

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Uma Proposta de Controle
Neural Adaptativo para o
Posicionamento de um Braço
Mecânico**

por

Ana Paula Lüdtke Ferreira

05225074

Dissertação submetida como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Paulo M. Engel
Orientador

Porto Alegre, junho de 1996.



UFRGS
Instituto de Informática
Biblioteca

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Ferreira, Ana Paula Lüdtke

Uma Proposta de Controle Neural Adaptativo para o Posicionamento de um Braço Mecânico / Ana Paula Lüdtke Ferreira.—Porto Alegre: CPGCC da UFRGS, 1996.

81 p.: il.

Dissertação (mestrado)—Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, 1996. Orientador: Engel, Paulo M.

Dissertação: Robótica, Redes Neurais, Controle Adaptativo

UFRGS		
INSTITUTO DE INFORMÁTICA		
BIBLIOTECA		
Nº CHAMADA:	Nº REE:	
62-526(043)	32694	
F3837	DATA:	
	20/11/96	
ORIGEM: 1	DATA:	VALOR:
	20/11/96	R\$ 20,00
FUNDO:	FORN.:	
IF	II	

Sumário

Lista de Figuras	5
Lista de Tabelas	6
Resumo	7
Abstract	8
1 Introdução	9
1.1 Inteligência e Comportamento Adaptativo	9
1.2 Robótica Inteligente	10
1.3 Controle de Movimentos	10
1.4 Proposta de Tese	11
1.4.1 Motivação	11
1.4.2 Avaliação do Sistema de Controle	12
1.5 Apresentação do Trabalho	13
2 Controle	15
2.1 Controle Biológico	15
2.2 Controle Adaptativo	16
2.3 Controle Neural	17
2.4 Controle Neural Adaptativo	19
2.5 Análise do Estado da Arte	20
3 A Rede “Attentional Mode”	21
3.1 Estado do Sistema	23
3.1.1 Restrições	23
3.2 Algoritmo de Adaptação	24
3.2.1 Erros Relativos ao Estado do Sistema	25
3.2.2 Cálculo do Gradiente do Erro	26
3.2.3 Função de Ativação	27
3.2.4 Atualização dos Pesos na Camada de Saída	28
3.2.5 Atualização dos Pesos nas Camadas Escondidas	29
4 Implementação do Modelo	31
4.1 Sistema	31
4.1.1 Modelo Geométrico	31
4.1.2 Modelo Cinemático	32
4.2 Controle Neural	34
4.2.1 Estrutura da Rede Neural	35
4.2.2 Parâmetros dos Erros Gerados pela Rede	35
4.2.2.1 Posição e Orientação	35
4.2.2.2 Restrições	36
4.3 Visualização	37
4.4 Ciclo de Controle	38
4.5 Complexidade Computacional	39
4.5.1 Complexidade do Algoritmo	39

4.5.2 Espaço de Memória	42
5 Resultados Obtidos	44
5.1 Métodos Teóricos <i>versus</i> Métodos Empíricos de Análise	44
5.2 Testes de Simulação	44
5.3 Convergência	45
5.4 Estabilidade	49
5.4.1 A Função de Ativação e os Valores dos Pesos	49
5.4.2 Variação dos Ângulos das Juntas	51
5.5 Repetitividade	51
5.6 Geração de Trajetórias	53
6 Conclusões	58
6.1 A aplicação do <i>controle neural adaptativo</i> no contexto deste trabalho	58
6.2 Discussão dos Resultados Obtidos	58
6.2.1 Complexidade do Sistema de Controle	59
6.2.2 Convergência	59
6.2.3 Estabilidade	60
6.2.4 Repetitividade	60
6.2.5 Geração de Trajetórias	61
6.2.6 Definição do Controlador	61
6.3 Trabalho Futuro	62
6.3.1 Desvio de Obstáculos	62
6.3.2 Dinâmica	62
6.3.2.1 Controle	62
6.3.2.2 Simulação	62
6.4 Considerações Finais	63
Anexo A Matrizes de Transformação	64
A.1 Matrizes de Rotação	64
A.1.1 Interpretação Geométrica	66
A.2 Matrizes de Transformação Homogêneas	67
A.2.1 Interpretação Geométrica	69
Anexo B Matrizes de Derivação	70
B.1 Aplicação ao Problema Tratado	71
Anexo C Fundamentos de Robótica	73
C.1 Geometria	73
C.2 Cinemática	73
C.3 Dinâmica	74
Anexo D GSE	75
D.1 Window Manager	75
D.2 Utilities	76
D.3 Objects	77
D.4 Environment	77
D.5 RoboWorld	77
Bibliografia	78

Lista de Figuras

FIGURA 1.1	Precisão e repetitividade	12
FIGURA 3.1	A rede neural <i>Attentional Mode</i>	22
FIGURA 3.2	Controle neural de malha fechada	23
FIGURA 3.3	A Função <i>tangente hiperbólica</i>	28
FIGURA 4.1	Módulos de implementação do sistema	32
FIGURA 4.2	Braço implementado	33
FIGURA 4.3	Evolução do parâmetro de orientação do sistema de controle	36
FIGURA 4.4	Evolução do valor da restrição de relaxamento	37
FIGURA 5.1	Distribuição dos erros relativos à convergência	47
FIGURA 5.2	Oscilação no Sistema de Controle	50
FIGURA 5.3	Minimização do Problema de Oscilação	52
FIGURA 5.4	Geração de trajetórias do sistema	54
FIGURA 5.5	Exemplo de trajetória gerada pela rede — I	55
FIGURA 5.6	Exemplo de trajetória gerada pela rede — II	56
FIGURA 5.7	Exemplo de trajetória gerada pela rede — III	57
FIGURA C.1	Relação entre os problemas cinemáticos inverso e direto . . .	74
FIGURA D.1	Interfaces do ambiente gráfico de simulação <i>GSE</i>	76

Lista de Tabelas

TABELA 4.1	Modelo geométrico do braço implementado	32
TABELA 5.1	Configuração do sistema de controle neural	45
TABELA 5.2	Convergência do sistema de controle	47
TABELA 5.3	Repetitividade do sistema de controle	53

Resumo

Inicialmente concebidos para operar em ambientes industriais fechados, os robôs vêm se tornando cada vez mais difundidos na sociedade. Suas atribuições não se limitam mais exclusivamente à execução de simples tarefas de repetição, mas à uma interação efetiva com o mundo em que se inserem. Para atingir tal objetivo, estes robôs devem possuir um controlador flexível, capaz de adaptar-se continuamente ao mundo dinâmico que o cerca.

A maioria das soluções para o posicionamento de um braço manipulador funcionam através do mapeamento de posições/orientações espaciais e de configurações das juntas do braço. Uma vez que a função cinemática direta não possui inverso global, diversas restrições devem ser adicionadas ao sistema de modo a diminuir a quantidade de soluções possíveis. Este tipo de controlador não é flexível, uma vez que qualquer modificação no estado do sistema pode tornar o controlador inútil. Este fato obriga a introdução de novos paradigmas na programação de robôs.

Redes neurais possuem a capacidade de solucionar problemas não-lineares que, de outra forma, tornam-se muito difíceis de tratar matematicamente. Devido à natureza altamente não-linear do controle de um braço manipulador articulado, seja da parte cinemática ou dinâmica do processo, redes neurais vêm sendo utilizadas sistematicamente na definição de sistemas de controle robóticos. Porém, apesar da grande versatilidade das redes neurais, estas têm sido, em grande parte, utilizadas apenas como sistemas de mapeamento não-linear. Tanto nos problemas cinemáticos quanto dinâmicos, existe um processo de treinamento onde a rede armazena diversos estados possíveis para o sistema e, após este processo, busca as soluções previamente armazenadas na rede. Contudo, esta abordagem não é a mais adequada para sistemas abertos, ou seja, sistemas que não são completamente conhecidos e que podem sofrer transformações no decorrer do seu funcionamento. Este fato leva a que soluções armazenadas para uma determinada configuração do sistema não funcionem para outras configurações.

Este trabalho apresenta uma estratégia de controle neural adaptativo para o posicionamento de um braço de robô no espaço. Diferentemente das abordagens tradicionais, não existe um processo de treinamento da rede, mas sim uma contínua adaptação do braço de modo a se aproximar da localização espacial (posição e orientação) desejada. Desta forma, qualquer que seja o estado corrente do ambiente no qual o sistema robótico esteja inserido, este é capaz de encontrar uma solução adaptativamente, sem as limitações impostas por configurações de braço previamente armazenadas.

TITLE: "Positioning a Robot Arm using an Adaptive Neural Controller"

Abstract

Initially conceived to work inside closed industrial environments, robots are becoming part of our everyday lives. They are not demanded to execute repeated simple tasks anymore, but to interact with the world around them in an efficient and intelligent way. In order to achieve this goal, those robots must have a flexible controller, capable of adapting itself to a dynamic world.

The majority of solutions to position a robot arm try to map a spatial position and orientation to a joint configuration. Since the forward kinematics function has no global inverse, several constraints must be added in order to prune the solution space, and the arm position will be restricted to the one previously mapped as the problem solution. This is not a flexible solution because any obstacle in the way will turn this approach useless. This fact obliges us to use new paradigms when programming robots, because known control techniques are, most of the time, no longer suitable.

The problem of positioning a robot arm in the three-dimensional space has been studied for a long time. However, most solutions developed until now, despite the fact of providing great reliability and accuracy, lack the necessary flexibility to permit the arm to move in an open environment. Most problems to be solved by a robot arm in uncontrolled environments are mostly like the ones we solve in a daily basis, such as pick and place tasks. Those tasks don't necessarily need the accuracy provided by the known methods to position an arm, but they do need the degree of adaptivity and flexibility that humans possess.

In this thesis we will present a neural adaptive approach to solve the problem of positioning a robot arm in the space. This method works by incorporating the state of the system into the network. The network input is the current state of the system (the current arm position and orientation) and the outputs are the changes in the state variables (the joint values) in order to approximate the current state to the desired one. This is a closed-loop neural control scheme and it is done in real time without needing any previous training phase.

Capítulo 1

Introdução

1.1 Inteligência e Comportamento Adaptativo

A capacidade humana de raciocínio abstrato e de expressão através de linguagem geralmente é considerada a definição de inteligência. Existe uma longa tradição filosófica que venera o conhecimento e a racionalidade humana como sendo estes os atributos que definem um ser inteligente. Sob este prisma, a inteligência é comumente equacionada com o *raciocínio deliberado*, ou seja, a habilidade de conscientemente raciocinar sobre algum problema.

Segundo esta visão tradicional, os mecanismos que suportam um comportamento inteligente consistem em processos que envolvem a manipulação de representações mentais do mundo. Tipicamente, estas representações são vistas como símbolos lingüísticos que substituem as entidades do mundo real que elas representam e a manipulação destes símbolos é vista como uma espécie de computação.

A idéia da representação mental certamente está de acordo com o senso comum a respeito do comportamento humano. Regularmente, entidades mentais como idéias, desejos e percepções são invocados para racionalizar as ações humanas. Entretanto, o fato de uma determinada maneira de ver o mundo ser útil sob certas circunstâncias não implica necessariamente que esta seja uma explicação científica completa. A questão fundamental a respeito desta hipótese representacional não é se ela é óbvia intuitivamente ou se é fundamentalmente falha, mas sim em que contextos ela fornece uma explicação suficiente e se ela é capaz de explicar uma grande gama de fenômenos que podem ser associados com comportamento inteligente.

A capacidade de um agente de interagir e se adaptar às condições do ambiente que o cerca é chamada *comportamento adaptativo* e cada vez mais é considerada a característica principal de um comportamento inteligente [BEE 90]. Somente a partir da interação com o mundo é que um agente é capaz de desenvolver suas habilidades e de aumentá-las de acordo com sua conveniência. Linguagem e raciocínio abstrato estão certamente entre os atributos que nos tornam inteligentes; entretanto, estas características apareceram relativamente tarde no cenário evolutivo. Existem fundações e desenvolvimentos prévios através de toda a linhagem humana. Muito deste substrato está presente em outros animais hoje em dia. O estudo deste substrato pode fornecer pistas de como o nível de raciocínio humano pode ser organizado. Funções cognitivas superiores são atributos mais elaborados desta competência mais básica de interagir com o mundo que nos cerca.

Uma vez estabelecida a versatilidade do comportamento inteligente, o primeiro problema a ser atacado é a interação sem restrições do sistema inteligente

com ambientes abertos. Somente desta maneira pode-se ter a certeza de que não será simplificado o mais fundamental dos problemas enfrentados por um agente inteligente: a adaptação ao seu ambiente [BRO 91].

1.2 Robótica Inteligente

O estudo dos métodos e ferramentas de inteligência artificial aplicado à robótica surgiu a partir da necessidade de uma maior flexibilização dos sistemas robóticos existentes. Apesar de controlados por computadores, a maioria dos robôs em atividade industrial são utilizados como simples máquinas de posicionamento, trabalhando em um ambiente fechado e completamente conhecido. Em muitos casos, esta abordagem é suficiente, mas quando se pretende que um robô substitua um ser humano em tarefas realizadas em ambientes inóspitos a este, ou mesmo que trabalhe em um ambiente que não é previamente conhecido – como qualquer ambiente que os humanos freqüentam no seu dia-a-dia – um outro tipo de abordagem torna-se necessária.

Além destes fatores, a robótica é talvez a disciplina mais interessante do ponto de vista do desenvolvimento de sistemas inteligentes. Dadas as suas características, sistemas robóticos são ideais para o teste de idéias relacionadas à inteligência.

1.3 Controle de Movimentos

O controle de movimentos é possivelmente a função mais importante do sistema nervoso, uma vez que um organismo não pode existir sem ele. Os movimentos do corpo humano requerem a coordenação de ações entre as diversas áreas do córtex cerebral, cerebelo, gânglios basais e medula espinhal, bem como nervos periféricos e receptores sensoriais [COR 94] – em outras palavras, grande parte do sistema nervoso. Juntas, estas estruturas neurais controlam um aparato biomecânico altamente complexo constituído de músculos e ossos.

Apesar de sua complexidade e importância no desenvolvimento de todos os animais superiores, em particular do ser humano, o controle de movimentos raramente é visto como uma característica inteligente. Isso é válido mesmo levando-se em consideração que este controle permite aos seres vivos interagir com o mundo complexo e imprevisível que os cerca, adaptando-se a esse de modo a conseguir não apenas a sobrevivência, mas também o aprendizado sobre a forma de lidar com ele de maneira a poder evoluir.

O controle de movimentos é abordado neste trabalho como a função básica que um sistema robótico deve possuir. Este controle, para poder ser considerado inteligente, deve possuir algumas características básicas. As suposições feitas neste trabalho sobre o sistema de controle são as seguintes:

- **ambiente aberto** — O sistema, constituído pelo braço manipulador, deve operar num ambiente aberto, ou seja, num ambiente que não é previamente conhecido e que pode variar no tempo.
- **adaptatividade** — O sistema deve adaptar-se ao seu ambiente e às mudanças deste, e tais mudanças devem poder ser levadas em consideração pelo sistema de controle.

- **flexibilidade** — O sistema deve ser flexível, isto é, possuir a capacidade de resolver diferentes problemas de forma automática, sem a necessidade de reprogramação ou de modificação do controlador.

1.4 Proposta de Tese

Este trabalho introduz uma estratégia de controle de movimentos baseada no conceito de adaptatividade, apresentando uma proposta de controlador neural para o controle de movimentos de um braço manipulador. Este controlador permite que o controle do braço seja realizado em tempo real, possibilitando a realização de qualquer tarefa e que, para tanto, o estado global do sistema seja levado em consideração. Desta forma, o sistema pode adaptar-se às transformações do meio ambiente e funcionar de acordo com elas.

1.4.1 Motivação

A necessidade de existirem sistemas de controle capazes de adaptarem-se permanentemente e eficientemente ao ambiente em que estão inseridos determina a realização de um volume significativo na pesquisas na área de sistemas adaptativos. Tais sistemas são capazes de perceber modificações no seu ambiente de trabalho e modificar seus parâmetros de forma a continuarem operando com eficácia sem prejuízo de nenhuma de suas características.

Aliada à necessidade de adaptação, os sistemas de controle devem possuir alguma tolerância a ruído de informação. Falhas eventuais na aquisição de informação não devem interferir significativamente na tarefa em execução. Redes neurais, além de apresentarem características de tolerância a ruído, são capazes de resolver problemas altamente não-lineares, como é o caso de sistemas de controle robótico.

Os sistemas adaptativos, em geral, por forma a resolverem problemas com características de não-linearidade, necessitam de uma quantidade muito grande de parâmetros, cujos valores devem ser adequadamente atribuídos, permitindo que estes sistemas funcionem conforme o esperado. Tal fato torna, não somente o modelamento, mas a própria implementação de tais sistemas consideravelmente complicada, uma vez que, sendo os estados futuros do ambiente desconhecidos, os valores destes parâmetros são difíceis de estimar.

As redes neurais, por sua vez, são capazes de aprender novos valores de parâmetros a partir de novos estados do sistema, embora muitas vezes estes estados devam ser previamente apresentados à rede durante uma fase de treinamento. Como as redes neurais possuem a capacidade de interpolar e de extrapolar conhecimento, este processo de treinamento não é considerado problemático na maior parte das aplicações que utilizam redes neurais.

Para que um sistema possa controlar um braço robótico num ambiente aberto, uma vez que os possíveis estados do ambiente são inúmeros e bastante diferentes entre si, é necessária a adaptação constante do sistema de controle ao seu ambiente, sendo que esta adaptação deve ser realizada em tempo real. A modificação dos parâmetros do sistema por uma rede neural capaz de perceber o estado do ambiente, de forma a adaptar o sistema a funcionar diante das novas circunstâncias, torna os *sistemas de controle neurais adaptativos* ideais para o controle de processos que funcionem em ambientes que não podem ser completamente ou previamente

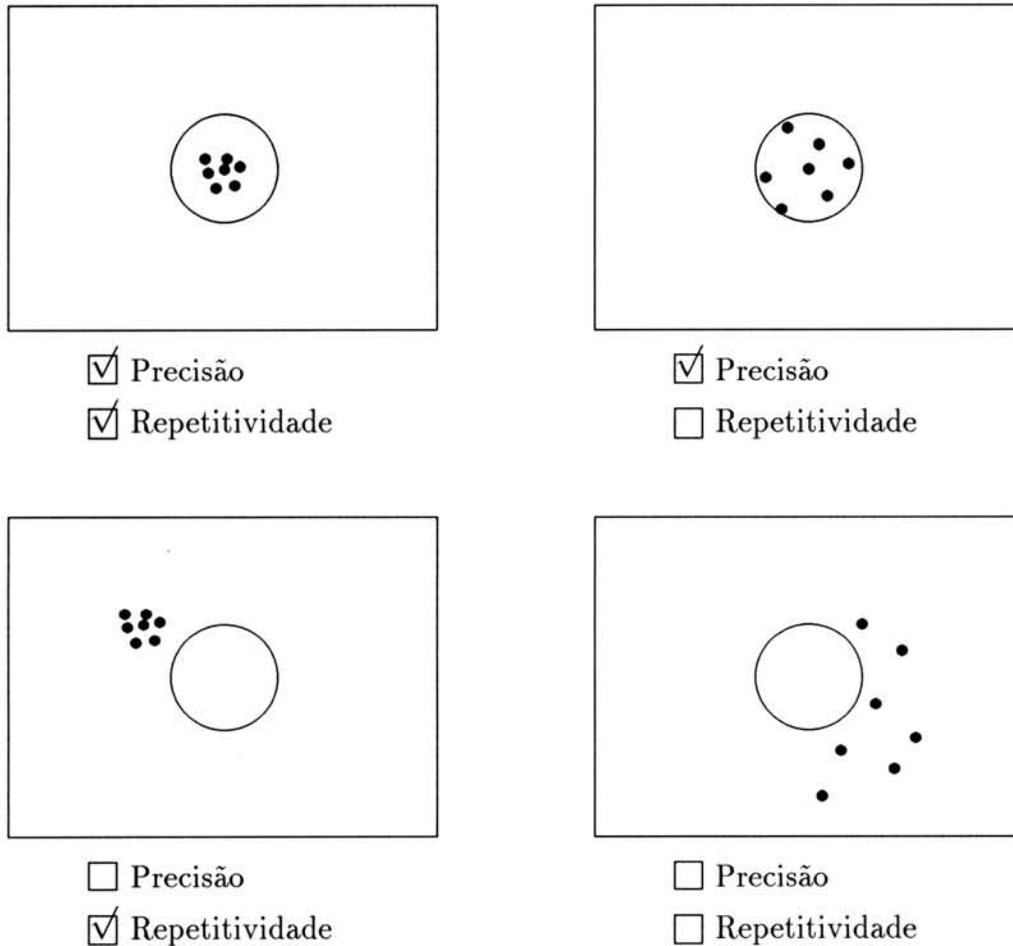


FIGURA 1.1 — Precisão e repetitividade

especificados.

1.4.2 Avaliação do Sistema de Controle

A avaliação de um sistema de controle é fundamental para o estabelecimento de suas capacidades e limitações. Existem duas propriedades principais que avaliam um braço robótico de modo a considerá-lo adequado ou não para a execução de uma variada gama de tarefas. Estas propriedades são suas capacidades de *convergência* e *repetitividade*. A *convergência*, ou *precisão*, diz respeito à capacidade do braço de atingir um alvo com um erro pequeno em qualquer circunstância. Já a *repetitividade*, ou *exatidão* refere-se à capacidade do braço de conseguir resultados semelhantes na execução de uma mesma tarefa, sejam quais forem as condições iniciais. A figura 1.1 mostra a diferença entre estas duas propriedades.

Outra importante propriedade do sistema de controle é a sua *estabilidade*, ou seja, a capacidade do braço aproximar-se de seu objetivo de modo contínuo, sem oscilações ao longo deste processo.

Estas três propriedades serão levadas em consideração quando da avaliação do controlador proposto nesta dissertação, e podem ser modeladas matematicamente de acordo com o que se segue:

- **convergência** — O sistema deve sempre convergir para a solução desejada, ou seja, deve existir um valor δ tal que, para todo o objetivo O e solução encontrada S pelo sistema, tenha-se sempre

$$|O - S| \leq \delta$$

- **estabilidade** — O sistema de controle deve ser estável, ou seja, convergir para a solução desejada sem grandes oscilações no estado do processo controlado. Dado um tempo t_o , existe sempre um valor δ tal que, sendo O o objetivo do sistema e $S(t)$ o estado do sistema após t unidades de tempo,

$$|O - S(t)| \leq \delta \quad t \geq t_o$$

- **repetitividade** — O controlador deve possuir coerência ao longo do tempo, isto é, repetir o desempenho com relação a uma mesma tarefa. Seja M_{ε_T} e ε_T respectivamente o valor médio e o valor do erro correspondente a uma tarefa arbitrária T . Então, deve existir δ tal que para todo ε_T

$$|M_{\varepsilon_T} - \varepsilon_T| \leq \delta$$

1.5 Apresentação do Trabalho

O Capítulo 2 apresenta um estudo do *estado da arte* em controle de movimentos de braços manipuladores, suas principais técnicas, resultados, vantagens e desvantagens. Este estudo não pretende ser exaustivo, mas produzir um apanhado geral do trabalho realizado até hoje nesta área.

