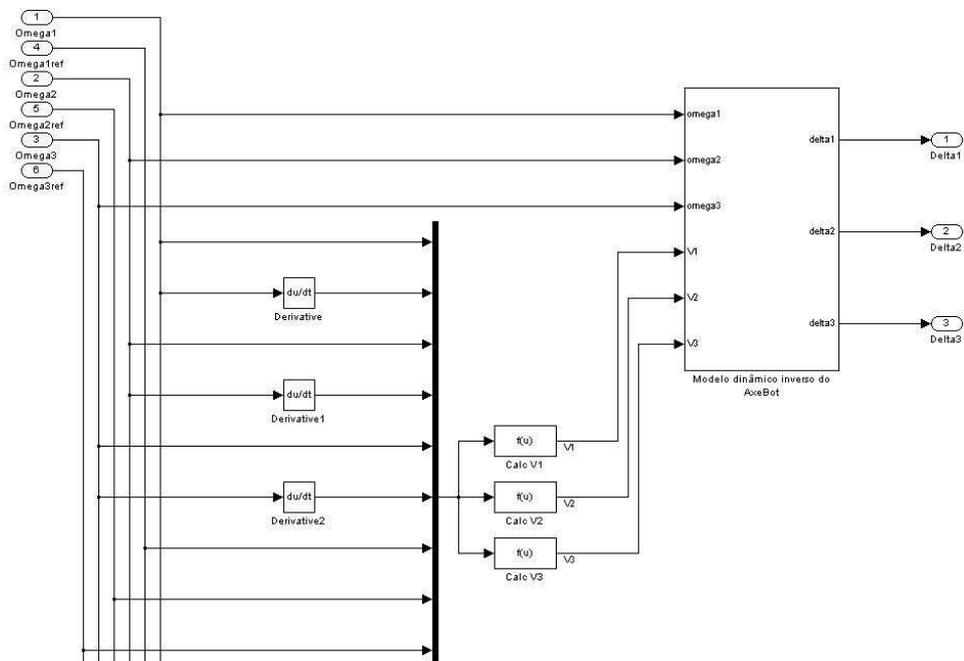


Figura B.5. Bloco do controlador dinâmico em detalhes

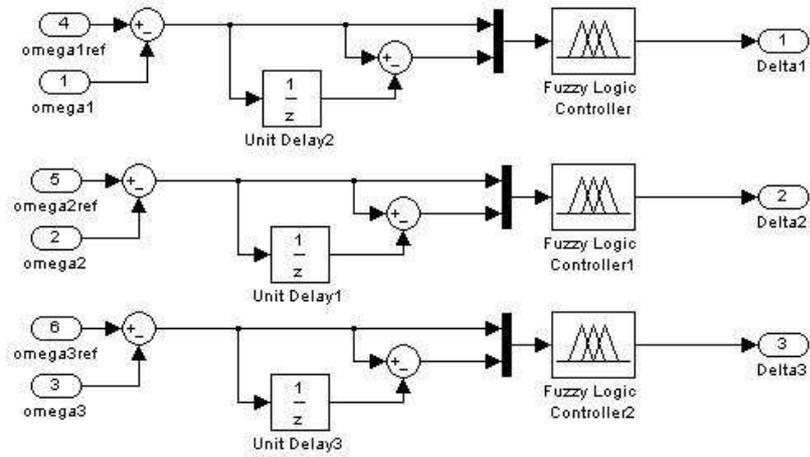


Figura B.6. Bloco dos compensadores de ruídos em detalhes



(a) Axebot de frente

(b) Axebot em movimento

Figura B.7. Visões do Axebot geradas pelo bloco de realidade virtual