No Capítulo 3, será apresentado o modelo teórico no qual a tese foi baseada. Esse capítulo introduz formalmente a rede neural denominada *Attentional Mode*, utilizada para implementar o controlador do braço, bem como apresenta considerações teóricas a respeito do trabalho realizado.

O Capítulo 4 apresenta a descrição da implementação do sistema de controle. Nesse capítulo, serão apresentados os detalhes de implementação para o problema de posicionamento de um manipulador com múltiplos graus de liberdade no espaço.

Os resultados produzidos pelo sistema de controle implementado são apresentados e discutidos no Capítulo 5, constituindo a etapa de validação do modelo.

O Capítulo 6 apresenta as conclusões obtidas acerca do experimento realizado. São ainda apresentadas sugestões para a melhoria do sistema de controle implementado, bem como outros aspectos que poderiam ser abordados em relação ao problema tratado.

Esta dissertação apresenta ainda quatro apêndices que fornecem informações relevantes a respeito do trabalho desenvolvido. O Apêndice A desenvolve a teoria básica sobre matrizes homogêneas de transformação de referenciais, suas propriedades e interpretação geométrica. Esse apêndice torna claro como foi construído o modelo cinemático do robô, bem como a maneira com que os erros referentes ao problema tratado foram calculados.

O Apêndice B introduz uma técnica sistemática de cálculo de derivadas de expressões referentes à transformações de referenciais. Esta técnica permite que o algoritmo de descida do gradiente seja utilizado na rede neural implementada de

uma maneira automática, sem necessidade de reestruturação do algoritmo no caso de modificações estruturais no braço manipulador.

O Apêndice C apresenta superficialmente os modelos matemáticos utilizados na definição de um braço de robô.

O Apêndice D introduz o *GSE*, (*Graphic Simulation Environment*), que é um ambiente desenvolvido para o teste e simulação de sistemas de controle robóticos. O *GSE* foi utilizado nos processos de simulação envolvidos neste trabalho.

Capítulo 2

Controle

Com o conhecimento da cinemática e dinâmica do braço manipulador, objetiva-se controlar as juntas do robô de modo a este seguir um determinado caminho previamente escolhido. Este controle pode ser realizado através de torques aplicados às suas juntas, ao que chamaremos *controle dinâmico* ou através da simples modificação dos valores destas, aqui denominado *controle cinemático*.

Em geral, o controle de movimentos consiste em dois problemas:

1. Obtenção dos modelos *geométrico*, *cinemático* e *dinâmico* do braço.
2. Utilização destes modelos para a determinação de leis ou estratégias de controle para obtenção da resposta ou do desempenho desejado para o sistema.

Neste capítulo, serão descritas algumas técnicas de controle referentes à segunda parte deste problema. O foco principal será centrado nos sistemas de controle *biológico*, *adaptativo*, *neural* e *neural adaptativo*, que possuem relação com o trabalho desenvolvido nesta tese. Para um estudo mais geral sobre os métodos de controle de movimento de um braço mecânico, Fu *et.al.* [FEA 94] introduz uma análise mais completa. A solução da primeira parte do problema posto acima é descrita no Apêndice C.

2.1 Controle Biológico

Em qualquer momento, nosso cérebro é capaz de controlar até 244 diferentes graus de liberdade mecânicos, envolvendo mais de 600 diferentes músculos corporais [RIT 92]. Na realidade, freqüentemente, dúzias de músculos distintos agem simultaneamente. Por exemplo, os músculos de cada braço ou perna controlam 30 graus de liberdade mecânicos, utilizando combinações musculares bastante complexas. Este controle é realizado através da realimentação de uma enorme variedade de sensores: sensores de pressão e estiramento nos músculos, sensores de tato na pele, sensores nas juntas e, freqüentemente essencial, o sistema visual. Subjacente a essas faculdades, está o vasto poder de coordenação do cérebro, do qual nossos conhecimentos são ainda bastante limitados. A complexidade deste controle torna-se óbvia logo que se tenta equipar um braço mecânico com apenas uma pequena fração da destreza humana.

O sistema nervoso central humano compõe o modelo dinâmico dos movimentos do corpo de uma forma gradual [SHA 94]. Este modelo serve para a predição e compensação de forças impostas pelo ambiente e é adaptado continuamente para se

adequar ao ambiente ao redor. É importante notar que o sistema nervoso central *não mantém registros de modelos de situações ambientais anteriores*. Uma mudança no ambiente provoca um descontrole inicial no controle de movimentos, seguido de um curto período de adaptação à nova situação. Se o ambiente sofrer novas modificações, por forma a voltar ao estado anterior, o período de adaptação vai novamente existir, ou seja, o sistema nervoso adapta-se continuamente ao estado presente do ambiente e não através de algum tipo de *look-up table*. O sistema nervoso mantém ainda um planejamento cinemático independente das condições dinâmicas do ambiente, isto é, as trajetórias traçadas pelos membros humanos são semelhantes entre si, mesmo em diferentes condições dinâmicas.

O objetivo final do controle de movimentos de um braço mecânico é a construção de um sistema de desempenho robusto num ambiente complexo e mutável, e não a obtenção da cópia pura e simples de sistemas biológicos. Sistemas biológicos desenvolveram-se sobre sistemas previamente existentes, de maneira a adequarem-se às mudanças no meio, conduzindo a que estes sistemas não sejam provavelmente muito otimizados, pois possivelmente restam diversas estruturas desnecessárias dentro deles. Entretanto, conceitos de sistemas de controle presentes na natureza não devem ser desprezados. Isto é particularmente verdade em diversos problemas de planejamento de movimentos e controle, em que sistemas biológicos são extremamente eficazes.

2.2 Controle Adaptativo

Um sistema de controle adaptativo é aquele que mede, de modo contínuo, o estado do ambiente em que está inserido, compara-o com o objetivo do controle e usa a diferença verificada para variar parâmetros ajustáveis do sistema por forma a garantir que o desempenho deste possa ser mantido independentemente das mudanças ambientais. Alternativamente, tal sistema pode avaliar continuamente seu próprio desempenho e modificar, se necessário, seus próprios parâmetros. Um sistema de controle adaptativo deve possuir, portanto, características de auto-organização.

Atualmente existe um grande esforço de investigação científica sendo conduzida na área de sistemas adaptativos. Diversas técnicas, já consideradas clássicas, são descritas em literatura [FUU 87]. Tais técnicas levam em consideração a mudança de carga numa tarefa, permitindo um modelamento dinâmico mais acurado e, conseqüentemente, uma melhoria na resposta do sistema. As pesquisas mais recentes em controle adaptativo levam em consideração não apenas a resposta dinâmica do sistema, mas sua resposta como um todo. Esta linha de trabalho é chamada de *comportamento adaptativo*.

Beer [BEE 90] argumenta que o comportamento adaptativo é a mais fundamental característica dos seres inteligentes e que este comportamento deve emergir do *hardware* no qual o agente está implementado. Ele desenvolveu uma arquitetura capaz de levar um agente móvel a ter diversos comportamentos diferentes de acordo com a situação em que se encontra, aprendendo ainda novos comportamentos a partir dos existentes. Outro trabalho relevante foi desenvolvido por Mataric [MAT 94, MAT 92] na área de controle baseado em comportamento (*behavior-based control*) em que robôs móveis interagem com o ambiente ao mesmo tempo em que cooperam entre si na resolução de tarefas em conjunto. Estes trabalhos levantam questões teóricas bastante relevantes a respeito da interação e adaptação ao meio,

mas os processos controlados (robôs móveis) são relativamente simples de controlar, sem necessidade de um sistema capaz de tratar não-linearidades para um correto controle de movimentos.

Śmieja e Beyer vêm desenvolvendo um trabalho relacionado com a cooperação entre dois braços robóticos [KEL 94], ao mesmo tempo em que cada braço funciona de maneira a reconhecer o meio e a movimentar-se de acordo com o ambiente no qual estão inseridos [SMI 94a, SMI 94b, SMI 94c]. Esta adaptação dá-se por meio de uma sociedade de agentes, bastante inspirada na proposta por Minsky [MIN 86], em que cada agente é responsável por seu próprio desempenho e avaliação. O controle funciona através da negociação, pelos agentes envolvidos, sobre quem tem prioridade na execução da tarefa em questão. Uma vez que os agentes são capazes de realizar sua própria avaliação, esta tarefa não impõe dificuldades. O problema é que estes agentes somente são capazes de aprender fora do sistema. Uma vez inseridos neste, avaliam suas capacidades de acordo com o seu próprio conhecimento, que, a partir do momento em que são inseridos no sistema, não é atualizado. Além disso, cada agente leva apenas em consideração a sua própria contribuição para atingir o objetivo, quando muitas vezes, a cooperação entre agentes torna-se necessária.

Um dos principais incentivadores da pesquisa em controle adaptativo para robôs, Brooks [BRO 91] argumenta que nenhum agente é capaz de aprender se não for capaz de interagir de uma forma real com seu ambiente. Para tanto, este agente, além de possuir capacidade de aprendizado, deve também possuir uma estrutura física que permita a ele movimentar-se e efetivamente entrar em contato com o mundo. No que diz respeito ao controle de movimentos, isto é particularmente verdade, uma vez que, sem a possibilidade de interação física e avaliação interna, nenhum agente é capaz de perceber seu próprio funcionamento no mundo que o cerca, avaliando e melhorando, desta forma, seu funcionamento dentro do contexto em que está inserido.

2.3 Controle Neural

O conhecimento atual sobre o controle de movimentos biológicos ainda é demasiado fragmentado para permitir a construção de um controlador robótico com bases biológicas capaz de competir com algoritmos de controle tradicionais. A abordagem do problema de controle robótico por redes neurais procura reproduzir estratégias de controle de movimentos em algoritmos neurais. Embora redes neurais sejam inspiradas em sistemas biológicos, não são um retrato fiel destes, nem possuem todas as suas capacidades. Apesar disso, existem um grande número de abordagens promissoras baseadas em conceitos de redes neurais.

Os modelos mais populares de redes neurais possuem duas fases distintas: *treinamento e classificação*. Num treinamento supervisionado, são apresentados à rede pares problema-solução, e um algoritmo de treinamento é utilizado para fazer a rede aprender o mapeamento correto destes pares. Já em um treinamento não-supervisionado, as amostras são apresentadas de forma a que a própria rede encontre uma relação entre elas. Na fase de classificação, uma amostra é apresentada à rede, e esta responde com o resultado aprendido na fase de treinamento.

Redes neurais têm como características principais as capacidades de generalização e de extrapolação do conhecimento adquirido e do tratamento de problemas não lineares de uma forma bastante simples. Por esta razão, são sistematicamente

utilizadas em problemas de reconhecimento de padrões e qualquer outra espécie de mapeamento não-linear que, de outro modo, seria impossível de tratar algoritmicamente. Pode-se mostrar que uma rede neural com uma camada intermediária e um suficiente número de neurônios nesta camada é capaz de mapear qualquer função contínua [SON 94]. Em problemas de controle, redes neurais também fornecem uma grande contribuição. Uma vez que problemas de controle são altamente não-lineares, um tratamento matemático formal torna-se bastante problemático. Redes neurais são capazes de fornecer soluções de controle para qualquer sistema previsível de uma maneira muito mais simples.

O controle de movimentos de um braço manipulador em ambientes abertos, contudo, possui características que dificultam bastante o uso de redes neurais neste tipo de controle. Esta dificuldade provém basicamente da fase de treinamento da rede e do fato de que as mudanças no ambiente são, em princípio, imprevisíveis. Como se sabe, o problema cinemático inverso para manipuladores com mais de três graus de liberdade não possui uma solução única. O mapeamento dos valores das juntas de um manipulador para este atingir determinada posição no espaço, restringir-se-á a apenas uma solução, enquanto muitas outras também seriam viáveis. Num sistema aberto, não se tem a garantia de que o braço possa utilizar os valores mapeados das juntas sem interferir com algum objeto presente em sua trajetória. Uma nova fase de treinamento, uma vez que este é um processo extremamente custoso, impediria o sistema de trabalhar em tempo real, uma vez que, decorrida a fase de treinamento, o ambiente possivelmente já poderia ter-se modificado novamente. Um sistema de controle de movimentos para ambientes abertos deve obrigatoriamente operar e aprender em tempo real.

Grande parte dos sistemas de controle neural de movimentos para braços robóticos incorrem no equívoco de assumirem que todos os obstáculos são conhecidos *a priori*, o que é uma consideração normalmente irreal para ambientes abertos. Barto [BAR 90] apresenta um apanhado geral sobre o uso de redes neurais em controle. O enfoque principal relaciona-se com o modelamento de plantas de controle que funcionam em estados sucessivos basicamente semelhantes a maior parte do tempo. Métodos de otimização de aprendizado também são citados, mas nenhuma referência é feita a sistemas que operam em ambientes abertos. Ritter *et al.* utilizam redes neurais para posicionamento e controle de um braço de robô [RIT 92]. Novamente, apesar dos resultados muito precisos, o robô é suposto operar sozinho em seu ambiente.

Kawato [KAW 90] propõe um modelo computacional de redes neurais em cascata para formação de trajetória e controle de um manipulador PUMA. O modelo proposto é um modelo de rede neural repetitivamente estruturada, invariante no tempo, em cascata, orientada à formação de trajetória baseada no critério do torque mínimo. Esta estrutura é capaz de aprender os modelos dinâmico e cinemático do objeto a ser controlado. Atkeson e Reinkensmeyer [ATK 90] utilizam um modelamento do sistema de controle através de uma memória associativa implementada em uma rede neural. A rede neural produz um mapeamento da dinâmica inversa do robô e procura melhorar o desempenho do seguimento de uma trajetória através de experiências similares previamente armazenadas. Esta memória foi usada para controlar um braço de duas juntas movimentando-se num plano. Memórias associativas, apesar das vantagens que apresentam no modelamento de sistemas dinâmicos, tendem à saturação depois de algum tempo de processamento. Além deste fato, as

limitações produzidas pelo mapeamento da dinâmica inversa do robô são conhecidas, bem como o fato de um braço planar ser mais simples de controlar e exigir um espaço de memória comparativamente menor.

2.4 Controle Neural Adaptativo

Um sistema de controle neural adaptativo pretende somar as vantagens da adaptação contínua ao ambiente apresentada pelos sistemas adaptativos à capacidade de tratamento de não-linearidades e versatilidade das redes neurais. Para tanto, é necessário que a rede neural utilizada tenha capacidade de aprender novas informações sem “esquecer” informações importantes previamente armazenadas. Este critério de relevância de informação torna o processo de desenvolvimento de um sistema de controle com tais características relativamente complicado. Observe-se que não existe ainda uma literatura volumosa sobre controle neural adaptativo, pois esta é efetivamente uma linha de pesquisa recente.

Mel [MEL 90] apresenta MURPHY, um planejador de movimentos de um braço robótico baseado em visão. MURPHY apresenta duas redes neurais, uma para o mapeamento do espaço e outra para o controle de movimentos propriamente dito. No controle de movimentos, é construído um modelo cinemático através de treinamento da rede por exemplos. Este modelo leva em consideração também a visualização da câmera sobre o ambiente. Deste modo, colisões são evitadas através de uma busca heurística sobre o espaço de configurações do braço e o modelamento do ambiente. O braço somente move-se quando uma solução é encontrada. O objetivo principal deste trabalho é a integração sensorial-motora, embora o ambiente utilizado seja consideravelmente artificial. As limitações deste sistema, enquanto controlador de movimentos, são derivadas da busca heurística sobre o espaço de configurações do braço e do ambiente, que precisa ter seu modelo reconstruído a cada movimento do braço. O modelo proposto por Kawato [KAW 90] é bastante interessante e possui algumas extensões que possibilitam o tratamento de obstáculos pela rede. Contudo, o treinamento utilizado não permite, por exemplo, que uma modificação da carga do braço seja levada em consideração pelo sistema, não sendo este sistema, portanto, totalmente adaptativo. Lewis [LEW 95] apresenta um controlador neural capaz de movimentar um manipulador ao longo de um caminho previamente definido. Este controle é realizado em tempo real, mas a rede neural utilizada possui duas camadas e mantém linearidade entre seus parâmetros. Por essa razão, utilizam-se numerosos parâmetros, sempre dependentes do processo a ser controlado. Esta característica torna a tarefa de ajuste do controlador relativamente complexa.

Nguyen e Widrow [NGU 90] propõem um controlador neural capaz de incorporar o estado do processo ao controlador. Engel discute e analisa esta proposta [ENG 96], propondo uma rede neural que funciona como um sistema de controle de malha fechada, efetivamente incorporando à rede o estado do sistema a ser controlado de uma maneira direta. Neste sistema, a rede sofre uma contínua realimentação do estado do processo e responde através da variação dos valores das variáveis de controle do processo a ser controlado, conduzindo ao estado final desejado. Os pesos da rede são ajustados de acordo com o estado global do sistema. O ajuste dos pesos de uma rede neural, quando esta é usada como parte de um sistema dinâmico, pode ser considerado como a parte adaptativa do processo de controle.

2.5 Análise do Estado da Arte

Em processos de controle, quando o comportamento do sistema a ser controlado é de predição relativamente simples, as redes neurais podem ser utilizadas com grande sucesso. A maioria dos trabalhos que utilizam redes neurais, adotam, de fato, a restrição de que o ambiente é sempre conhecido *a priori*. Em sistemas destinados ao controle de um braço, tema central desta pesquisa, este consegue alcançar seu objetivo de uma maneira bastante precisa, após a rede passar por um processo de treinamento apropriado. Contudo, uma modificação no estado do sistema implica obrigatoriamente que a rede passe por um novo processo de treinamento. Esta restrição implica que o braço deve operar sozinho, num ambiente que é completamente conhecido.

Sistemas abertos, são, por definição, sistemas cujos sucessivos estados não são completamente definidos. Em tais sistemas, qualquer estratégia de controle que utilize a restrição de conhecimento *a priori*, ou que não seja capaz de modificar-se para adequar-se às novas necessidades, está fadada ao insucesso. Um sistema adaptativo funciona, então, analisando o estado do sistema e modificando suas leis ou parâmetros de controle, de modo a continuar operando de uma forma eficiente.

Nos trabalhos analisados neste capítulo, no que diz respeito ao controle de um braço manipulador, pode-se verificar que todos possuem deficiências no tratamento de sistemas abertos. Estas deficiências variam desde a falta de flexibilidade no tratamento de sistemas mutáveis, exigindo novo processo de treinamento da rede, até a complexidade dos sistemas de controle propostos, em que a definição e o ajuste dos parâmetros é bastante complexa. Um sistema de controle para um braço de robô que opera num ambiente aberto deve, além de posicionar corretamente o braço, levar em consideração o estado global do sistema, de modo a proceder corretamente a este posicionamento.

Com base neste pressuposto, neste trabalho será apresentado um sistema de controle neural adaptativo para um braço manipulador. Este sistema deverá ser genérico o suficiente para suportar a definição de *qualquer geometria de braço* e não oferecer limitações de soluções para um correto posicionamento espacial do mesmo. O sistema desenvolvido será baseado no trabalho desenvolvido por Engel [ENG 96], tendo sido escolhida esta abordagem por parecer ser a mais promissora, uma vez que a rede neural proposta neste trabalho não apresenta a necessidade de restrições de solução que a maioria das abordagens vistas até então apresentaram.

Neste trabalho será desenvolvido apenas o controle cinemático do robô, sem levar em consideração o seu modelo dinâmico. Esta abordagem poderia ser considerada limitada à primeira vista, mas considera-se que, uma vez que o braço robótico é implementado neste trabalho através de simulação, esta abordagem é, até este ponto, suficiente.

Capítulo 3

A Rede “Attentional Mode”

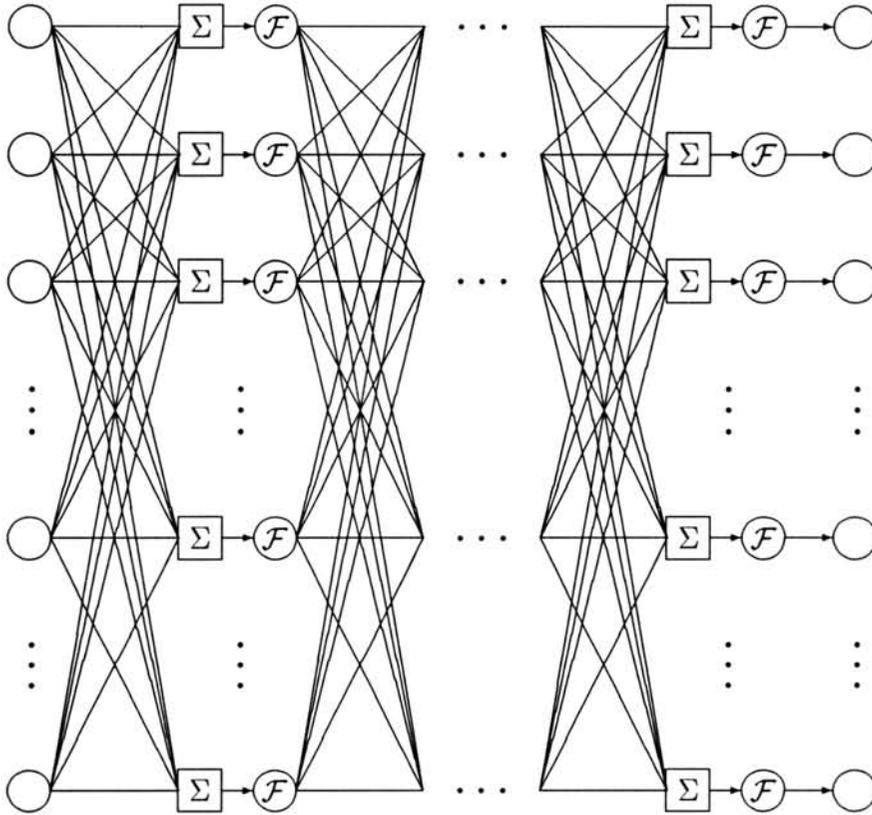
Apesar das vantagens do tratamento de não-linearidades apresentado por redes neurais, muitas vezes estas são usadas simplesmente como versáteis mapeadores não-lineares. Depois de treinada, a rede serve apenas como recuperadora de informações previamente adquiridas. Mesmo com todas as capacidades de generalização, extrapolação e tolerância a ruídos na informação, muitas vezes esta abordagem não é suficiente. Em problemas de controle em que o ambiente a ser controlado muda rapidamente de estado, qualquer conhecimento que possa eventualmente ter sido armazenado tornar-se-á obsoleto em pouco tempo. Como o processo de treinamento de uma rede neural é um processo custoso e demorado, torna-se impossível o controle em tempo real de um sistema qualquer que sofra modificações no tempo.

Uma rede treinada para posicionar um braço mecânico no espaço possui restrições de configuração de posicionamento que não se coadunam com sistemas que funcionam em ambientes abertos. Esta deficiência deriva de que, se existem mais graus de liberdade do que coordenadas, não existe uma relação biunívoca entre os valores dos ângulos das juntas e as coordenadas do órgão terminal do braço, isto é, o problema cinemático inverso não possui uma única solução.

A rede neural *Attentional Mode* [ENG 96] foi desenvolvida tendo em atenção evitar a ocorrência deste problema. A principal característica desta rede é a eliminação da fase de treinamento, fazendo que a mesma adapte-se continuamente e em tempo real ao seu ambiente, conseguindo solucionar um problema de posicionamento sem necessitar que este seja prévia e fixamente determinado quanto à configuração espacial dos diversos agentes presentes no meio.

A rede Attentional Mode é constituída por *elementos processadores*, ou *neurônios*, estruturados em camadas, em que os elementos da camada n são conectados apenas com os elementos da camada $n + 1$, sem recursividade (rede neural *feed-forward*). A figura 3.1 apresenta a estrutura desta rede, onde Σ representa o somatório dos elementos da camada anterior multiplicados pelos pesos das respectivas conexões e \mathcal{F} representa uma função de ativação não-linear.

O algoritmo de aprendizado consiste no método da descida do gradiente, também conhecido na literatura como *steepest descent method* [FRE 91]. A diferença fundamental entre esta abordagem e o algoritmo de *backpropagation* [FRE 91] é a introdução do estado do processo na rede. O erro calculado neste algoritmo não se limita à diferença entre o valor desejado e o valor obtido em cada neurônio da camada de saída da rede, mas efetua o cálculo da contribuição daquele valor ao estado total do sistema.

FIGURA 3.1 — A rede neural *Attentional Mode*

Seja um sistema que possui um estado S . O que se procura são os valores dos pesos que minimizem o erro resultante da diferença entre o estado atual e o estado desejado do sistema. Para tanto, em cada iteração, são fornecidos à rede, como entrada, os valores atuais das variáveis do sistema significativas para o controle e, à saída da rede, temos os valores de modificação das variáveis de controle. Este processo corresponde a um sistema de controle de malha fechada, como é mostrado na figura 3.2.

Se o_k é um neurônio qualquer da última camada da rede e w_{kj} é um peso genérico entre o neurônio k da última camada e o neurônio j da penúltima camada, temos que o gradiente do erro do sistema em relação ao peso w_{kj}^o é dado por

$$\frac{\partial S}{\partial w_{kj}^o} = \overbrace{\frac{\partial S}{\partial o_k}}^a \underbrace{\frac{\partial o_k}{\partial w_{kj}^o}}_b$$

Uma vez que o estado do sistema S depende diretamente dos valores de o_k ¹, o primeiro termo da derivada, indicado por “a”, sempre existe. O segundo termo, indicado por “b”, é conhecido do algoritmo *backpropagation* e possui um desenvolvimento que será apresentado na próxima seção.

¹Os valores de saída da rede o_k são variáveis de controle, e, portanto, influem diretamente no estado do processo.

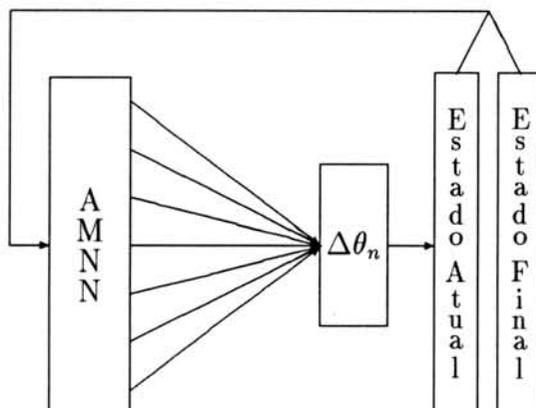


FIGURA 3.2 — Controle neural de malha fechada

Observe-se que o termo $\frac{\partial S}{\partial o_k}$ é dependente do problema a ser tratado. As próximas seções apresentam uma descrição das equações da rede para o problema do posicionamento de um braço mecânico com múltiplas juntas no espaço, resolvendo simultaneamente os problemas cinemático inverso e de geração de trajetória entre dois pontos.

3.1 Estado do Sistema

O estado do sistema, para o caso de posicionamento de um braço no espaço, corresponde ao estado do próprio braço, ou seja, sua posição e orientação espacial relativos a um sistema de referência. Estes valores são conhecidos conhecendo-se os valores das juntas do braço a cada instante (problema cinemático direto). Considera-se que estes valores são conhecidos com uma margem de erro negligenciável. Esta consideração é válida no sentido de que a grande maioria dos robôs possui controladores capazes de fornecer tais valores.

O problema cinemático direto é resolvido através do uso de matrizes de transformação de referenciais, formalmente descritas no Apêndice A. Cada junta do braço possui uma matriz que transforma o referencial associado a esta junta no referencial associado à junta anterior. Seja M_i a matriz de transformação associada à junta i ($i = 1, \dots, N$), onde N é o número de graus de liberdade do braço. Então, a matriz que tranforma o referencial da garra no ponto correspondente ao sistema de referência pode ser definida como

$$M_{ref} = M_1 M_2 \dots M_{n-1} M_n$$

onde M_{ref} é uma matriz de transformação homogênea.

3.1.1 Restrições

Além da posição e orientação do braço no espaço, o estado do sistema pode ainda ser estabelecido segundo restrições de diversos tipos. Estas restrições servem à inserção de uma medida de custo ou desempenho na rede e podem servir a diferentes propósitos, como dar preferência a certas configurações sobre outras, limitar a velocidade dos movimentos e forçar o braço a se mover a partir de um determinado

modo. As restrições usadas neste trabalho são apresentadas a seguir:

a) **Variação dos ângulos** — Nos movimentos realizados pelo braço humano, a variação dos ângulos das juntas (ombro, cotovelo e punho) é contínua. Para reproduzir este movimento, uma vez que os ângulos das juntas variam de modo discreto, a cada iteração da rede, a seguinte restrição é usada:

$$\sum_{k=1}^N |\Delta\theta_k| = 0$$

onde N é o número de juntas do braço do robô e $\Delta\theta_k$ é a variação do valor da junta k . Esta restrição requer que a cada intervalo de tempo Δt , ou a cada iteração da rede, a variação de cada junta seja a menor possível, simulando assim um movimento contínuo do braço.

b) **Deslocamento espacial da garra** — Do mesmo modo como os ângulos das juntas do braço humano variam contínua e lentamente, a extremidade do nosso braço (mão) também se move de uma maneira suave, sem movimentos bruscos em situações normais. Para formalizar esta restrição, temos que:

$$(x_{ant} - x_{atual})^2 + (y_{ant} - y_{atual})^2 + (z_{ant} - z_{atual})^2 = 0$$

ou seja, o deslocamento da garra no espaço deve ser o menor possível a cada iteração da rede.

As restrições de variação dos ângulos das juntas e de deslocamento espacial da garra tornam o tempo de convergência da rede maior, porém permitem que o controle seja feito de um modo mais seguro. Variações bruscas de posição podem gerar colisões ou oscilações indesejáveis.

c) **Relaxamento do braço** — Esta restrição de relaxamento diz respeito à preferência humana por configurações de braço que usem ângulos médios, ou seja, evita-se configurações de braço em que este esteja completamente estendido ou muito contraído sempre que possível. Esta restrição permite que o braço mecânico pareça mais “natural” ou “relaxado” sob um ponto de vista humano. Esta tendência de relaxamento pode ser modelada como:

$$\sum_{k=1}^N |\theta_k - \theta_k^{(m)}| = 0$$

onde θ_k é o valor atual da junta k do braço e $\theta_k^{(m)}$ é o seu valor médio.

Além do aspecto estético, esta restrição tem a importante função de impedir que as juntas excedam seus valores máximos e de evitar que o braço colida consigo mesmo.

3.2 Algoritmo de Adaptação

O algoritmo de adaptação, que modifica os pesos da rede de acordo com a diferença entre o estado atual e o estado desejado do sistema, como já foi referido, é o algoritmo *steepest descent*, onde procede-se à modificação dos pesos de modo a minimizar o erro do sistema em relação a estes mesmos pesos. Sendo K o número total de pesos na rede, o espaço de pesos é um espaço de K dimensões. Para cada

ponto deste espaço, existe um valor de erro correspondente ao estado desejado do processo. O objetivo do algoritmo de adaptação é percorrer esta superfície de K dimensões de modo a encontrar o seu mínimo, tendo-se assim encontrado o valor dos pesos que minimizam o erro total. O gradiente do erro fornece o valor para onde a função em questão é maximizada. Para obtermos o mínimo, portanto, devemos seguir a direção negativa do gradiente.

3.2.1 Erros Relativos ao Estado do Sistema

O erro relativo ao estado do sistema S é calculado como sendo a diferença entre o estado desejado e o estado atual de S . Este erro pode ser definido como

$$E_S = (S_{final} - S_{atual})$$

O *erro quadrado total* gerado pela rede corresponde à soma dos erros quadrados que contribuem para o estado do sistema. Estes erros são relativos à posição e à orientação espacial da garra em relação à posição e à orientação desejadas somados aos erros referentes às restrições utilizadas para garantir um movimento adequado do braço. O erro quadrado total é então dado por:

$$E_S^2 = \frac{1}{2} (\varepsilon_{pos}^2 + \varepsilon_{ori}^2 + \varepsilon_{restr}^2)$$

O erro relativo à posição do braço diz respeito à diferença de valor entre as coordenadas espaciais x , y e z onde a garra está localizada e a posição espacial desejada. Sendo assim,

$$\varepsilon_{pos}^2 = \varepsilon_x^2 + \varepsilon_y^2 + \varepsilon_z^2$$

onde $\varepsilon_x^2 = (x_f - x)^2$, $\varepsilon_y^2 = (y_f - y)^2$ e $\varepsilon_z^2 = (z_f - z)^2$ correspondem ao erro quadrado medido a respeito de cada coordenada espacial do braço com relação ao ponto final desejado.

O erro em relação à orientação do braço mecânico no espaço é a diferença entre a orientação real e a orientação desejada. Matematicamente, tem-se que:

$$\varepsilon_{ori}^2 = \varepsilon_{u_x}^2 + \varepsilon_{u_y}^2 + \varepsilon_{u_z}^2 + \varepsilon_{v_x}^2 + \varepsilon_{v_y}^2 + \varepsilon_{v_z}^2 + \varepsilon_{w_x}^2 + \varepsilon_{w_y}^2 + \varepsilon_{w_z}^2$$

onde os vetores \vec{u} , \vec{v} e \vec{w} são os vetores que designam a orientação do braço no espaço de coordenadas $OUVW$ associado à garra em relação aos vetores diretores do espaço de referência $OXYZ$. Estes erros são facilmente calculados subtraindo-se os valores da submatriz 3×3 superior esquerda da matriz que representa o estado atual do robô dos valores da mesma submatriz da matriz objetivo. Da mesma forma, os erros correspondentes aos valores de posição do braço podem ser imediatamente encontrados através do mesmo cálculo nas submatrizes 3×1 superiores direitas destas matrizes.

Os erros gerados pelas restrições usadas apresentadas na seção 3.1.1 são definidos a seguir.

a) **Variação dos ângulos** — Uma vez que pretende-se minimizar o valor da variação dos ângulos das juntas, o erro referente a esta restrição é o próprio valor da

variação $\Delta\theta_k$ para cada junta k . O erro quadrado correspondente à esta restrição é então definido como:

$$\epsilon_{\Delta\theta_k}^2 = \Delta\theta_k^2 \quad (3.1)$$

b) **Deslocamento espacial da garra** — Esta restrição tem como objetivo minimizar o valor do deslocamento espacial a cada intervalo de tempo; logo, o valor do erro é o próprio valor do deslocamento. O erro quadrado referente à esta restrição é então dado por:

$$\epsilon_{d_{garra}}^2 = (x_{ant} - x_{atual})^2 + (y_{ant} - y_{atual})^2 + (z_{ant} - z_{atual})^2 \quad (3.2)$$

c) **Relaxamento do braço** — O erro quadrado correspondente a cada um dos ângulos θ_k é o valor ao quadrado da diferença entre o ângulo atual de cada junta e o valor desejado para aquela junta, ou seja, seu valor médio:

$$\epsilon_{relax}^2 = (\theta_k - \theta_k^{(m)})^2 \quad (3.3)$$

3.2.2 Cálculo do Gradiente do Erro

O gradiente do erro quadrado do sistema tratado com relação às suas variáveis de controle é dado pelo somatório das derivadas parciais de cada parcela do erro total do sistema em relação a estas variáveis, ou seja,

$$\frac{\partial E_S^2}{\partial \Delta\theta_k} = \frac{1}{2} \left(\frac{\partial \epsilon_{pos}^2}{\partial \Delta\theta_k} + \frac{\partial \epsilon_{ori}^2}{\partial \Delta\theta_k} + \frac{\partial \epsilon_{restr}^2}{\partial \Delta\theta_k} \right)$$

A posição e a orientação do braço, bem como as restrições de movimento consideradas em 3.2.1, são fundamentalmente dependentes da variação dos ângulos das juntas do braço, que correspondem aos valores das variáveis de controle.

As derivadas dos erros das variáveis de posição e orientação do braço em relação aos ângulos das juntas podem ser escritas como

$$\frac{\partial \epsilon_x^2}{\partial \Delta\theta_k} = -(\mathbf{v}_f - \mathbf{v}) \frac{\partial \mathbf{v}}{\partial \Delta\theta_k}$$

onde \mathbf{v} é qualquer uma das variáveis de posição x , y , z ou de orientação u_x , u_y , u_z , v_x , v_y , v_z , w_x , w_y , w_z . As derivadas destas variáveis em relação aos ângulos podem ser facilmente encontradas através da multiplicação das matrizes de transformação e de derivação².

As derivadas do erro quadrado das restrições de movimento do braço também são diretamente calculadas como é mostrado a seguir.

a) **Variação dos ângulos** — O gradiente do erro quadrado correspondente a esta restrição, levando-se em conta a equação 3.1, pode ser calculado da seguinte maneira:

²Matrizes de derivação foram desenvolvidas no decorrer deste trabalho para proporcionar um método sistemático de derivação dos termos das equações cinemáticas diretas em relação a cada uma das juntas. Esta técnica encontra-se formalmente descrita no Apêndice B.

$$\frac{\partial \varepsilon^2_{\Delta\theta_k}}{\partial \Delta\theta_k} = \frac{1}{2} \left(\frac{\partial \Delta\theta_k^2}{\partial \Delta\theta_k} \right) = \Delta\theta_k$$

b) **Deslocamento espacial da garra** — Tomando-se a equação 3.2, temos que a derivada parcial em relação aos ângulos das juntas referente à restrição de deslocamento espacial da garra é dado por

$$\frac{\partial \varepsilon^2_{d_{garra}}}{\partial \Delta\theta_k} = \frac{1}{2} \left(\frac{\partial ((x_{atual} - x_{ant})^2 + (y_{atual} - y_{ant})^2 + (z_{atual} - z_{ant})^2)}{\partial \Delta\theta_k} \right)$$

ou, ainda,

$$\frac{\partial \varepsilon^2_{d_{garra}}}{\partial \Delta\theta_k} = (x_{atual} - x_{ant}) \frac{\partial x_{atual}}{\partial \Delta\theta_k} + (y_{atual} - y_{ant}) \frac{\partial y_{atual}}{\partial \Delta\theta_k} + (z_{atual} - z_{ant}) \frac{\partial z_{atual}}{\partial \Delta\theta_k}$$

c) **Relaxamento do braço** — E, finalmente, o gradiente do erro quadrado referente à restrição de relaxamento do braço para cada uma das juntas k , levando-se em consideração a equação 3.3, pode ser escrito como

$$\frac{\partial \varepsilon^2_{relax}}{\partial \Delta\theta_k} = \frac{1}{2} \left(\frac{\partial (\theta_k - \theta_k^{(m)})^2}{\partial \Delta\theta_k} \right)$$

ou, ainda,

$$\frac{\partial \varepsilon^2_{relax}}{\partial \Delta\theta_k} = (\theta_k - \theta_k^{(m)}) \frac{\partial \theta_k}{\partial \Delta\theta_k}$$

mas $\theta_k = \theta_{ant_k} + \Delta\theta_k$, sendo que o valor anterior de θ_k , não é dependente de $\Delta\theta_k$. Então, pode-se dizer que

$$\frac{\partial \varepsilon^2_{relax}}{\partial \Delta\theta_k} = (\theta_k - \theta_k^{(m)}) \frac{\partial \theta_{ant_k} + \Delta\theta_k}{\partial \Delta\theta_k} = (\theta_k - \theta_k^{(m)})$$

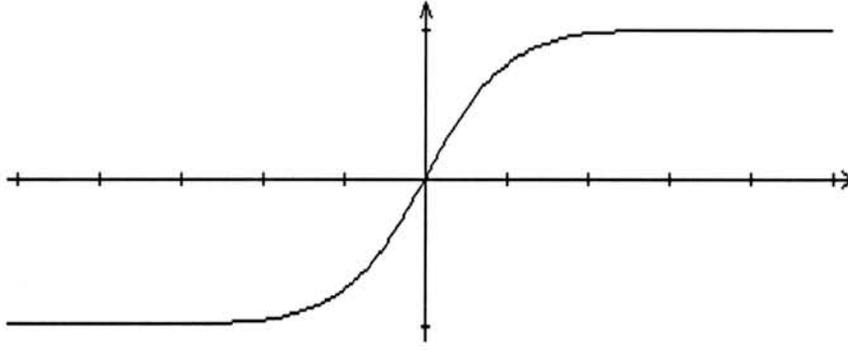
ou seja, é o próprio valor da diferença entre os ângulos, como já era esperado.

3.2.3 Função de Ativação

A função de ativação \mathcal{F} utilizada é a função *tangente hiperbólica*. O gráfico desta função encontra-se na figura 3.3. Como pode-se ver, o contra-domínio desta função é o intervalo $[-1, +1]$. Este fato torna esta função mais adequada a este problema, uma vez que este requer que os resultados variem num intervalo que permita a mudança, positiva ou negativa, dos valores das juntas do braço do robô.

A derivada da função tangente hiperbólica é dada pela equação a seguir:

$$\frac{d[\tanh(x)]}{dx} = 1 - \tanh(x)^2 \quad (3.4)$$

FIGURA 3.3 — A Função *tangente hiperbólica*

3.2.4 Atualização dos Pesos na Camada de Saída

A atualização dos pesos na camada de saída é feita através do cálculo do gradiente do estado do sistema em relação aos pesos desta camada, ou seja,

$$\frac{\partial E^2}{\partial w_{kj}^o} = \frac{\partial \varepsilon_{pos}^2}{\partial \Delta\theta_k} \frac{\partial \Delta\theta_k}{\partial w_{kj}^o} + \frac{\partial \varepsilon_{ori}^2}{\partial \Delta\theta_k} \frac{\partial \Delta\theta_k}{\partial w_{kj}^o} + \frac{\partial \varepsilon_{restr}^2}{\partial \Delta\theta_k} \frac{\partial \Delta\theta_k}{\partial w_{kj}^o}$$

As derivadas dos ângulos em relação aos pesos da camada de saída $\Delta\theta_k$, são dadas por

$$\frac{\partial \Delta\theta_k}{\partial w_{kj}^o} = \frac{\partial \mathcal{F}_k^o}{\partial net_k^o} \frac{\partial net_k^o}{\partial w_{kj}^o} \quad (3.5)$$

onde $net_k^o = \sum_{j=1}^L w_{kj}^o i_j$, com L sendo o número de neurônios da penúltima camada da rede e i_j sendo os valores dos neurônios desta camada.

A derivada de net_k^o em relação ao peso w_{kj}^o pode ser calculada do seguinte modo:

$$\frac{\partial net_k^o}{\partial w_{kj}^o} = \frac{\partial}{\partial w_{kj}^o} \left(\sum_{j=1}^L w_{kj}^o i_j \right) = i_j \quad (3.6)$$

Combinando as equações 3.4, 3.5 e 3.6, a variação dos pesos da camada de saída pode ser então calculado como segue:

$$\Delta w_{kj}^o = \left(\alpha \frac{\partial \varepsilon_{pos}^2}{\partial \Delta\theta_k} + \beta \frac{\partial \varepsilon_{ori}^2}{\partial \Delta\theta_k} + \sum_r \gamma_r \frac{\partial \varepsilon_{restr_r}^2}{\partial \Delta\theta_k} \right) (1 - \Delta\theta_k^2) i_j \quad (3.7)$$

onde r é o número de restrições utilizadas na implementação da rede e α , β e γ_r são os parâmetros que medem a importância relativa de cada erro no comportamento global do sistema. Os valores destes parâmetros e seus significados serão discutidos no próximo capítulo.

Definindo a quantidade δ_k^o como sendo

$$\delta_k^o = \left(\alpha \frac{\partial \varepsilon_{pos}^2}{\partial \Delta\theta_k} + \beta \frac{\partial \varepsilon_{ori}^2}{\partial \Delta\theta_k} + \sum_r \gamma_r \frac{\partial \varepsilon_{restr_r}^2}{\partial \Delta\theta_k} \right) (1 - \Delta\theta_k^2)$$

temos que

$$\Delta w_{kj}^o = \delta_k^o i_j \quad (3.8)$$

e a modificação de um peso genérico da camada de saída pode ser então escrita como

$$w_{kj}^o = w_{kj}^o - \mu \delta_k^o i_j \quad (3.9)$$

onde μ é a taxa de aprendizado da rede.

3.2.5 Atualização dos Pesos nas Camadas Escondidas

O erro quadrado total varia em relação aos pesos da camada escondida da seguinte forma:

$$\frac{\partial E^2}{\partial w_{ji}^h} = \sum_k \left(\frac{\partial \varepsilon_{pos}^2}{\partial \Delta \theta_k} + \frac{\partial \varepsilon_{ori}^2}{\partial \Delta \theta_k} + \sum_r \gamma_r \frac{\partial \varepsilon_{restrr}^2}{\partial \Delta \theta_k} \right) \frac{\partial \Delta \theta_k}{\partial net_k^o} \frac{\partial net_k^o}{\partial i_j} \frac{\partial i_j}{\partial net_j^h} \frac{\partial net_j^h}{\partial w_{ji}^h} \quad (3.10)$$

Cada um dos fatores da equação 3.10 podem ser calculados explicitamente das equações anteriores. O resultado obtido é

$$\frac{\partial E^2}{\partial w_{ji}^h} = \sum_k \left(\frac{\partial \varepsilon_{pos}^2}{\partial \Delta \theta_k} + \frac{\partial \varepsilon_{ori}^2}{\partial \Delta \theta_k} + \sum_r \gamma_r \frac{\partial \varepsilon_{restrr}^2}{\partial \Delta \theta_k} \right) (1 - \Delta \theta_k^2) w_{kj}^o (1 - i_j^2) i_i$$

ou, ainda,

$$\frac{\partial E^2}{\partial w_{ji}^h} = \sum_k \delta_k^o w_{kj}^o (1 - i_j^2) i_i$$

ou, uma vez que o somatório dá-se na variável k ,

$$\frac{\partial E^2}{\partial w_{ji}^h} = (1 - i_j^2) i_i \sum_k \delta_k^o w_{kj}^o$$

onde j é o índice do elemento da camada intermediária para qual os pesos serão atualizados, i é o índice dos elementos da camada anterior e k o índice dos elementos da camada posterior.

Definindo a quantidade δ_j^h como

$$\delta_j^h = (1 - i_j^2) \sum_k \delta_k^o w_{kj}^o$$

temos que a variação dos pesos das camadas intermediárias é dada por

$$\Delta w_{ji}^h = \delta_j^h i_i \quad (3.11)$$

e os novos pesos das camadas intermediárias são

$$w_{ji}^h = w_{ji}^h - \mu \delta_j^h i_i$$

Note-se que esta equação é perfeitamente análoga à equação 3.9, podendo-se, então, desta forma, calcular-se recursivamente as variações dos pesos para todas as

camadas intermediárias.

Capítulo 4

Implementação do Modelo

A implementação do modelo proposto pode ser esquematizado conforme a figura 4.1. Os módulos do *controle*, formado pela rede neural *Attentional Mode* implementada, e do *sistema*, formado pela informação referente ao braço robótico simulado, compõem um sistema de controle de malha fechada, onde existe uma realimentação contínua do estado do sistema para o controlador, que leva este estado em consideração no próximo ciclo de controle. O módulo de *visualização* possui o propósito de permitir que seja vista a movimentação do braço até o seu objetivo, permitindo uma melhor avaliação da evolução do sistema no tempo. Os módulos do controle, do sistema, da visualização e o funcionamento do ciclo de controle são descritos a seguir.

4.1 Sistema

O módulo do sistema é constituído pelo braço robótico implementado. O braço aqui apresentado é um entre os muitos diferentes tipos de manipuladores testados em simulação. Este manipulador foi escolhido para ser apresentado nesta dissertação não por ter apresentado os melhores resultados, mas por ser o braço mais complexo implementado. Nos manipuladores mais simples (de tamanho menor ou dotados de menor número de graus de liberdade), não se apresentam muitas das dificuldades que surgem ao se lidar com um braço de maior porte, com maior número de articulações, por isso mais difícil de controlar. Acredita-se que o exemplo a ser apresentado será bem mais ilustrativo dos problemas de controle que se pretende atacar (e da sua solução). O braço implementado possui modelos *geométrico* e *cinemático* associados, os quais serão descritos a seguir.

4.1.1 Modelo Geométrico

O modelo geométrico de um braço manipulador diz respeito às suas características físicas, ou seja, às suas dimensões espaciais. A sua importância reside em que o conjunto das dimensões dos elementos do braço e os tipos e as capacidades de deslocamento (*angular* e *linear*) das juntas determinarão o *espaço de trabalho* do robô.

O braço implementado experimentalmente possui 5 (cinco) graus de liberdade e suas características geométricas encontram-se descritas na tabela 4.1.

Na descrição do modelo geométrico do braço, *turn* e *down* são juntas rotacionais. A primeira gira o segmento em torno do eixo que liga as extremidades deste

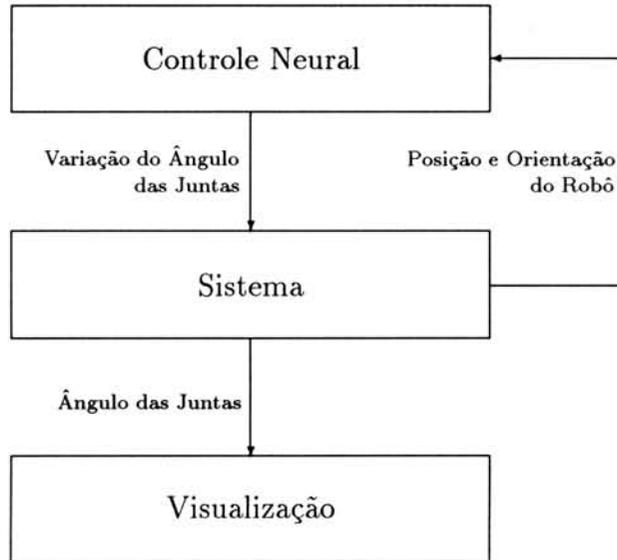


FIGURA 4.1 — Módulos de implementação do sistema

TABELA 4.1 — Modelo geométrico do braço implementado

Segmento	Comprimento	Tipo de Junta
1	150mm	<i>turn</i>
2	200mm	<i>down</i>
3	180mm	<i>down</i>
4	150mm	<i>down</i>
5	50mm	<i>turn</i>

segmento e a segunda gira ao redor do eixo normal ao eixo que liga as extremidades do segmento em questão¹. A figura 4.2 mostra um esquema do braço implementado.

4.1.2 Modelo Cinemático

A cinemática de um braço robótico diz respeito à descrição analítica do deslocamento espacial do braço como função do tempo. O modelo cinemático de um braço manipulador pode ser definido através de *matrizes homogêneas de transformação de referenciais*². As matrizes de transformação definidas para este braço são as seguintes:

$$M_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

¹Em relação às variáveis espaciais x , y e z , uma junta *turn* gira ao redor do eixo z e a junta *down* ao redor de x ou y , sem perda de generalidade.

²Ver Apêndices A e Apêndice C.

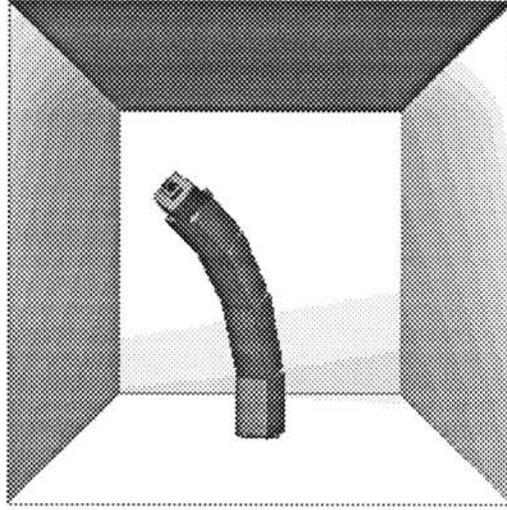


FIGURA 4.2 — Braço implementado

$$M_2 = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 & d_2 \sin \theta_2 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_2 & 0 & \cos \theta_2 & d_2 \cos \theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} \cos \theta_3 & 0 & \sin \theta_3 & d_3 \sin \theta_3 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_3 & 0 & \cos \theta_3 & d_3 \cos \theta_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_4 = \begin{bmatrix} \cos \theta_4 & 0 & \sin \theta_4 & d_4 \sin \theta_4 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_4 & 0 & \cos \theta_4 & d_4 \cos \theta_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_5 = \begin{bmatrix} \cos \theta_5 & -\sin \theta_5 & 0 & 0 \\ \sin \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

onde M_i ($i = 1, \dots, 5$) é a matriz que transforma um ponto do referencial i no referencial $i - 1$, sendo o referencial 0, o sistema de coordenadas de referência $OXYZ$; e o referencial 5, o referencial associado à garra; θ_i e d_i ($i = 1, \dots, 5$) são respectivamente o valor da junta i e o comprimento do segmento i do braço manipulador.

As matrizes de derivação³ D_i correspondem às derivadas dos elementos das matrizes de transformação e são dadas por

³Ver Apêndice B.

$$D_1 = \begin{bmatrix} -\sin \theta_1 & -\cos \theta_1 & 0 & 0 \\ \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} -\sin \theta_2 & 0 & \cos \theta_2 & d_2 \cos \theta_2 \\ 0 & 0 & 0 & 0 \\ -\cos \theta_2 & 0 & -\sin \theta_2 & -d_2 \sin \theta_2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D_3 = \begin{bmatrix} -\sin \theta_3 & 0 & \cos \theta_3 & d_3 \cos \theta_3 \\ 0 & 0 & 0 & 0 \\ -\cos \theta_3 & 0 & -\sin \theta_3 & -d_3 \sin \theta_3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D_4 = \begin{bmatrix} -\sin \theta_4 & 0 & \cos \theta_4 & d_4 \cos \theta_4 \\ 0 & 0 & 0 & 0 \\ -\cos \theta_4 & 0 & -\sin \theta_4 & -d_4 \sin \theta_4 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D_5 = \begin{bmatrix} -\sin \theta_5 & -\cos \theta_5 & 0 & 0 \\ \cos \theta_5 & -\sin \theta_5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Com estas matrizes, dado um vetor de ângulos $\vec{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)^T$ que representam os valores das respectivas juntas do braço, pode-se calcular facilmente a posição e a orientação do braço no espaço, bem como as derivadas de cada variável de posição e de orientação em relação a qualquer ângulo de junta.

Em relação à definição do braço, as matrizes de transformação e de derivação do manipulador são as únicas informações estruturais necessárias ao funcionamento do controlador. Uma vez que essas matrizes podem ser diretamente obtidas do modelo geométrico do braço, a abordagem utilizada facilita consideravelmente a definição do sistema de controle.

4.2 Controle Neural

O controle do sistema propriamente dito é realizado por uma rede neural *Attentional Mode*⁴. Os valores de entrada da rede são as variáveis de posição e de orientação do braço no espaço, ou seja, os elementos da submatriz superior de dimensões 3×4 da matriz $M_{ref} = M_1 M_2 M_3 M_4 M_5$, que representam estas variáveis. Os valores de saída são os valores de modificação das juntas do braço do robô $\Delta\theta_k$ ($k = 1, \dots, 5$). Esses valores, tanto de os de entrada como os de saída, estão normalizados, ou seja, pertencem ao intervalo $[-1, 1]$. Uma vez que os vetores de

⁴O modelo e funcionamento da rede neural *Attentional Mode* são formalmente descritos no capítulo 3.

orientação são vetores diretores do espaço e, por esta razão, encontram-se normalizados, é necessário proceder apenas à normalização dos valores de posição do braço. Estes valores são normalizados da seguinte maneira:

$$n_v = \frac{v_{var}}{c_{man}}$$

onde n_v é o valor normalizado da variável em questão, v_{var} é o valor não normalizado desta variável e c_{man} é o comprimento do manipulador robótico. Uma vez que o centro do sistema de referência é a posição da base do robô, verifica-se que

$$v_{var} \in [-c_{man}, +c_{man}]$$

ou, ainda, que

$$n_v \in [-1, 1]$$

Os valores de saída também encontram-se normalizados no mesmo intervalo $[-1, 1]$. Para encontrar o valor de modificação da junta, basta multiplicar os valores conseguidos pelo valor máximo de modificação da junta. Este valor máximo de modificação é fornecido como parâmetro do sistema de controle. Ainda como parâmetros do sistema de controle, tem-se a configuração da rede neural (número de camadas e número de neurônios em cada camada), a taxa de aprendizado μ , o intervalo de valores para inicialização dos pesos, o número de graus de liberdade do braço, as dimensões dos segmentos do braço e todos os outros parâmetros definidos para este sistema.

4.2.1 Estrutura da Rede Neural

A rede neural implementada neste trabalho possui quatro camadas de elementos processadores. A camada de entrada é constituída de doze elementos, três representando a localização espacial (variáveis x , y e z) do braço, e as nove restantes, a orientação da garra em relação ao sistema de coordenadas de referência.

Pode ser demonstrado que redes neurais com duas camadas intermediárias são mais adequadas para solução de problemas de controle, como os experimentos realizados confirmaram [SON 94].

4.2.2 Parâmetros dos Erros Gerados pela Rede

No cálculo da variação dos pesos da rede neural que controla o processo, cada parcela de erro é multiplicada por um parâmetro que representa a relevância daquele erro para o erro global do sistema como um todo.

4.2.2.1 Posição e Orientação

Os parâmetros de posição e orientação do braço definidos na equação 3.7 (denominados α e β) são diferentes, para permitirem um melhor controle do robô, devendo variar ao longo do tempo de modo a permitirem a existência tanto de uma convergência rápida, como de um sistema estável. No início de um movimento, se a garra está longe do objetivo a ser atingido, a posição do braço é mais importante que sua orientação. À medida que a garra aproxima-se do objetivo, a orientação

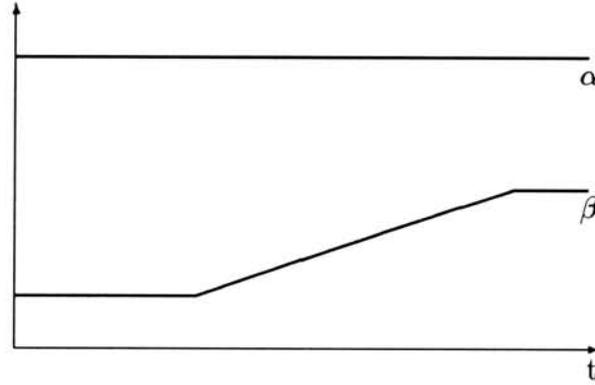


FIGURA 4.3 — Evolução do parâmetro de orientação do sistema de controle

do braço torna-se mais importante. A adaptação destes parâmetros de acordo com o estado do braço são muito importantes para a convergência do mesmo para a situação final desejada.

Se, no início do movimento, o parâmetro de orientação possuir um valor muito alto, o braço, a cada nova iteração do ciclo de controle, será movimentado de maneira a orientar-se de acordo com o objetivo. Este fato torna a convergência do sistema, como um todo, mais demorada, pois, apesar de ser a orientação um fator importante, não é preponderante estando o braço muito longe do seu objetivo. Mas, na medida em que o braço aproxima-se do seu objetivo final, a orientação passa a ser crítica. Então, o parâmetro correspondente deve ter seu valor aumentado, de modo a ter o erro referente à orientação aumentado dentro do sistema, possibilitando, assim, que o braço oriente-se corretamente.

Note-se que as variáveis de orientação (componentes dos vetores de orientação da extremidade do braço em relação aos vetores de orientação do espaço fixo de referência) são em número maior do que as variáveis de posição do braço. Por esta razão, mesmo no final da tarefa, quando o erro da posição atual do braço em relação à posição desejada é relativamente pequeno, o parâmetro de orientação deve sempre ser menor que o de posição. De outra forma, o controlador procurará uma correta orientação do braço sem, no entanto, considerar a adequação de posição tão importante quanto deveria. A figura 4.3 mostra a evolução ao longo do tempo do parâmetro de orientação (β) com relação ao parâmetro de posição (α).

4.2.2.2 Restrições

Os parâmetros das restrições possuem um papel fundamental no ciclo de controle do processo. O papel destas restrições é garantir que o braço seja posicionado corretamente de uma maneira contínua e estável. Contudo, um certo cuidado é necessário na atribuição dos seus valores.

A respeito das restrições de variação dos ângulos do braço e de deslocamento espacial, deve ser observada a existência de um compromisso entre a velocidade e a estabilidade da convergência à posição final. Parâmetros muito altos farão que o braço mova-se de uma maneira extremamente suave, mas talvez mais lenta do que o aceitável. Por outro lado, valores muito baixos podem fazer que o braço chegue rapidamente ao ponto objetivo e que, então, fique oscilando em torno desse, devido

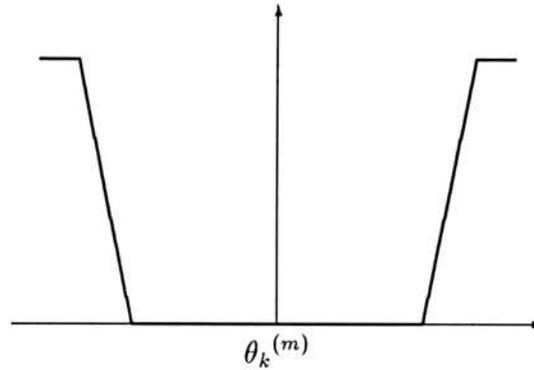


FIGURA 4.4 — Evolução do valor da restrição de relaxamento

à inércia⁵ imposta pelos erros gerados ao longo do movimento. Os valores desses parâmetros também devem depender da quantidade de graus de liberdade do braço em questão. Quanto maior for este número, mais inércia será gerada, e maiores deverão ser os valores desses parâmetros.

A restrição de relaxamento do braço, além de impedir que o robô exceda os limites de variação de suas juntas, permite que o braço tenha sempre configurações semelhantes, em forma, às do braço humano. Os valores deste parâmetro necessitam sempre ser modificados, dependendo da configuração atual do braço e do andamento da tarefa. Com valores fixos, o braço adotará a configuração desejada, mas, ao começar a aproximar-se do objetivo, os valores dos erros de posição e de orientação diminuirão, fazendo que o erro gerado por esta restrição seja dominante. O resultado é que o braço irá afastar-se do seu objetivo para valores médios dos ângulos de suas juntas.

Para contornar esse problema, o parâmetro de relaxamento do braço deve sempre ser muito menor que os de posição e de orientação, exceto quando alguma junta aproximar-se de seu limite. Neste caso, o parâmetro de relaxamento deve tornar-se maior, com o intuito de impedir que o braço atinja seus limites. Logo que esta situação seja superada, contudo, este parâmetro deve voltar a apresentar valores baixos, de modo a não interferir na execução da tarefa. A figura 4.4 apresenta um esquema da evolução do valor da restrição de relaxamento de acordo com os valores das juntas do braço, onde $\theta_k^{(m)}$ é o valor médio da junta em questão.

4.3 Visualização

Ferramentas de simulação são uma necessidade básica na construção de ambientes robóticos para o desenvolvimento de técnicas de controle. Para que tal desenvolvimento possa ser considerado completo, é necessário o teste de diversos tipos de robôs, com várias configurações distintas, e de diferentes métodos de controle aplicados a cada uma destas configurações. Estes testes tendem a tornar-se proibitivos em termos de custos se uma ferramenta de simulação não estiver disponível. Tais ferramentas requerem um alto grau de esforço, com relação ao tempo e ao pessoal

⁵A palavra *inércia* neste contexto refere-se às modificações nos pesos da rede neural gerados pelo acúmulo de erros no decorrer do tempo e não à quantidade física comumente designada por este termo

envolvido, para serem construídas. Independentemente deste fato, quase sempre tais ferramentas são construídas com o único propósito de proceder-se a um determinado teste em específico. Normalmente, generalidade, portabilidade e reutilização de código não são levados em conta na construção destes programas.

Durante o desenvolvimento do projeto *JANUS* [KEL 94, SMI 94a, SMI 94b, SMI 94c, SMI 92, SMI 90], diversas ferramentas para simulação e interface com o usuário foram construídas. Estas ferramentas, apesar de possuírem capacidades de uso geral, estavam profundamente imiscuídas dos objetivos contidos nos programas de definição e controle do robô *JANUS*. Este fato impedia a utilização dessas ferramentas por outra aplicação que não o próprio simulador do robô.

O ambiente de simulação aqui descrito foi construído a partir do código do robô *JANUS*, com o objetivo de fornecer uma interface amigável para a construção de simuladores robóticos para robôs das mais diversas configurações e que possuam as mais variadas formas de controle. Este sistema também deveria formar um pacote de utilização que não fosse dependente da aplicação e que pudesse tornar-se de domínio público juntamente com sua documentação. O ambiente construído é totalmente modular, na forma de bibliotecas, usando o paradigma de *orientação a objetos*, escrito em *C++*, sendo executável sob o *X Window System* em sistemas operacionais *UNIX*⁶. A estrutura e o funcionamento deste sistema, chamado *GSE - Graphic Simulation Environment* são descritos no Apêndice D.

Com este sistema disponível e funcional, tornou-se possível a realização de experimentos para teste de técnicas de controle em ambientes robóticos.

4.4 Ciclo de Controle

O ciclo de controle pode ser esquematizado e sumarizado através do seguinte algoritmo:

```

se (braço atingiu o objetivo)
  então
    parar
  senão
    enviar posição e orientação do braço para a rede
    propagar valores pela rede
    obter valores de saída da rede
    somar valores da junta fornecidos pela rede com os valores atuais
    calcular valores dos erros gerados pela nova configuração
    adaptar valores dos pesos da rede
  fim senão

```

A condição “*braço atingiu o objetivo*” depende do valor do erro tolerado para a tarefa em questão, podendo ser verificada de diversas formas. Por exemplo, se o objetivo é um ponto de um caminho traçado para o braço, mas não é ainda o objetivo final, então este erro pode ser razoavelmente grande, sem prejuízo da solução final. De outra forma, se o ponto correspondente ao objetivo for o objetivo final, então o valor do erro aceitável para dar o movimento por encerrado depende do grau de

⁶Este trabalho foi desenvolvido na sede da *German Research Center for Computer Science* (GMD) ao abrigo do Acordo Brasil-Alemanha para Cooperação em Sistemas Adaptativos.

tolerância específica da tarefa. O cálculo deste erro pode ser feito levando-se em consideração a distância euclidiana da garra ao objetivo traçado, ou a distância de cada coordenada (x, y, z) ao objetivo, ou ainda a orientação do braço.

Neste trabalho, o erro máximo tolerado para se considerar que o braço tenha atingido o seu objetivo pode variar de valor de acordo com os testes realizados. No entanto, o cálculo deste erro é fixo, consistindo na distância euclidiana do centro da garra ao objetivo, sendo dado por

$$erro_d = \sqrt{(x_{obj} - x)^2 + (y_{obj} - y)^2 + (z_{obj} - z)^2}$$

A orientação da garra não é considerada devido a que, nos testes realizados, esta convergia sempre, sendo o erro final quase sempre negligenciável.

Caso o braço não tenha ainda atingido seu objetivo, há uma nova etapa a ser executada no ciclo de controle. Esta etapa, apresentada sob forma de algoritmo acima, é uma descrição algorítmica do esquema apresentado na figura 3.2. Note-se que um ciclo de controle da rede é bastante semelhante a um ciclo de treinamento de uma rede neural que use o método de descida do gradiente. A principal diferença é que, na rede *Attentional Mode*, o erro usado é o erro real do sistema — e não uma estimativa do erro global baseada no erro da amostra usada naquele ciclo de treinamento. Este fato é o principal fator que permite à rede convergir rapidamente para a solução final.

4.5 Complexidade Computacional

Um fator preponderante na análise de qualquer algoritmo é a sua complexidade. Mesmo que um algoritmo seja capaz de encontrar uma solução para um determinado problema, não significa que esta solução possa ser aplicada na prática. A relação entre a complexidade do problema a ser tratado e o tempo e a memória gastos para encontrar uma solução para o problema são os fatores que definem a utilidade de um algoritmo. A análise do sistema de controle neural para um braço robótico é realizada nas próximas seções.

4.5.1 Complexidade do Algoritmo

O estudo da ordem de complexidade de um algoritmo é normalmente realizado em termos da entrada deste. Num sistema de controle de malha fechada, como o apresentado neste trabalho, o tamanho da entrada é fixo, correspondente às variáveis de estado do processo. A saída do algoritmo de controle também é fixa, correspondendo aos valores das variáveis de controle do processo. Desta forma, a complexidade algorítmica a ser apresentada nesta seção diz respeito à complexidade do processo de controle de, a partir da entrada, gerar as saídas esperadas para o sistema.

A complexidade C do algoritmo de controle pode ser definida como

$$C = C_{rede} + C_S + C_{\Delta pesos} \quad (4.1)$$

onde C_{rede} é a complexidade da propagação das entradas pela rede, C_S é a complexidade da transformação dos valores fornecidos pela rede no novo estado do sistema e $C_{\Delta pesos}$ é a complexidade da modificação dos pesos da rede pelo algoritmo de

adaptação.

A complexidade da propagação das entradas pela rede C_{rede} depende exclusivamente da topografia da rede neural. Sendo uma rede estruturada em camadas, com uma camada de entrada com I neurônios, uma camada de saída com O neurônios e L camadas intermediárias com L_1, L_2, \dots, L_L neurônios cada uma, temos que a complexidade da propagação das entradas pela rede com tal configuração pode ser escrita como

$$C_{rede} = (L_1 I) + (L_2 L_1) + (L_3 L_2) + \dots + (L_{L-1} L_{L-2}) + (L_L L_{L-1}) + (O L_L)$$

sendo $A = \max(I, L_1, \dots, L_L, O)$, temos que esta complexidade é da ordem de

$$C_{rede} = (L + 1) A^2 \quad (4.2)$$

A transformação dos valores fornecidos pela rede no novo estado do sistema corresponde à transformação do referencial associado à garra no referencial equivalente do sistema fixo de referência. Esta transformação é realizada através da multiplicação das matrizes de transformação associadas à cada junta do braço robótico. Sendo G o número de graus de liberdade do braço em questão, é realizada a multiplicação deste mesmo número de matrizes de transformação, ou seja,

$$M_{ref} = M_1 M_2 \dots M_{G-1} M_G$$

Uma vez que cada matriz de transformação possui um tamanho fixo de 4 linhas e 4 colunas, e, dado que a complexidade da multiplicação de duas matrizes 4×4 , contando apenas as multiplicações envolvidas, é da ordem de 64 operações, temos que a complexidade da obtenção do novo estado do sistema é

$$C_S = 64G \quad (4.3)$$

A atualização dos pesos da rede neural é dependente do cálculo da variação de cada peso da rede. Esta variação é dada pelas equações 3.7 e 3.11.

O cálculo da variação dos pesos na camada de saída, como pode-se ver pela equação 3.7, é dependente do cálculo dos erros correspondentes ao estado do sistema. Os erros referentes ao estado do sistema são calculados diretamente a partir da multiplicação das matrizes de transformação e de derivação do braço. A matriz que contém as derivadas dos erros referentes ao estado do sistema em relação a uma junta genérica k é dada por

$$M_{\frac{\partial \epsilon_S^2}{\partial \Delta_k}} = M_1 M_2 \dots D_k M_{G-1} M_G$$

Uma vez que este erro deve ser calculado em relação a cada junta separadamente, esta operação deverá ser realizada tantas vezes quantas forem os graus de liberdade do braço, ou seja, G vezes. A complexidade derivada desde cálculo pode então, levando em consideração a equação 4.3, ser escrita como

$$C_{dS} = 64 G^2 \quad (4.4)$$

A complexidade da variação propriamente dita dos pesos da camada de saída da rede neural, levando em consideração as equações 3.7 e 4.4, e sendo L_L o número

de neurônios da penúltima camada da rede neural, pode então ser escrita como

$$C_{w_{kj}^o} = L_L (64 G^2) \quad (4.5)$$

O cálculo da variação dos pesos das camadas intermediárias da rede, dado pela equação 3.11, bem como a atualização dos pesos propriamente dita, são dependentes somente da topologia da rede. Seja L_i a camada cujos pesos serão atualizados. Então pode-se escrever que a complexidade desta atualização é

$$C_{w_{ji}^h} = L_{i-1} L_i (L_i L_{i+1})$$

ou ainda,

$$C_{w_{ji}^h} = L_{i-1} L_i^2 L_{i+1} \quad (4.6)$$

ou ainda, para toda a rede neural,

$$C_{\Delta pesos} = I L_1^2 L_2 + \dots + L_{i-1} L_i^2 L_{i+1} + \dots + L_{L-1} L_L^2 O + L_L (64 G^2)$$

sendo $A = \max(I, L_1, \dots, L_L, O)$, temos que a complexidade da atualização dos pesos em toda a rede neural é, no pior caso, da ordem de

$$C_{\Delta pesos} = L A^4 + A 64 G^2 \quad (4.7)$$

Combinando as equações 4.1, 4.2, 4.3 e 4.7, temos que

$$C = (L + 1) A^2 + 64 G + L A^4 + A 64 G^2$$

Mas, de acordo com o problema tratado, sabe-se que $G = O$, ou seja, o número de graus de liberdade do braço, por uma questão de definição do sistema de controle, é sempre igual ao número de elementos da última camada da rede. Nas implementações práticas de redes neurais, o número de neurônios das camadas intermediárias é sempre maior que o número de neurônios das camadas de entrada e saída. Como A corresponde ao número de neurônios da camada portadora de maior número de neurônios da rede, temos que $G < A$. Assim, a operação que demanda a maior complexidade de cálculos é a atualização dos pesos da rede neural a cada ciclo de controle. Desta forma, podemos escrever que a complexidade algorítmica de um ciclo do controlador neural é da ordem de

$$C \approx L A^4$$

isto é, a complexidade do algoritmo de controle depende diretamente da topografia da rede neural utilizada.

Obviamente, uma vez fixada a estrutura da rede neural utilizada, o número de operações envolvidas em cada ciclo de controle passa a ser conhecido e é fixo ao longo de todo o processo. Esta análise de complexidade serve, então, para demonstrar o modo pelo qual a complexidade do processo de controle aumenta conforme as modificações realizadas na estrutura topológica da rede neural.

4.5.2 Espaço de Memória

A complexidade da memória utilizada é um fator importante na análise de um algoritmo. Se o *tempo* é um aspecto importante na resposta de um sistema, o *espaço* também é um aspecto crítico. Algoritmos que contêm algum tipo de aprendizado são conhecidos por muitas vezes requererem um espaço muito grande de memória. Este espaço irá fatalmente traduzir-se em excessivo tempo de acesso.

O espaço de memória M utilizado pode ser dividido em duas partes: o espaço utilizado pela rede neural e o espaço utilizado para armazenar a informação sobre o sistema. Ou seja,

$$M = M_{rede} + M_S \quad (4.8)$$

O espaço de memória utilizado pela rede consiste dos elementos de cada camada da rede e do espaço reservado para armazenar os valores dos pesos das conexões. O espaço de memória reservado para os elementos de cada camada é simplesmente a soma destes elementos. Ou seja, se a camada de entrada possui I elementos, a camada de saída O elementos e cada uma das L camadas intermediárias L_1, \dots, L_L elementos, respectivamente, temos que o espaço de memória utilizado será dado por

$$M_{elem} = I + L_1 + L_2 + \dots + L_L + O$$

Sendo $A = \max(I, L_1, \dots, L_L, O)$, podemos ainda considerar que, para o pior caso, tem-se

$$M_{elem} = (L + 2)A$$

O espaço de memória utilizado para armazenar os pesos das conexões corresponde ao número de conexões existente na rede, isto é,

$$M_{pesos} = (L_1 I) + (L_2 L_1) + (L_3 L_2) + \dots + (L_{L-1} L_{L-2}) + (L_L L_{L-1}) + (O L_L)$$

ou ainda, sendo $A = \max(I, L_1, \dots, L_L, O)$, obtém-se

$$M_{pesos} = (L + 1) A^2$$

Assim, o espaço de memória ocupado pela rede pode ser escrito como

$$M_{rede} = (L + 2)A + (L + 1) A^2 \quad (4.9)$$

O espaço ocupado pelo armazenamento do estado do sistema corresponde aos valores das juntas do braço e às matrizes de transformação e derivação referentes ao tipo das juntas, ou seja,

$$M_S = G + 32G \quad (4.10)$$

Combinando as equações 4.8, 4.9 e 4.10, temos que

$$M = (L + 2)A + (L + 1) A^2 + 33G$$

Como os valores de A , L e N permanecem constantes ao longo de todo o processamento, pode-se dizer que $M \approx c$, onde c é um fator constante, isto é, o espaço alocado de memória permanece constante ao longo de todo o processo.

Capítulo 5

Resultados Obtidos

Neste capítulo são apresentados os resultados obtidos no controle neural do braço robótico através de testes realizados. Estes testes visam avaliar o sistema de acordo com critérios de convergência, estabilidade, repetitividade e geração de trajetórias. A análise dos resultados obtidos neste trabalho será inteiramente realizada sob a perspectiva da análise de comportamento estatístico e não de uma maneira formal, devido às próprias características do sistema de controle implementado.

5.1 Métodos Teóricos *versus* Métodos Empíricos de Análise

Um dos pontos fortes da pesquisa em redes neurais é a utilização de técnicas matemáticas que incorporam, simultaneamente, representações simbólicas e uma metodologia experimental, como as das técnicas tradicionais de inteligência artificial. Em sistemas complexos, não-lineares e abertos, estas características são muito importantes, devido à natureza do problema em questão, mas não existem garantias formais de que uma solução seja sempre encontrada, nem sequer garantias de que uma solução exista.

Obviamente é desejável que seja mantido o rigor matemático no tratamento dos problemas, mas uma abordagem mais experimental e heurística parece ser essencial para o desenvolvimento de aplicações envolvendo sistemas complexos não-lineares. Tal abordagem é caracterizada pela premência de experimentação, mesmo quando garantias teóricas inexistem. Este procedimento é muitas vezes criticado, mas garantias teóricas são freqüentemente obtidas à custa de suposições extremamente restritivas, que, quase sempre, são violadas na prática. Certamente que contar-se com uma teoria rigorosa é importante, e uma constatação válida em qualquer contexto de pesquisa é que esta evolui na ignorância de um quadro teórico relevante. Mas uma metodologia experimental parece ser necessária para o desenvolvimento de aplicações de controle que envolvam sistemas complexos não-lineares.

5.2 Testes de Simulação

Foram implementadas três diferentes maneiras de testar a capacidade do braço atingir um determinado objetivo:

1. Fornecimento da posição e orientação desejadas na forma de uma matriz de transformação

TABELA 5.1 — Configuração do sistema de controle neural

Neurônios	Entrada	12
	Intermediárias	45
		15
	Saída	5
Parâmetros	μ	0,1
	Inicialização dos Pesos	$[-0,3;0,3]$
	$\Delta\theta_k$	5°
	Posição	15
	Orientação	$[0;1]$

2. Fornecimento de uma configuração possível de ângulos $\vec{\theta}$ que definem uma determinada posição e orientação no espaço¹.
3. Geração aleatória da posição/orientação desejada do braço.

As simulações foram realizadas com o objetivo de analisarem-se as características do sistema de controle, bem como as trajetórias geradas pela rede. Dependendo do objetivo da análise, foram usadas diferentes maneiras de geração de objetivos. Seguem-se os resultados obtidos nessas simulações.

5.3 Convergência

Para análise da convergência do manipulador para a situação final desejada, utilizou-se uma geração aleatória de posições e de orientações do braço no espaço. Esta geração foi feita gerando-se vetores de ângulo $\vec{\theta}$ e tomando-se a matriz gerada por esta configuração como a configuração final desejada para o braço. Note-se que, devido à configuração geométrica do braço, os ângulos finais podem ser diferentes de $\vec{\theta}$, e a configuração desejada ainda assim pode ser atingida. A geração de posições e orientações feita deste modo garante a existência de pelo menos uma configuração do robô que permita atingir a posição e a orientação desejadas.

A configuração do sistema de controle neural utilizado para a realização dos testes encontra-se na tabela 5.1. Nesta tabela temos o número de neurônios em cada camada da rede neural, o parâmetro de aprendizagem (μ), o intervalo de inicialização dos pesos da rede, o valor de modificação dos ângulos das juntas do braço robótico ($\Delta\theta_k$), e os parâmetros para geração de erro de posição e de orientação. Note-se que o parâmetro de orientação varia no tempo, entre os valores fornecidos pela tabela, enquanto que o de posição permanece fixo.

Os resultados obtidos com esta metodologia encontram-se na tabela 5.2. Os valores médios obtidos apresentados nesta tabela são decorrentes dos resultados da execução de 500 testes distintos para cada valor de erro máximo permitido imposto. Cada um destes 500 testes correspondeu à geração aleatória de uma configuração objetivo, com o subsequente movimento do braço em direção à configuração gerada. Os elementos de cada coluna da tabela possuem o seguinte significado:

¹Note-se que esta configuração irá gerar uma matriz de transformação que será usada como objetivo. O robô pode atingir este objetivo com uma configuração de braço diferente de $\vec{\theta}$

- **Erro Máximo** — O erro máximo corresponde à distância mínima entre a posição da garra e a posição-alvo, definida por forma a considerar-se que o braço tenha atingido o objetivo.
- **Média de Iterações** — número médio de iterações necessárias para o término da tarefa. O número máximo de iterações permitido para a execução de cada tarefa é definido como sendo 1000. Após a execução de 1000 iterações, mesmo que a rede não tenha conseguido posicionar o braço com um erro menor do que o estipulado, a tarefa é interrompida. Este número de configurações foi escolhido como limite por considerar-se o tempo médio necessário para o processamento de 1000 iterações (cerca de dois minutos e meio no computador utilizado) o tempo máximo aceitável para uma tarefa de posicionamento.
- **Erro Médio** — erro médio correspondente a cada variável espacial (x, y, z) , bem como ao erro médio espacial, que corresponde à distância euclidiana entre a localização final da garra e o objetivo requerido como sua posição final.
- **Variância** — A variância é uma medida da dispersão dos valores do erro final gerado pelo sistema em relação à média deste erro. Se os valores de erro tendem a concentrar-se perto da média, a variância é pequena; caso os valores venham a tender a se afastarem da média, a variância é grande. A variância é dada pela expressão

$$\sigma^2(X) = E [(X - E(X))^2]$$

onde X é a variável considerada (neste caso o erro médio espacial obtido para cada um dos 500 testes realizados), e E representa a função que designa o valor médio do seu argumento.

- **Erro Relativo** — O erro relativo refere-se à diferença relativa entre o erro máximo espacial permitido e o erro médio espacial obtido nas simulações realizadas. Este valor é importante na análise do sistema de controle, uma vez que, mesmo que o erro médio absoluto seja pequeno, pode não ser aceitável em termos relativos considerando-se o valor esperado para o erro máximo permissível. O erro relativo é dado pela expressão

$$\varepsilon_{rel} = \left| \frac{\varepsilon_{m\acute{a}x} - \varepsilon_{m\acute{e}d}}{\varepsilon_{m\acute{a}x}} \right|$$

onde $\varepsilon_{m\acute{a}x}$ é o erro máximo admitido para a tarefa em questão (primeira coluna da tabela) e $\varepsilon_{m\acute{e}d}$ é o erro médio espacial obtido nas simulações efetuadas (correspondente à sexta coluna da tabela).

Pode-se notar pela análise da tabela, que, como seria de se esperar, a um maior nível de exigência quanto ao erro máximo admitido na execução de uma tarefa, corresponde um maior número necessário de iterações para se atingir o objetivo proposto. O braço robótico é capaz de convergir rapidamente para um erro razoavelmente pequeno, mas demora algum tempo até atingir a precisão desejada, o que deve-se ao próprio algoritmo de adaptação da rede neural. Quanto mais próximo o braço estiver do seu objetivo menores serão os erros de posição e orientação gerados

TABELA 5.2 — Convergência do sistema de controle

Erro Máximo Permitido (mm)	Média de Iterações	Erros Médios Obtidos (mm)				Variância σ^2	Erro Relativo
		x	y	z	espacial		
0,01	920,81	0,46	0,52	0,63	1,15	7,96	114,19
0,05	867,29	0,47	0,53	0,63	1,17	7,93	22,31
0,10	831,33	0,48	0,54	0,64	1,19	7,88	10,86
0,50	694,01	0,57	0,64	0,76	1,40	7,46	1,80
1,00	603,21	0,70	0,78	0,94	1,72	6,94	0,72
1,50	524,37	0,83	0,95	1,17	2,09	6,45	0,40
2,00	465,32	1,00	1,13	1,40	2,50	5,98	0,25
2,50	402,82	1,16	1,32	1,67	2,93	5,56	0,17

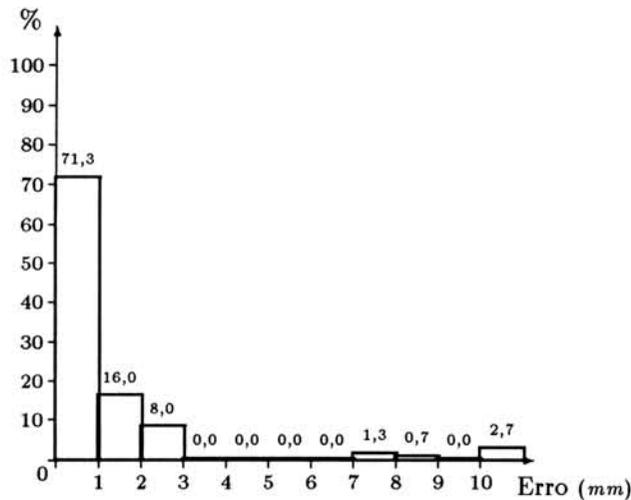


FIGURA 5.1 — Distribuição dos erros relativos à convergência

Em consequência disso, menores serão as alterações nos pesos da rede, tornando a convergência mais lenta quando o sistema estiver próximo do seu objetivo do que quando dele estiver afastado. Uma maneira de contornar este problema é aumentar os valores dos parâmetros referentes aos erros de posição e orientação da rede. Contudo, um certo cuidado é necessário, pois esta modificação não deve gerar oscilações nas etapas finais do processo de posicionamento.

A distribuição dos erros obtidos nos testes a que se refere à primeira *linha* da tabela 5.2 é apresentada na figura 5.1. Como o erro exigido neste caso (0,01mm) é bastante baixo, esta distribuição mostra a real capacidade do braço atingir a localização especificada. Através destes dados, podemos notar que a grande maioria das amostras concentraram-se em erros abaixo de 1mm. Apenas 2,7% dos posicionamentos simulados revelaram erros acima de 10mm. O pior caso testado, igual para todas as simulações, independentemente do erro máximo tolerado, apresentou um erro de deslocamento espacial de 20,71mm. Nas amostras que obtiveram um erro relativamente grande, este fato deveu-se não à falta de convergência do sistema, mas à demora deste atingir o objetivo, sendo a tarefa interrompida na milésima iteração.

Este tipo de situação pode ocorrer sob diversas circunstâncias. Uma vez que a inicialização dos pesos da rede é aleatória, estes podem determinar uma dificuldade

maior na convergência para determinada situação desejada. Outra situação em que uma demora na convergência se apresenta é quando o objetivo localiza-se nas vizinhanças do ponto de partida. Este fato determina que a convergência seja lenta desde o início das iterações e que não seja gerada uma certa inércia² necessária para levar a garra até seu objetivo. Esta inércia poderia ser gerada, por exemplo, fazendo-se o valor do parâmetro de posição ser consideravelmente mais alto do que aquele utilizado, de forma fixa, na realização dos testes. Como este parâmetro esteve fixo ao longo de todos dos testes, tal não ocorreu.

Também é interessante notar que as curvas de distribuição dos erros obtidos nas tarefas de posicionamento para as simulações cujo erro máximo permitido é menos exigente (mais alto) são bastante semelhantes às da figura 5.1. À medida que o erro máximo admitido aumenta, os erros médios obtidos também aumentam, pela razão de que a tarefa é interrompida num ponto em que a garra está dentro da esfera de localização admitida para o término da tarefa, porém poderia aproximar-se um pouco mais do objetivo final. E é somente nisso que as curvas diferem um pouco, mas os objetivos que não foram atingidos com um erro pequeno são iguais em todas as simulações. Este fato demonstra que existe um refinamento a ser feito no controlador, de modo que o processo final de aproximação da garra ao objetivo seja otimizado, levando o braço a convergir para o seu objetivo de modo mais rápido e eficiente.

Pode-se ver ainda que a diferença entre os erros espaciais obtidos para uma exigência de um erro admissível muito pequeno ($0,01mm$) e para um erro razoavelmente grande ($2mm$) não é muito significativa em termos absolutos (nos testes realizados, variou de $1,15mm$ a $2,93mm$). Na realidade, dependendo da exigência de precisão para a tarefa em questão, estes valores podem ser até mesmo irrelevantes. Porém, uma grande diferença encontrar-se-á no número de ciclos de controle necessários para atingir o ponto objetivo. Esta diferença é substancial em termos de processamento computacional, sendo que os posicionamentos com um erro máximo de $2mm$ requerem quase metade das iterações necessárias ao posicionamento com uma precisão muito grande ($0,01mm$). É neste ponto que a diferença torna-se significativa. A execução de um ciclo de controle, se a rede neural implementada possuir uma estrutura razoavelmente grande, exigirá um tempo que talvez não esteja disponível e que talvez não seja necessário, uma vez que o controlador é capaz de posicionar o braço com um erro aceitável em menor número de iterações. Ainda, como pode-se ver, o erro relativo tende rapidamente a zero à medida que o erro máximo admitido cresce. Aqui, novamente, pode-se ver o quanto o controlador neural pode ter sua aplicabilidade melhorada se heurísticas mais refinadas forem a ele adicionadas.

Apesar de o erro médio encontrado na realização das tarefas de posicionamento ter ficado sempre acima do erro máximo admitido, pode-se considerar que este erro foi relativamente baixo na média geral. Um erro de $2mm$ corresponde a aproximadamente $0,17\%$ do espaço de trabalho do robô medido diametralmente. Em grande parte das tarefas de manipulação de objetos, que não exigem uma grande precisão no posicionamento da garra, este erro é bastante aceitável.

O valor relativamente alto da variância pode ser explicado pelo grande número de erros com valores inferiores a $1mm$. Uma vez que a média foi sempre mais alta

²Novamente, neste contexto, o termo *inércia* possui um significado diverso da quantidade física designada por tal termo

que este valor, devido ao fato de existirem amostras que geraram erros mais altos, os valores afastam-se da média, ainda que isso ocorra de uma maneira favorável aos resultados obtidos em termos da precisão do controle da garra. O sistema pode, então, ser considerado robusto, uma vez que é capaz de convergir para a posição espacial desejada em todas as situações apresentadas e, na grande maioria das situações, mostra-se capaz de executar a tarefa que lhe é exigida no tempo devido.

Novas heurísticas podem e devem ser ainda desenvolvidas para melhoria destes resultados, mas pode-se concluir que a garra do robô pode, de fato, ser corretamente posicionada através da técnica de controle proposta.

5.4 Estabilidade

Para análise da estabilidade do sistema de controle, foram realizados diversos testes, tanto com geração aleatória de objetivos, quanto com a pré-determinação de objetivos. Enquanto que a análise de convergência pôde ser realizada automaticamente, a análise de estabilidade, ao contrário, foi realizada através da visualização do movimento do braço e do valor do erro a cada instante. Se o gráfico do erro espacial (distância euclidiana da garra ao objetivo) corresponder aproximadamente a uma curva assintótica, considera-se que o sistema conseguiu atingir seu objetivo de uma maneira estável. Se, pelo contrário, a curva de erro oscilar continuamente, então o sistema, apesar de eventualmente conseguir atingir o seu objetivo, não será considerado suficientemente estável.

Nas primeiras análises, existiam certos objetivos que provocavam uma oscilação muito grande do braço. Para tal problema ser resolvido, teve-se que proceder a um exame mais detalhado das características da rede neural de controle implementada, bem como de seus parâmetros.

5.4.1 A Função de Ativação e os Valores dos Pesos

Os pesos da rede neural *Attentional Mode* são inicializados com valores aleatórios. Estes valores são gerados, normalmente, dentro do intervalo $[-1, 1]$, pois todos os valores da rede, tanto os de entrada, como os de saída, estão normalizados neste intervalo.

A função de ativação utilizada pela rede é a função *tangente hiperbólica*. Como pode-se notar pelo seu gráfico na figura 3.3, ela tende para 1 (ou para -1) muito rapidamente. Desta forma, uma vez que existem vários neurônios nas camadas intermediárias, se os valores dos pesos forem muito altos, as saídas sempre irão tender para 1, resultando na modificação máxima dos ângulos das juntas. Este fato torna os movimentos bruscos e rápidos e, embora a tendência seja sempre a de convergir para o valor final, oscilações muito grandes e indesejáveis podem ocorrer dependendo do objetivo.

Na figura 5.2, temos, acima, três diferentes perspectivas do braço em movimento, e, abaixo, a variação dos valores de x , y e z da garra. Pode-se notar que os valores das coordenadas da garra oscilam de forma bastante acentuada, principalmente no início do movimento. Essa oscilação inicial gera uma inércia que, por sua vez, gera mais oscilações ao longo do movimento. Note-se que o braço, independentemente da inércia inicial gerada, consegue atingir o seu objetivo, mas este tipo

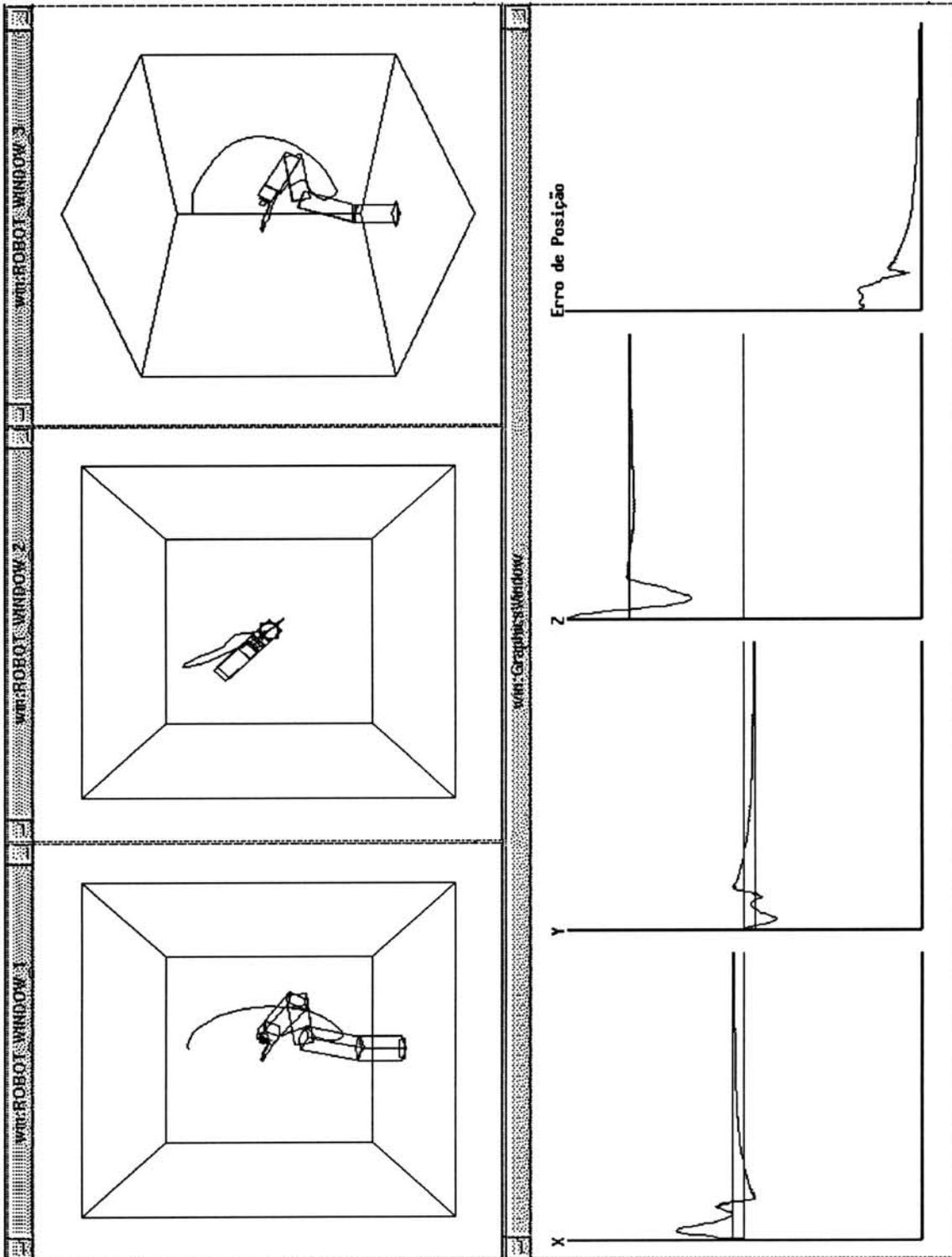


FIGURA 5.2 — Oscilação no Sistema de Controle

de comportamento não é desejável no sistema.

A fim de minimizar este problema de oscilação, os pesos da rede são inicializados com valores pequenos (tipicamente no intervalo $[-0,3; 0,3]$). Desta forma, os valores das juntas variam de forma menos brusca, somente atingindo seus valores máximos se estas estiverem realmente longe de sua posição ideal. Desta forma, os gráficos de posição, orientação e erro aproximam-se de uma assíntota, tornando o sistema mais estável, como pode-se ver na figura 5.3.

5.4.2 Variação dos Ângulos das Juntas

A cada iteração da rede, esta retorna para cada junta do robô um valor pertencente ao intervalo $[-1, 1]$. Para transformar este valor na variação angular propriamente dita, o valor fornecido pela rede é multiplicado por um valor máximo de variação de cada ângulo a cada instante. Este valor depende do problema tratado e pode ser visto como um fator determinante da velocidade do braço, ou seja, quanto maior este valor, maior a velocidade do movimento.

Variações na velocidade do manipulador são importantes na realização de diversas tarefas. Por exemplo, se o manipulador está realizando uma tarefa, que constitui-se em apanhar um objeto, uma primeira aproximação ao objeto a ser apanhado pode ser realizada em velocidade alta. Já a tarefa de apanhar o objeto propriamente dito deve ser feita a uma velocidade mais baixa, para garantir uma aproximação correta e um apanhamento e sujeição seguros.

A variação máxima dos ângulos das juntas deve ser, ela própria, um valor bem pré-estimado. Valores muito altos tornam as variações de posição e de orientação do braço no espaço muito grandes, comprometendo a estabilidade do sistema. Por outro lado, valores muito baixos comprometem a convergência, exigindo muitas iterações da rede para se atingir um valor de erro aceitável. Durante os experimentos realizados, verificou-se que os valores aceitáveis de variação da posição angular das juntas são da ordem de 3 a 10 graus.

5.5 Repetitividade

A repetitividade de um sistema de controle diz respeito à capacidade deste conseguir resultados coerentes ao longo do tempo. É necessário que o sistema tenha a capacidade de repetir um determinado comportamento em diferentes tentativas, mesmo que sob diferentes circunstâncias. Uma medida fundamental para a avaliação da capacidade do sistema repetir um desempenho é a variância do erro espacial final encontrado após o sistema ser levado a uma determinada configuração-objetivo por diversas vezes. O objetivo do sistema de controle é conseguir valores coerentes, ou seja, com uma variância pequena ao longo do tempo. Note-se que o objeto de análise aqui não é o valor do erro em si, mas a capacidade do sistema em repeti-lo para uma determinada configuração testada.

Para o teste da repetitividade da resposta do sistema, procedeu-se ao teste de variadas configurações-objetivo por diversas vezes, variando-se os pesos da rede neural a cada início de teste. Esta variação dá-se de modo a garantir que a rede seja capaz de posicionar o braço na posição correta independentemente das suas condições iniciais.

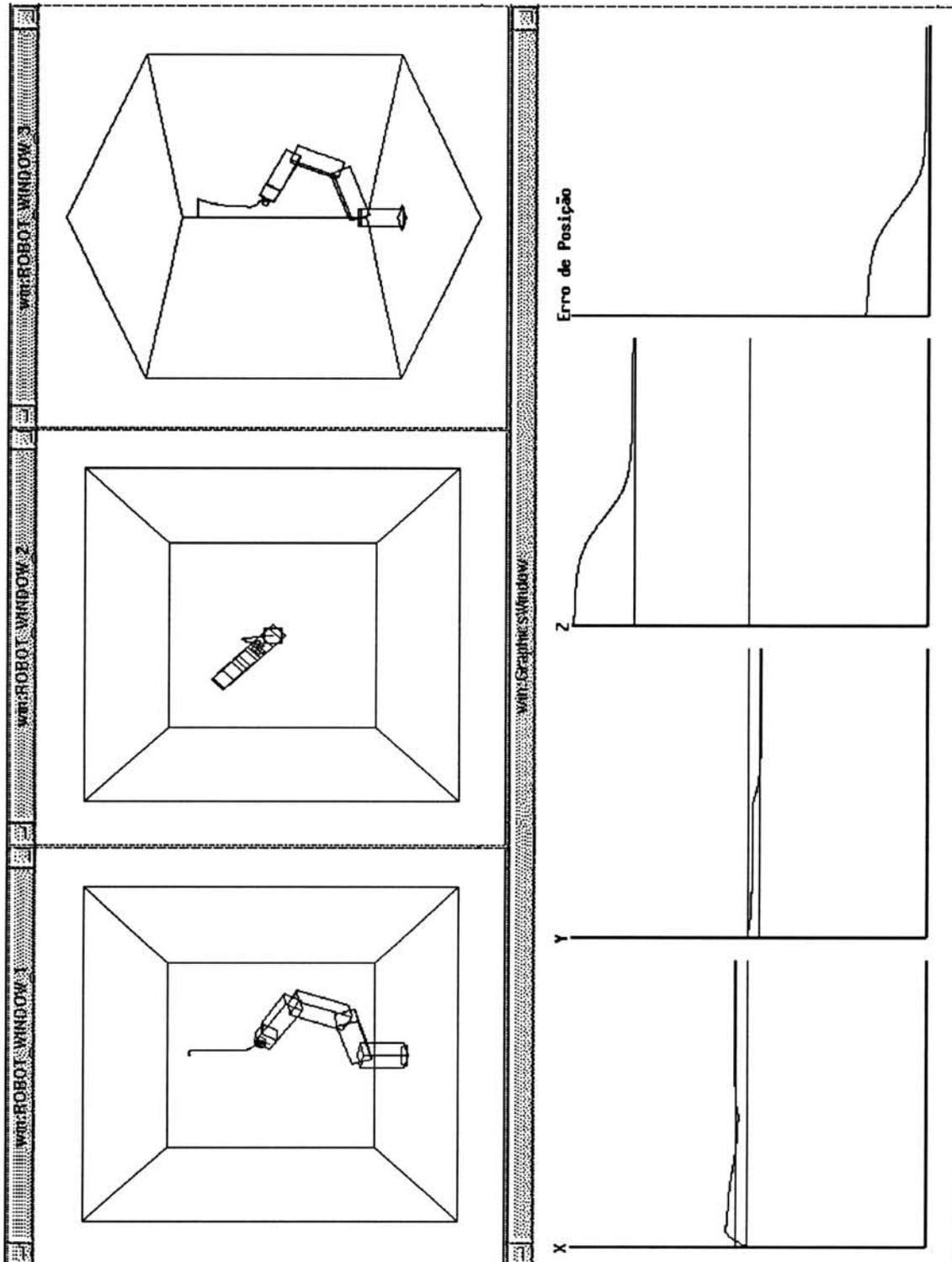


FIGURA 5.3 — Minimização do Problema de Oscilação

TABELA 5.3 — Repetitividade do sistema de controle

Amostra	Posição			Média de Iterações	Erros Médios Obtidos				Variância σ^2
	x	y	z		x	y	z	espacial	
1	0,0	400,0	330,0	300,1	0,08	0,79	1,23	1,47	0,00
2	0,0	-580,0	150,0	1000,0	0,00	2,67	0,02	2,67	0,23
3	150,4	-223,0	449,6	622,5	0,95	1,14	0,73	1,65	0,14
4	0,7	20,7	647,2	1000,0	6,17	17,65	0,05	18,70	1,67
5	-299,9	-69,2	573,2	613,3	0,73	0,20	1,37	1,57	0,15

Os resultados mostram que o sistema de controle é capaz de manter-se coerente ao longo de diversas tentativas, repetindo os resultados obtidos, tanto quando o erro é pequeno, como quando é razoavelmente grande, como pode ser verificado na tabela 5.3. Nesta tabela, o erro espacial máximo admitido para se considerar a tarefa como concluída é de 1mm; cada configuração-objetivo foi repetida 50 vezes e a configuração do sistema de controle neural é a mesma da tabela 5.1.

Note-se que, nas amostras 2 e 4 da tabela 5.3, o sistema não foi capaz de atingir seus objetivos dentro do número máximo de ciclos de controle estipulado. Note-se também que o erro médio total da amostra 4 é razoavelmente alto, mas, em ambas as amostras a variância é baixa. Isto significa que o sistema é incapaz de atingir o ponto-objetivo com a precisão requerida, mas é capaz de manter coerência em todas as tentativas. Reportando-se à figura 1.1, observe-se que este é o caso de um objetivo para o qual o sistema possui repetitividade, mas não precisão.

5.6 Geração de Trajetórias

Devido à restrição para o deslocamento do braço integrada à rede (apresentada na seção 3.1.1), garante-se que a trajetória escolhida pela rede entre dois pontos no espaço é sempre a que traduz-se em menor esforço de movimento, ou seja, o caminho mais curto. Este fato tem conseqüências importantes no que diz respeito às trajetórias geradas pelo sistema de controle.

Se o ponto correspondente à posição atual da garra e o ponto correspondente ao objetivo proposto forem tais como mostra a figura 5.4, pode-se notar que o caminho mais próximo para o ponto-objetivo é o mostrado pela trajetória possível T_c . Porém, esta trajetória determinará que, se o robô tentar atingir o ponto, os limites de operação do braço sejam excedidos. A restrição de relaxamento do braço entrará em ação no decorrer da trajetória T_c , e o braço entrará num estado de oscilação entre a tentativa de aproximação do objetivo e o impedimento de que o valor da junta seja excedido. A trajetória desejada, então, deverá ser algo como a curva T_d , que, embora provoque um deslocamento maior, não produz nenhum problema de configuração para o robô. Por esta razão, alguns pontos intermediários deverão ser gerados para garantir uma aproximação da trajetória desejada.

Este tipo de problema aparece freqüentemente em sistemas que funcionam com métodos matemáticos de minimização de uma função. Como não existe nenhuma espécie de planejamento, podem aparecer estados que são conhecidos na literatura como *stuck states*. Estes estados aparecem quando o algoritmo de minimização encontra um mínimo local da função a ser minimizada, fazendo com que o sistema entre num estado de onde não há saída sem alguma espécie de estratégia que permita

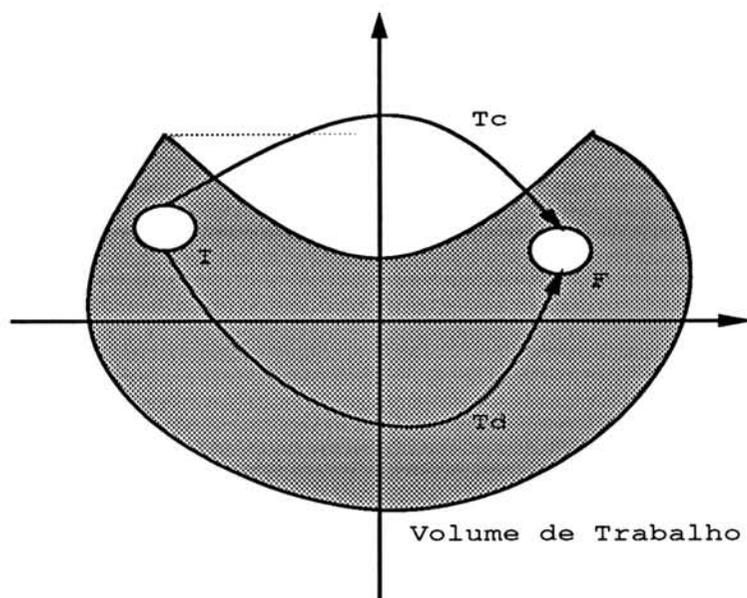


FIGURA 5.4 — Geração de trajetórias do sistema

ao sistema dele escapar.

As figuras 5.5, 5.6 e 5.7 mostram algumas das trajetórias geradas pelo sistema de controle. Nestas figuras, pode-se observar o braço em três diferentes perspectivas, os valores das coordenadas (x , y e z) da garra ao longo do tempo, e o erro espacial correspondente à distância euclidiana entre a posição real e a posição desejada da garra durante o processo de simulação. Essas trajetórias mostram o braço movendo-se em direção a três objetivos distintos, prestando-se ainda, à ilustração de como o sistema comporta-se ao longo do tempo. Note-se que, apesar de existirem pequenas oscilações no final do movimento, a convergência para o ponto-objetivo pode ser considerada estável.

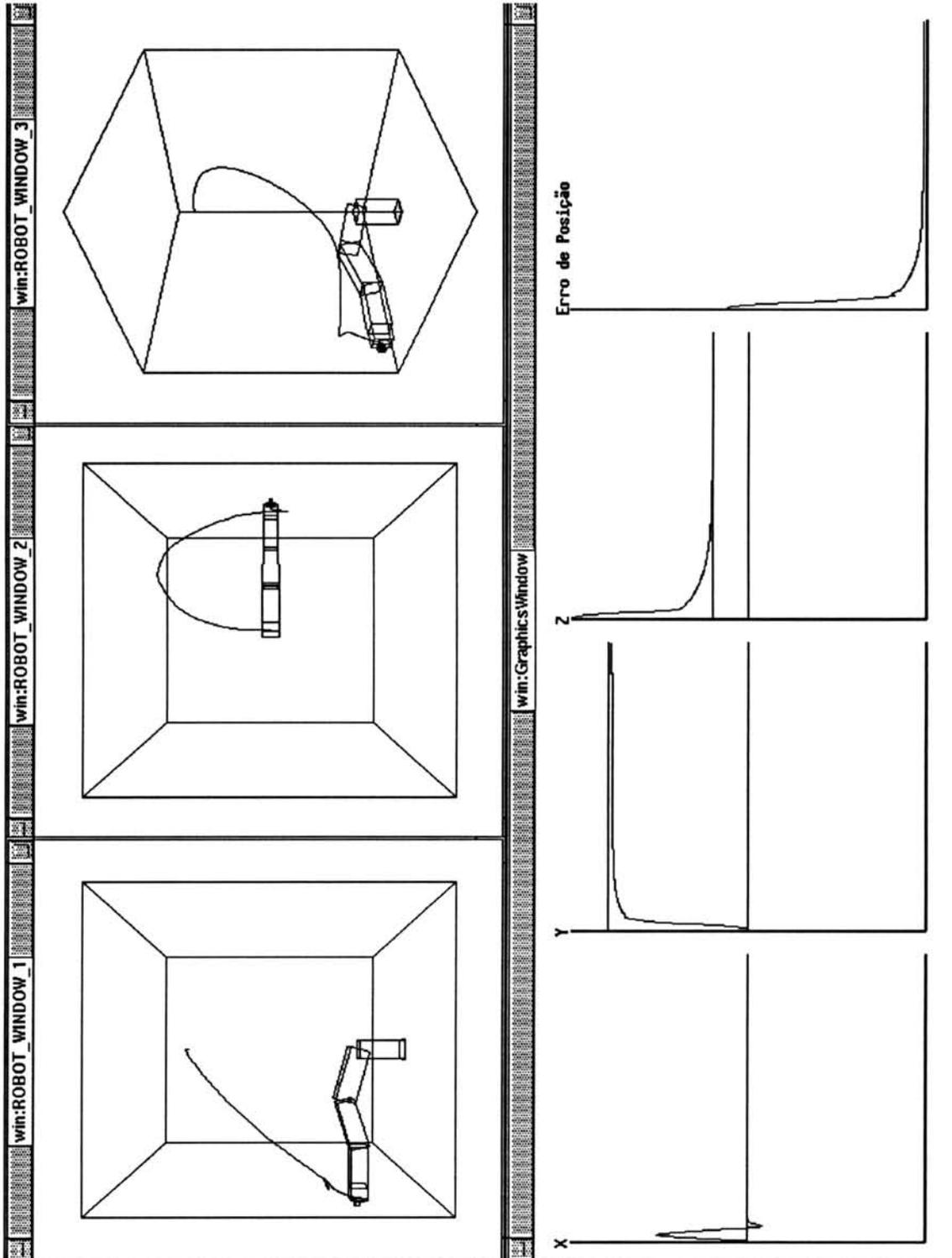


FIGURA 5.5 — Exemplo de trajetória gerada pela rede — I

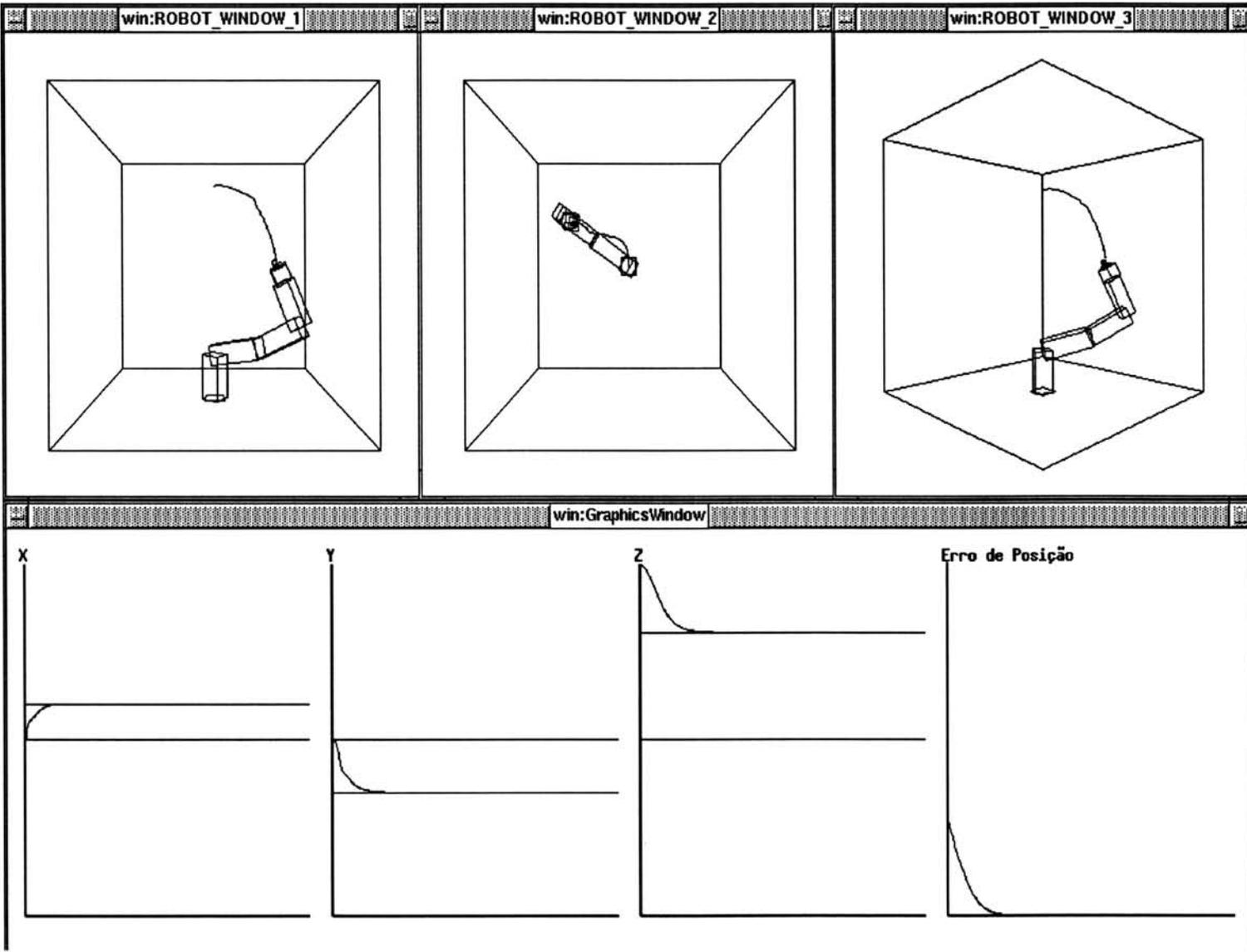


FIGURA 5.6 — Exemplo de trajetória gerada pela rede — II

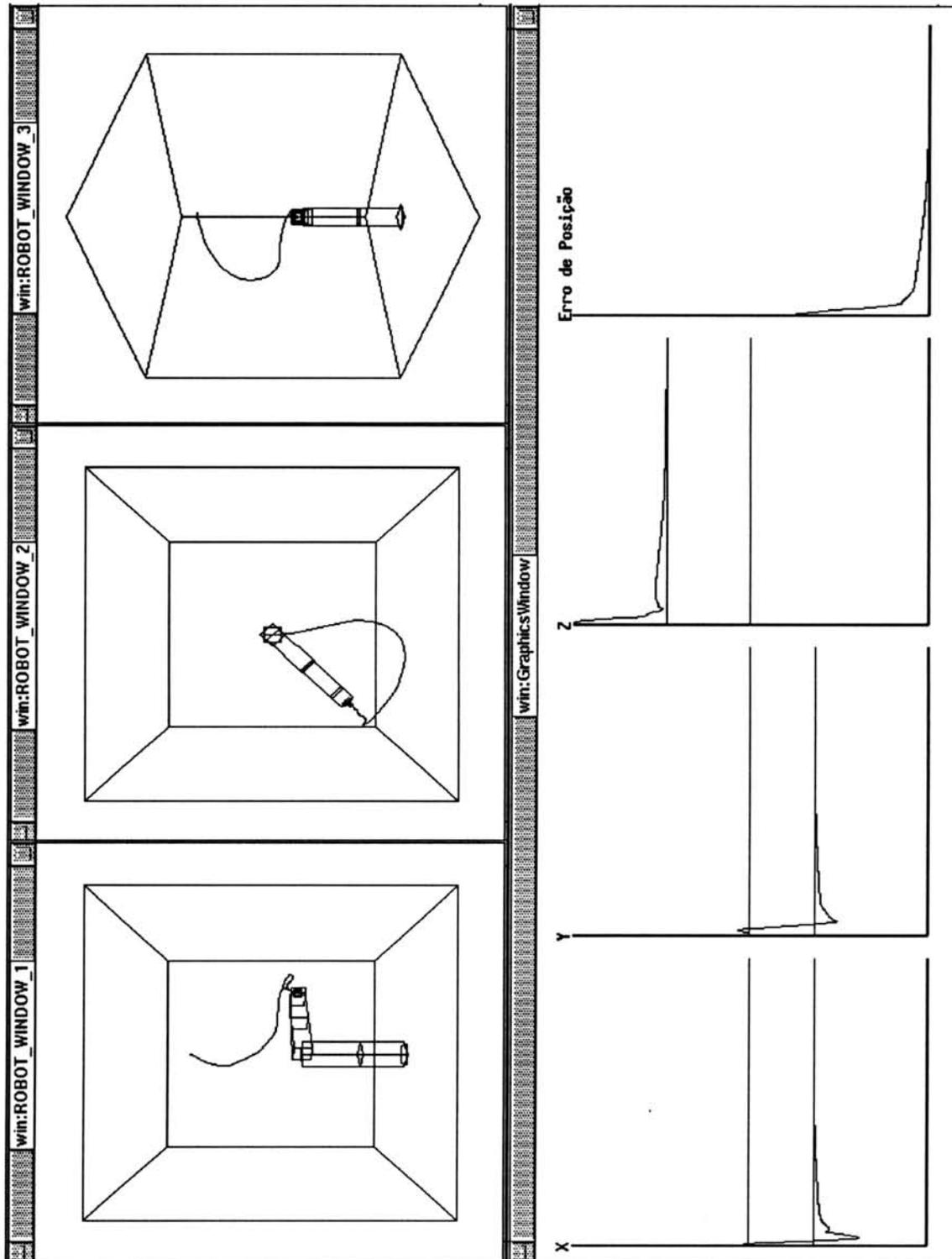


FIGURA 5.7 — Exemplo de trajetória gerada pela rede — III

Capítulo 6

Conclusões

6.1 A aplicação do *controle neural adaptativo* no contexto deste trabalho

Somando as vantagens dos sistemas de controle adaptativo às das redes neurais, os sistemas de controle neural adaptativo tornam-se progressivamente um paradigma para a construção de sistemas de controle não-lineares que operam em ambientes abertos. Embora a pesquisa realizada a respeito destes sistemas encontre-se ainda num estágio muito inicial, esses sistemas vêm conseguindo resultados que podem ser considerados excelentes no que diz respeito à flexibilidade, quando comparados com técnicas de controle tradicionais.

Num sistema linear, o erro médio quadrático é uma função quadrática dos seus parâmetros com um único mínimo (mínimo global). Sistemas não-lineares, por outro lado, podem gerar superfícies de erro com diversos mínimos locais de tal forma que os resultados dependem fortemente dos parâmetros iniciais usados e do processo de adaptação. Apesar destas complexidades, que tornam impossível provar a convergência global dos resultados, como pode-se fazer para modelos lineares, a estimativa *on-line* dos parâmetros em modelos não-lineares parece ser um caminho promissor em problemas de controle.

Neste trabalho foi implementado um sistema de controle neural capaz de guiar um braço robótico de um ponto inicial a um ponto final de seu espaço de trabalho. A técnica utilizada baseou-se na rede neural *Attentional Mode*, que é capaz de adaptar-se em tempo real às modificações no estado do sistema tratado. Esta abordagem permite a definição do controlador baseado somente na estrutura da rede utilizada e no modelo geométrico do braço a ser controlado. Além destas características, o sistema de controle implementado pode ter heurísticas de movimentação incorporadas de maneira direta, facilitando o tratamento de um sistema aberto. Como experimento, foi simulado um braço robótico com 5 (cinco) graus de liberdade, e as características do controlador no que diz respeito à convergência, repetitividade e estabilidade foram testadas, com o intuito de efetuar a validação do modelo. Os resultados encontrados durante este processo serão discutidos a seguir.

6.2 Discussão dos Resultados Obtidos

Devido à complexidade e não-linearidade de um sistema de controle para um braço manipulador com múltiplos graus de liberdade, grande parte das técnicas exis-

tentes não se aplicam em situações de ambiente aberto, por possuírem soluções por demais restritivas. Tais restrições derivam da limitação nas configurações possíveis do braço decorrentes da definição do sistema de controle e da complexidade deste sistema em si, cuja formulação matemática é extremamente trabalhosa e, muitas vezes, não leva em consideração modificações possíveis no estado do ambiente ou no estado do próprio sistema a ser controlado. Num ambiente aberto, devido à presença e movimentação contínua de obstáculos, tais restrições podem levar a uma impossibilidade do sistema de controle atingir determinado objetivo, mesmo quando exista uma solução possível que o sistema não leve em consideração.

6.2.1 Complexidade do Sistema de Controle

A complexidade computacional do sistema de controle demonstrou-se equivalente à complexidade da rede neural utilizada por este sistema, conforme demonstrado na seção 4.5. A complexidade do algoritmo deriva do número de articulações do braço robótico a ser controlado e da conseqüente estrutura da rede neural utilizada. Obviamente que esta é a complexidade de cada ciclo de controle, não existindo prova formal da complexidade global de convergência do movimento do braço para um objetivo pré-determinado. O cálculo da complexidade global foi estimado estatisticamente através de diversos testes de simulação, e tal complexidade mostrou-se relativamente baixa na maior parte dos casos. Os resultados obtidos a respeito da convergência do sistema demonstram que a complexidade pode ser ainda mais baixa com alguns refinamentos do sistema de controle neural.

A memória utilizada pelo controlador também é diretamente proporcional à estrutura da rede neural e é invariante no tempo. Esta característica é bastante importante nos sistemas que possuem capacidade de aprendizado, uma vez que o acúmulo de conhecimento torna a busca de dados onerosa, tornando o sistema lento e ineficiente após um longo tempo de processamento. Uma vez que a estrutura do sistema é fixa e que este aprende de acordo com os resultados obtidos a cada iteração da rede neural, este problema não se apresenta.

6.2.2 Convergência

A convergência de movimentação da garra com relação à configuração-objetivo verificada durante os testes de simulação pode ser considerada satisfatória para tarefas que não requerem um elevado grau de precisão no posicionamento da garra. Note-se que o braço controlado pela rede é capaz de aproximar-se rapidamente das vizinhanças do ponto-objetivo, e tal aproximação dá-se para qualquer objetivo testado. A aproximação final é bastante mais lenta, devido a que os erros gerados pela rede são então muito pequenos e, conseqüentemente, a alteração dos pesos também o é. Tal comportamento pode ser significativamente melhorado se heurísticas apropriadas forem incorporadas à rede, de modo a que esta aproximação final torne-se mais rápida, sem que, no entanto, isso determine a ocorrência de oscilações neste processo.

As curvas de convergência obtidas a partir dos testes realizados mostram que o erro final obtido para uma tarefa de posicionamento tem a ver com os valores iniciais dos pesos da rede, os valores dos parâmetros da rede e com a localização espacial do objetivo. Verifica-se que, quanto maior for o erro máximo admitido,

mais rapidamente o controlador fará o braço atingir seu objetivo, pois a demora na convergência encontra-se localizada na aproximação final à localização espacial desejada. Adicionalmente, o erro relativo obtido para as tarefas de posicionamento tende rapidamente a zero quando o erro admissível cresce.

Em termos do problema de controle de braços de robôs, observe-se que, à medida que a exigência sobe do intervalo de precisão requerido nas tarefas de montagem de equipamentos eletro-mecânicos delicados ou de precisão (0,05 a 0,10mm) para o das tarefas menos exigentes em termos de precisão, como a soldagem ou a pintura (a partir de 0,5mm) ou a movimentação de peças e dispositivos entre equipamentos (a partir de 1mm), o erro relativo obtido tornou-se progressivamente menos significativo.

6.2.3 Estabilidade

A estabilidade de um sistema de controle é fundamental para a análise da utilidade deste sistema para as diferentes aplicações a serem por ele controladas. Um sistema para o qual oscilações podem comprometer a sua utilidade devem possuir um controlador capaz de gerar soluções diretamente convergentes para o objetivo ou que para ele convirjam com um mínimo de oscilação possível.

A estabilidade obtida nos posicionamentos realizados pelo controlador neural, conforme revelada pelos diversos testes realizados (não tendo sido extensivamente descrita nesta dissertação), demonstrou ser bastante aceitável. As restrições de variação dos ângulos e de deslocamento espacial da garra tiveram um papel fundamental nesses resultados, impedindo o sistema de gerar uma inércia tal que provocasse uma oscilação muito grande dos valores de posição da garra no entorno do objetivo proposto. Desta forma, o controlador neural guia o braço de uma maneira bastante suave para o objetivo proposto e, embora algumas oscilações possam existir, estas não comprometeriam a utilização do sistema no desempenho de tarefas que permitam uma pequena oscilação.

A estabilidade do sistema proposto pode ainda ser melhorada, da mesma forma que a sua convergência. As dificuldades encontradas na convergência final da garra são justamente os fatores que provocam as oscilações do sistema. Acredita-se que, com o aperfeiçoamento das heurísticas a serem utilizadas para garantir uma convergência mais rápida do braço, as oscilações decorrentes sejam consideravelmente minimizadas.

6.2.4 Repetitividade

O sistema de controle provou possuir uma boa repetitividade mesmo quando o erro final absoluto alcançado é relativamente alto. Esta característica do sistema de controle é extremamente importante, uma vez que mostra que o controle é coerente, ou seja, é capaz de repetir um desempenho independentemente das condições iniciais do sistema (configuração do braço e pesos das conexões neurais).

Além da importância da existência de uma boa repetitividade no sistema, os resultados obtidos neste trabalho mostram que existe uma independência entre as capacidades de *convergência* e *repetitividade* do controlador, ou seja, independentemente do valor do erro médio final absoluto encontrado na repetição de uma tarefa de posicionamento, o sistema é capaz de repetir este erro com uma variância muito

pequena em diversas tentativas. Este fato tem implicações importantes no controle neural desenvolvido neste trabalho. Uma vez que a convergência final do braço pode ainda ser melhorada, esta melhoria não deverá interferir na repetitividade do sistema de controle, tornando-o, deste modo, mais eficiente e robusto, sem prejuízo das suas características de eficiência atuais.

6.2.5 Geração de Trajetórias

Devido às restrições incorporadas ao sistema de controle neural, o braço tende a mover-se em direção ao objetivo pela trajetória que minimize o esforço total dispendido pelo braço, ou seja, a garra segue sempre na direção do caminho mais curto. Esta característica nem sempre é vantajosa, uma vez que o caminho mais curto entre a posição atual e a posição desejada da garra pode conduzir a que o braço exceda o limite de rotação de suas juntas. Este problema é relativamente comum em sistemas que utilizam métodos numéricos de minimização de funções, em oposição a um planejamento de movimentos pré-estabelecido, menos genérico e mais difícil de definir. Esta restrição, contudo, não prejudica o sistema, uma vez que um caminho pode ser dividido em alguns pontos intermediários que garantam que o braço siga um caminho que não provoque a ocorrência de situações de excesso dos limites físicos do manipulador.

6.2.6 Definição do Controlador

O sistema de controle neural utilizado é definido somente pela estrutura da rede neural, pelos parâmetros utilizados pela rede e pelo modelo geométrico do braço a ser controlado. Não há necessidade de nenhuma espécie de modelamento matemático da estrutura e do funcionamento do braço, nem de previsão sobre as tarefas que este deva realizar. Qualquer tarefa pode ser levada a cabo sem necessidade de reprogramação do braço ou da introdução de modificações no controlador.

A definição do controle feito desta maneira tem vantagens significativas em relação ao que ocorre com a maioria dos sistemas robóticos. O controlador é capaz de perceber modificações no sistema através de percepção sensorial e de agir de acordo com elas, tornando a definição do controlador muito mais simples. Por outro lado, não há necessidade de reprogramação do braço a cada nova tarefa, como na maioria dos sistemas em operação atualmente, tornando o sistema verdadeiramente flexível, podendo atuar de forma independente.

Além disso, a própria maneira como o controlador neural computa os erros relativos do sistema em relação à situação final desejada constitui-se numa vantagem adicional. Uma vez que os erros são calculados para cada junta separadamente, heurísticas diferentes podem funcionar para cada junta em particular, tornando o sistema ainda mais versátil.

Essa flexibilidade de atuação é importante na medida em que, sendo este sistema inserido num sistema mais amplo, que englobe tomada de decisões, estas não serão dependentes de programações de trajetórias previamente estabelecidas. O sistema de controle aqui apresentado possui, portanto, generalidade e flexibilidade suficientes para poder ser inserido num sistema inteligente mais global.

6.3 Trabalho Futuro

Com relação ao trabalho desenvolvido nesta tese, existem alguns pontos que poderiam ser abordados de uma forma mais completa e outros que não chegaram a ser desenvolvidos, mas que possuem possibilidades bastante promissoras. A seguir, alguns destes pontos serão comentados.

6.3.1 Desvio de Obstáculos

A detecção e desvio de obstáculos é um problema fundamental se o objetivo é possuir um sistema de controle verdadeiramente adaptativo. Os algoritmos numéricos que utilizam a minimização de uma função para desvio de obstáculos sofrem, em certas situações, o problema conhecido como *stuck-state*, ou seja, caem em mínimos locais durante o processo, sem chegar a solução alguma.

Um sistema que seja capaz de desviar de obstáculos, tanto móveis como fixos, deve possuir a capacidade de reconhecer um ambiente e agir de acordo com o mesmo. Para tanto, a utilização de sistemas neurais adaptativos parece ser efetivamente promissora, uma vez que, para além de reconhecer seu ambiente, mesmo na ocorrência de ruídos na informação, o sistema deve ser capaz de adaptar-se a ele à medida que este vai-se modificando.

6.3.2 Dinâmica

O controle dinâmico de um braço manipulador é fundamental para a construção de um protótipo real. Uma vez que, neste trabalho, apenas a cinemática foi tratada, uma abordagem interessante para a realização de trabalhos futuros seria o da realização do tratamento da dinâmica do robô, fazendo-se o braço mover-se não através de modificações dos ângulos das juntas, mas através de torques aplicados a elas.

6.3.2.1 Controle

O braço humano move-se não de maneira a minimizar o espaço percorrido pela sua extremidade (mão), mas minimizando o esforço total dispendido. Este modo de movimento é conhecido na literatura como *critério do torque mínimo*. Uma vez que toda a nossa capacidade de movimentação tem a ver com a dinâmica do nosso braço, com a carga em cada tarefa e com as condições dinâmicas do meio, faz sentido o desenvolvimento de um sistema de controle dinâmico para um braço robótico. Esta abordagem não somente é mais útil do ponto de vista prático, como também é mais plausível do ponto de vista da analogia com sistemas biológicos.

Dadas as características do controle dinâmico, um sistema de controle adaptativo é capaz de solucionar grande parte dos problemas somente resolvidos pelas técnicas de controle conhecidas à custa de muitas simplificações e de cálculos bastante complicados.

6.3.2.2 Simulação

Ainda que a implementação de um sistema de controle dinâmico não seja possível de ser realizada num braço real, pode-se ainda construir um simulador

de braço que leve em conta as características dinâmicas do robô e do ambiente como um todo. Para tal, este simulador deve ser capaz de simular as condições dinâmicas de um ambiente e seu efeito no processo de movimentação. A construção desse simulador poderia ser realizada como uma extensão do ambiente de simulação utilizado nesta tese.

6.4 Considerações Finais

O objetivo deste trabalho foi apresentar um sistema de controle capaz de posicionar um braço mecânico articulado no espaço, de acordo com uma posição e com uma orientação espacial previamente definidas. O braço robótico controlado deveria ser capaz de operar num ambiente aberto e, deste modo, não possuir, em seu controlador, nenhuma limitação referente a configurações possíveis de suas juntas.

Os resultados obtidos neste trabalho revelam uma estratégia promissora no controle de um braço robótico. Embora somente o posicionamento espacial do braço tenha sido tratado e nenhum tratamento de obstáculos tenha sido proposto nesta investigação, foi demonstrado que o sistema de controle construído não possui nenhuma espécie de restrição sobre seu funcionamento.

O sistema de controle apresentado nesta dissertação é capaz de posicionar a garra no espaço com um grau de precisão médio que pode ser considerado aceitável para tarefas de montagem que não exijam uma precisão muito grande. Esta precisão, como foi visto neste texto, pode ainda ser bastante melhorada com a adição de heurísticas apropriadas e com um estudo mais detalhado das características de funcionamento da rede neural *Attentional Mode*, tornando o sistema de controle mais útil do ponto de vista prático. Como o objetivo inicialmente previsto era o da construção de um sistema flexível, pode-se considerar que este foi atingido.

Anexo A

Matrizes de Transformação

Uma matriz de transformação é definida como uma matriz que opera sobre um vetor de posição em um espaço tridimensional e produz um mapeamento entre as coordenadas do sistema de coordenadas rotacionado e/ou transladado $OUVW$ e o sistema de coordenadas de referência $OXYZ$.

Sejam $(\mathbf{i}_x, \mathbf{j}_y, \mathbf{k}_z)$ e $(\mathbf{i}_u, \mathbf{j}_v, \mathbf{k}_w)$ os vetores unitários ao longo dos eixos dos sistemas de coordenadas $OXYZ$ e $OUVW$ respectivamente. Um ponto \mathbf{p} no espaço pode ser representado por suas coordenadas em relação a quaisquer dos sistemas de coordenadas, como

$$\mathbf{p}_{uvw} = (p_u, p_v, p_w)^T \quad \text{e} \quad \mathbf{p}_{xyz} = (p_x, p_y, p_z)^T$$

onde \mathbf{p}_{xyz} e \mathbf{p}_{uvw} representam o mesmo ponto \mathbf{p} no espaço com relação a diferentes sistemas de coordenadas¹.

A.1 Matrizes de Rotação

Uma matriz de rotação é tal que transforma um ponto do referencial $OUVW$, depois deste ter sofrido uma rotação, no ponto correspondente ao referencial $OXYZ$.

Seja \mathbf{R} a matriz de rotação que transforma as coordenadas de \mathbf{p}_{uvw} nas coordenadas expressas em relação ao sistema de coordenadas $OXYZ$ após o sistema de coordenadas $OUVW$ ter sido rotacionado. Isto é,

$$\mathbf{p}_{xyz} = \mathbf{R}\mathbf{p}_{uvw} \tag{A.1}$$

Note-se que o ponto \mathbf{p}_{uvw} moveu-se juntamente com o seu sistema de coordenadas.

Um vetor pode ser definido através de suas componentes. A notação desta definição é dada por

$$\mathbf{p}_{uvw} = p_u \mathbf{i}_u + p_v \mathbf{j}_v + p_w \mathbf{k}_w \tag{A.2}$$

onde p_u , p_v e p_w representam as componentes do vetor \mathbf{p} ao longo dos eixos OX , OY e OZ respectivamente (ou as projeções de \mathbf{p} sobre os eixos respectivos). Assim, usando a definição de produto escalar e a equação A.2, temos que

¹O sobrescrito T sobre vetores e matrizes indica a operação de transposição.

$$\begin{cases} p_x = \mathbf{i}_x \cdot \mathbf{p} = \mathbf{i}_x \cdot \mathbf{i}_u p_u + \mathbf{i}_x \cdot \mathbf{j}_v p_v + \mathbf{i}_x \cdot \mathbf{k}_w p_w \\ p_y = \mathbf{j}_y \cdot \mathbf{p} = \mathbf{j}_y \cdot \mathbf{i}_u p_u + \mathbf{j}_y \cdot \mathbf{j}_v p_v + \mathbf{j}_y \cdot \mathbf{k}_w p_w \\ p_z = \mathbf{k}_z \cdot \mathbf{p} = \mathbf{k}_z \cdot \mathbf{i}_u p_u + \mathbf{k}_z \cdot \mathbf{j}_v p_v + \mathbf{k}_z \cdot \mathbf{k}_w p_w \end{cases}$$

ou ainda, em notação matricial,

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} \mathbf{i}_x \cdot \mathbf{i}_u & \mathbf{i}_x \cdot \mathbf{j}_v & \mathbf{i}_x \cdot \mathbf{k}_w \\ \mathbf{j}_y \cdot \mathbf{i}_u & \mathbf{j}_y \cdot \mathbf{j}_v & \mathbf{j}_y \cdot \mathbf{k}_w \\ \mathbf{k}_z \cdot \mathbf{i}_u & \mathbf{k}_z \cdot \mathbf{j}_v & \mathbf{k}_z \cdot \mathbf{k}_w \end{bmatrix} \begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix}$$

Utilizando esta notação, a matriz \mathbf{R} da equação A.1 é dada por

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_x \cdot \mathbf{i}_u & \mathbf{i}_x \cdot \mathbf{j}_v & \mathbf{i}_x \cdot \mathbf{k}_w \\ \mathbf{j}_y \cdot \mathbf{i}_u & \mathbf{j}_y \cdot \mathbf{j}_v & \mathbf{j}_y \cdot \mathbf{k}_w \\ \mathbf{k}_z \cdot \mathbf{i}_u & \mathbf{k}_z \cdot \mathbf{j}_v & \mathbf{k}_z \cdot \mathbf{k}_w \end{bmatrix} \quad (\text{A.3})$$

Similarmente, pode-se obter as coordenadas de \mathbf{p}_{uvw} a partir das coordenadas de \mathbf{p}_{xyz} :

$$\mathbf{p}_{uvw} = \mathbf{Q} \mathbf{p}_{xyz} \quad (\text{A.4})$$

ou ainda

$$\begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix} = \begin{bmatrix} \mathbf{i}_u \cdot \mathbf{i}_x & \mathbf{i}_u \cdot \mathbf{j}_y & \mathbf{i}_u \cdot \mathbf{k}_z \\ \mathbf{j}_v \cdot \mathbf{i}_x & \mathbf{j}_v \cdot \mathbf{j}_y & \mathbf{j}_v \cdot \mathbf{k}_z \\ \mathbf{k}_w \cdot \mathbf{i}_x & \mathbf{k}_w \cdot \mathbf{j}_y & \mathbf{k}_w \cdot \mathbf{k}_z \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (\text{A.5})$$

Uma vez que o produto escalar é comutativo, pode-se notar pelas equações A.3 a A.5 que

$$\mathbf{Q} = \mathbf{R}^{-1} = \mathbf{R}^T$$

e que

$$\mathbf{Q} \mathbf{R} = \mathbf{R}^T \mathbf{R} = \mathbf{R}^{-1} \mathbf{R} = \mathbf{I}$$

onde \mathbf{I} é a matriz identidade. A transformação dada na equação A.1 (ou na equação A.4) é chamada de transformação *ortogonal* e, uma vez que os vetores no produto escalar são os vetores unitários do sistema de coordenadas, esta transformação também é chamada transformação *ortonormal*.

Se o sistema de coordenadas $OUVW$ é rotacionado em um ângulo qualquer ao redor de um dos eixos principais do sistema $OXYZ$, então o ponto \mathbf{p}_{uvw} , tendo como coordenadas $(p_u, p_v, p_w)^T$ em relação ao sistema $OUVW$, terá diferentes coordenadas (p_x, p_y, p_z) em relação ao sistema de referência $OXYZ$.

Dada a equação A.3, as matrizes de rotação que transformam um ponto \mathbf{p}_{uvw} em um ponto \mathbf{p}_{xyz} podem ser facilmente calculadas:

$$\mathbf{R}_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad (\text{A.6})$$

$$\mathbf{R}_{y,\phi} = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix} \quad (\text{A.7})$$

$$\mathbf{R}_{z,\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A.8})$$

As matrizes $\mathbf{R}_{x,\alpha}$, $\mathbf{R}_{y,\phi}$ e $\mathbf{R}_{z,\theta}$ são chamadas *matrizes de rotação básicas*. Outras matrizes de rotação podem ser obtidas a partir destas.

Matrizes de rotação básicas podem ser multiplicadas conjuntamente para representar uma seqüência finita de rotações ao redor dos principais eixos de rotação do sistema de coordenadas $OXYZ$. Em adição à rotação ao redor dos eixos principais do quadro de referência, o sistema de coordenadas $OUVW$ pode mover-se em relação aos seus próprios eixos. Neste caso, a matriz resultante (matriz de composição) pode ser obtida através das seguintes regras:

- Inicialmente ambos os sistemas de coordenadas são coincidentes, assim a matriz de rotação é a matriz identidade \mathbf{I} .
- Se o sistema de coordenadas $OUVW$ está sofrendo uma rotação em relação a um dos eixos do sistema de coordenadas de referência $OXYZ$, então multiplica-se a matriz de rotação básica apropriada pela matriz resultante das rotações anteriores.
- Se o sistema de coordenadas $OUVW$ está sendo rotacionado em relação a um dos seus próprios eixos, então multiplica-se a matriz resultante das rotações anteriores pela matriz de rotação básica apropriada.

A.1.1 Interpretação Geométrica

A interpretação geométrica das matrizes básicas de rotação costuma ser bastante útil. Seja \mathbf{p} um ponto fixo do sistema de coordenadas $OUVW$ com valor $\mathbf{p} = (1, 0, 0)^T$, isto é, $\mathbf{p}_{uvw} \equiv \mathbf{i}_u$. Então, a primeira coluna da matriz de rotação representa as coordenadas deste ponto em relação ao sistema de coordenadas $OXYZ$. Do mesmo modo, escolhendo $\mathbf{p} = (0, 1, 0)^T$ ou $\mathbf{p} = (0, 0, 1)^T$, pode-se identificar os elementos da segunda e terceira colunas da matriz de rotação como sendo a representação dos eixos OV e OW , respectivamente, do sistema de coordenadas $OUVW$ em relação ao sistema de coordenadas $OXYZ$.

Assim, dados um sistema de referência $OXYZ$ e uma matriz de rotação, os vetores-coluna da matriz de rotação representam os eixos principais do sistema de coordenadas $OUVW$ em relação ao sistema de referência. Em outras palavras, uma matriz de rotação representa geometricamente os eixos principais do sistema de coordenadas rotacionado em relação ao sistema de coordenadas de referência.

Uma vez que o inverso de uma matriz de rotação é equivalente à sua transposta, os vetores-linha da matriz de rotação representam os eixos principais do sistema de referência $OXYZ$ em relação ao sistema de coordenadas $OUVW$. Algumas das propriedades das matrizes de rotação são listadas a seguir:

- Cada vetor-coluna de uma matriz de rotação é a representação do vetor unitário do eixo rotacionado expresso em termos dos vetores unitários dos eixos do

sistema de coordenadas de referência, e cada vetor-linha é a representação do vetor unitário do eixo de referência em termos dos vetores unitários dos eixos do sistema rotacionado.

- Como cada linha e coluna é a representação de um vetor unitário, a magnitude de cada linha e de cada coluna deve ser igual a 1. Esta é uma propriedade de sistemas de coordenadas ortonormais. O determinante de uma matriz de rotação é +1 para um sistema de coordenadas direito e -1 para um sistema de coordenadas esquerdo.
- Como cada linha é a representação vetorial de vetores ortonormais, o produto interno de cada linha por qualquer outra linha é igual a zero. Similarmente, o produto interno de quaisquer duas colunas também é igual a zero.
- O inverso de uma matriz de rotação é a sua transposta. A multiplicação de uma matriz de rotação por sua transposta é a matriz identidade.

A.2 Matrizes de Transformação Homogêneas

Uma vez que uma matriz de rotação 3×3 não provê elementos para representar a translação de um sistema de coordenadas em relação a outro, uma quarta coordenada ou componente deve ser introduzida no vetor de posição $\mathbf{p} = (p_x, p_y, p_z)^T$ no espaço tridimensional, o que torna $\mathbf{p} = (wp_x, wp_y, wp_z, w)^T$. Diz-se, então, que o vetor \mathbf{p} está expresso em *coordenadas homogêneas*.

O conceito de uma representação de pontos em coordenadas homogêneas num espaço tridimensional euclidiano é útil no desenvolvimento de transformações de referenciais por meio de matrizes que incluam rotação, translação, transformação de perspectiva e escala. Em geral, a representação de um vetor de posição de N componentes por um vetor de $N+1$ componentes é chamada *representação em coordenadas homogêneas*.

Numa representação em coordenadas homogêneas, a transformação de um vetor de N dimensões é realizada no espaço de $N+1$ dimensões e o vetor físico de N dimensões é obtido dividindo-se as coordenadas homogêneas pela $(N+1)$ -ésima coordenada w . Assim, no espaço de três dimensões, o vetor de posição $\mathbf{p} = (p_x, p_y, p_z)^T$ é representado pelo vetor expandido $(wp_x, wp_y, wp_z, w)^T$ na representação por coordenadas homogêneas. As coordenadas físicas estão relacionadas com as coordenadas homogêneas da seguinte maneira:

$$p_x = \frac{wp_x}{w} \quad p_y = \frac{wp_y}{w} \quad p_z = \frac{wp_z}{w}$$

Não existe uma única representação homogênea para um vetor \mathbf{p} . Desta maneira, pode-se ver a quarta componente do vetor como um fator de escala. Se esta coordenada for a unidade ($w = 1$), então as coordenadas homogêneas de um vetor de posição são idênticas às coordenadas físicas deste vetor.

Uma *matriz homogênea de transformação* é, então, uma matriz 4×4 que produz o mapeamento do vetor de posição expresso em coordenadas homogêneas de um sistema de coordenadas para outro sistema de coordenadas. Uma matriz homogênea de transformação consiste em quatro sub-matrizes:

$$\mathbf{T} = \left[\begin{array}{c|c} \mathbf{R}_{3 \times 3} & \mathbf{p}_{3 \times 1} \\ \hline \mathbf{f}_{1 \times 3} & \mathbf{d}_{1 \times 1} \end{array} \right] = \left[\begin{array}{c|c} \text{matriz de rotação} & \text{vetor de posição} \\ \hline \text{perspectiva} & \text{escala} \end{array} \right] \quad (\text{A.9})$$

A submatriz 3×3 superior esquerda representa a matriz de rotação; a submatriz superior direita 3×1 representa o vetor de posição da origem do sistema de coordenadas rotacionado com relação ao sistema de referência; a submatriz 1×3 inferior esquerda representa transformação de perspectiva e o quarto elemento diagonal é o fator de escala global. Estes dois últimos elementos são normalmente utilizados apenas em sistemas que realizam visualização gráfica.

A matriz homogênea de transformação contém a relação geométrica entre o sistema de coordenadas modificado e o sistema de coordenadas de referência $OXYZ$.

Se um vetor de posição \mathbf{p} é expresso em coordenadas homogêneas, então, usando o conceito de matriz de transformação, uma matriz 3×3 pode ser estendida a uma matriz 4×4 \mathbf{T}_{rot} para operações puras de rotação. Deste modo, as equações A.6, A.7 e A.8 tornam-se

$$\mathbf{R}_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{y,\phi} = \begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{z,\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Estas matrizes 4×4 de rotação são chamadas *matrizes homogêneas básicas de rotação*.

A submatriz 3×1 superior direita da matriz homogênea de transformação produz o efeito de transladar o sistema de coordenadas $OUVW$ que possui os eixos paralelos ao sistema de coordenadas de referência $OXYZ$ mas cuja origem encontra-se a uma distância (dx, dy, dz) de $OXYZ$:

$$\mathbf{T}_{tran} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Esta matriz de transformação é chamada *matriz homogênea básica de translação*.

A.2.1 Interpretação Geométrica

Seja um ponto \mathbf{p} fixo no sistema de coordenadas $OUVW$ e representado em coordenadas homogêneas como $(0, 0, 0, 1)^T$, isto é, \mathbf{p} é a origem do sistema de coordenadas $OUVW$. Então a matriz 3×1 superior direita representa a posição da origem do sistema de coordenadas $OUVW$ em relação ao sistema de coordenadas de referência $OXYZ$. Seja agora o ponto \mathbf{p} escolhido como sendo $(1, 0, 0, 1)^T$, isto é, $\mathbf{p}_{uvw} \equiv \mathbf{i}_u$. Ainda, assumamos que a origem de ambos os sistemas de coordenadas coincidem na origem O . Isto tem o efeito de tornar os elementos da submatriz 3×1 superior direita um vetor nulo. Então, a primeira coluna da matriz homogênea de transformação representa as coordenadas do eixo OU do sistema $OUVW$ com respeito à $OXYZ$. Similarmente, escolhendo-se \mathbf{p} como $(0, 1, 0, 1)^T$ ou $(0, 0, 1, 1)^T$, pode-se identificar que os elementos da segunda e terceira colunas da matriz de transformação são os eixos OV e OW respectivamente, do sistema de coordenadas $OUVW$ com relação ao sistema de coordenadas de referência. Assim, dados um sistema de referência $OXYZ$ e uma matriz homogênea de transformação \mathbf{T} , os vetores-coluna da submatriz de rotação representam os eixos principais do sistema $OUVW$ com relação ao sistema $OXYZ$. O quarto vetor-coluna de \mathbf{T} representa a posição da origem do sistema $OUVW$ em relação à origem do sistema $OXYZ$. Em outras palavras, uma matriz homogênea de transformação representa a *localização* de um sistema de coordenadas rotacionado (posição e orientação) em relação a um sistema de coordenadas de referência.

Matrizes homogêneas de rotação e translação podem ser multiplicadas conjuntamente para obter uma matriz que representa uma série finita de transformações de referenciais. Como a multiplicação de matrizes não é uma operação comutativa, as seguintes regras servem para a obtenção da matriz homogênea de transformação composta corretamente:

- Inicialmente, ambos os sistemas de coordenadas são coincidentes, assim a matriz homogênea de transformação inicial é a matriz identidade \mathbf{I} .
- Se o sistema de coordenadas $OUVW$ está sendo rotacionado/transladado em relação a um dos eixos do sistema de coordenadas de referência $OXYZ$, então multiplica-se a matriz de transformação adequada pela matriz de transformação anterior.
- Se o sistema de coordenadas $OUVW$ está sendo rotacionado em relação a um dos seus próprios eixos, então multiplica-se a matriz de transformação anterior pela matriz de transformação apropriada.

Anexo B

Matrizes de Derivação

Devido à natureza das equações utilizadas neste trabalho, é freqüentemente necessário o cálculo de derivadas de elementos resultantes da multiplicação de matrizes. Para tanto, foi desenvolvido um método sistemático para solução deste problema, aqui introduzido formalmente conforme o seguinte teorema:

Teorema 1 *Sejam \mathbf{F} e \mathbf{G} matrizes de funções reais de variável real, e sejam estas variáveis distintas. Seja \mathbf{F}' a matriz cujos elementos são as derivadas dos elementos de \mathbf{F} . Então, as derivadas dos elementos da matriz $\mathbf{M} = \mathbf{F} \mathbf{G}$ em relação à variável das funções de \mathbf{F} são os elementos da matriz $\mathbf{M}' = \mathbf{F}' \mathbf{G}$.*

Demonstração:

Sejam \mathbf{F} e \mathbf{G} matrizes de funções reais de variável real, com $F : \mathfrak{R} \rightarrow \mathfrak{R}$ e $G : \mathfrak{R} \rightarrow \mathfrak{R}$, ou seja,

$$\mathbf{F} = \begin{bmatrix} f_{11}(x) & f_{12}(x) & \dots & f_{1n}(x) \\ f_{21}(x) & f_{22}(x) & \dots & f_{2n}(x) \\ \dots & \dots & \dots & \dots \\ f_{m1}(x) & f_{m2}(x) & \dots & f_{mn}(x) \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} g_{11}(y) & g_{12}(y) & \dots & g_{1p}(y) \\ g_{21}(y) & g_{22}(y) & \dots & g_{2p}(y) \\ \dots & \dots & \dots & \dots \\ g_{n1}(y) & g_{n2}(y) & \dots & g_{np}(y) \end{bmatrix}$$

e sejam ainda as variáveis x e y distintas.

Sejam \mathbf{F}' e \mathbf{G}' chamadas de *matrizes de derivação* e sejam estas matrizes tais que cada elemento de \mathbf{F}' e \mathbf{G}' é dado por:

$$\mathbf{F}'_{ij} = \frac{d\mathbf{F}_{ij}}{dx} \quad i = 1, \dots, m \quad j = 1, \dots, n$$

$$\mathbf{G}'_{ij} = \frac{d\mathbf{G}_{ij}}{dy} \quad i = 1, \dots, n \quad j = 1, \dots, p$$

ou ainda,

$$\mathbf{F}' = \begin{bmatrix} f'_{11}(x) & f'_{12}(x) & \cdots & f'_{1n}(x) \\ f'_{21}(x) & f'_{22}(x) & \cdots & f'_{2n}(x) \\ \cdots & \cdots & \cdots & \cdots \\ f'_{m1}(x) & f'_{m2}(x) & \cdots & f'_{mn}(x) \end{bmatrix}$$

$$\mathbf{G}' = \begin{bmatrix} g'_{11}(y) & g'_{12}(y) & \cdots & g'_{1p}(y) \\ g'_{21}(y) & g'_{22}(y) & \cdots & g'_{2p}(y) \\ \cdots & \cdots & \cdots & \cdots \\ g'_{n1}(y) & g'_{n2}(y) & \cdots & g'_{np}(y) \end{bmatrix}$$

Seja \mathbf{M} a matriz resultante da multiplicação de \mathbf{F} e \mathbf{G} . Cada elemento da matriz \mathbf{M} pode ser escrito como

$$\mathbf{M}_{ij} = \sum_{k=1}^n f_{ik}(x)g_{kj}(y)$$

onde $i = 1, \dots, m$ e $j = 1, \dots, p$.

A derivada em relação a x^1 de cada elemento da matriz \mathbf{M} é dada por

$$\frac{d\mathbf{M}_{ij}}{dx} = \frac{d}{dx} \sum_{k=1}^n f_{ik}(x)g_{kj}(y)$$

e, uma vez que a derivada de um somatório equivale ao somatório das derivadas, temos que

$$\frac{d\mathbf{M}_{ij}}{dx} = \sum_{k=1}^n \frac{d}{dx} (f_{ik}(x)g_{kj}(y))$$

Como as variáveis x e y são distintas e as funções da matriz \mathbf{G} são independentes de x , temos que

$$\frac{d\mathbf{M}_{ij}}{dx} = \sum_{k=1}^n f'_{ik}(x)g_{kj}(y)$$

Mas os elementos $f'_{ik}(x)$, onde $i = 1, \dots, m$ e $k = 1, \dots, n$, são, por definição, os elementos da matriz de derivação \mathbf{F}' . Sendo assim, as derivadas dos componentes da matriz resultante da multiplicação das matrizes \mathbf{F} e \mathbf{G} são correspondentes aos elementos de uma matriz resultante da multiplicação de \mathbf{F}' por \mathbf{G} .

B.1 Aplicação ao Problema Tratado

Conhecendo-se as características do problema a ser tratado, pode-se afirmar que

- As derivadas das variáveis de posição e de orientação espacial referentes ao sistema de coordenadas de referência $OXYZ$ são obtidas através da derivada de cada elemento da matriz resultante da multiplicação das matrizes de transformação associadas ao braço mecânico tratado.

¹Sem perda de generalidade, considere-se a variável x . A variável y também pode ser considerada com resultados idênticos

- Cada uma destas matrizes de transformação possui termos em uma única variável θ_k , $k = 1, \dots, n$, onde n é o número de graus de liberdade do robô.

Dadas estas condições, pode-se então aplicar o teorema 1, resultando o seguinte:

- Cada matriz de transformação possui uma *matriz de derivação* associada que corresponde à matriz de transformação com cada um de seus termos derivados.
- Para encontrarmos as derivadas das variáveis de posição e de orientação referente ao sistema de coordenadas de referência $OXYZ$ em relação a uma variável de junta θ_k , basta procedermos à seguinte multiplicação:

$$M_{d_k} = M_1 M_2 \dots D_k \dots M_{N-1} M_N$$

onde M_{d_k} é a matriz que possui as derivadas resultantes em relação à junta k , e M_i , $i = 1, \dots, k-1, k+1, \dots, N$ são as matrizes de transformação correspondentes às juntas do braço. A matriz D_k representa a matriz de derivação correspondente à variável de derivação k .

Desta forma, sabendo-se a estrutura de uma matriz de transformação (ver Apêndice A), tem-se, a cada instante, as derivadas de cada variável de posição e orientação do braço em relação a qualquer junta do robô.

Anexo C

Fundamentos de Robótica

C.1 Geometria

Um manipulador ou braço robótico mecânico pode ser modelado como uma cadeia articulada aberta composta de diversos segmentos rígidos conectados em série por juntas prismáticas ou por juntas de revolução, acionadas por atuadores. Uma extremidade da cadeia é ligada a uma base de suporte enquanto a outra é livre e ligada a uma ferramenta (a *garra*) que serve para manipulação de objetos e para a execução de tarefas de montagem. O movimento relativo das juntas resulta no movimento dos segmentos que posicionam a garra numa determinada posição com uma determinada orientação. Na maioria das aplicações robóticas, o interesse maior é a descrição espacial da extremidade livre do manipulador em relação a um sistema de coordenadas de referência fixo.

C.2 Cinemática

A cinemática de um braço manipulador trata do estudo analítico da geometria do movimento de um braço em relação a um sistema fixo de referência como função do tempo, sem levar em conta as forças/momentos que causam o movimento. Desta maneira, a cinemática trata da descrição analítica do deslocamento espacial do robô como função do tempo, em particular as relações entre o espaço das variáveis das juntas e a posição da extremidade livre do braço.

Para um dado braço manipulador, dados o vetor correspondente ao ângulo das juntas $q(t) = (q_1(t), q_2(t), \dots, q_n(t))^T$ onde n é o número de graus de liberdade do braço e os parâmetros geométricos dos segmentos do braço, a posição e a orientação da garra no espaço podem ser calculados diretamente. Este problema é conhecido como *problema cinemático direto*. Por outro lado, dadas a posição e orientação desejadas, pode-se encontrar o vetor dos ângulos das juntas correspondentes. Este problema é conhecido como *problema cinemático inverso*. A relação entre os problemas cinemáticos direto e inverso é mostrada na figura C.1.

As variáveis independentes num braço robótico são as variáveis das juntas e, uma vez que uma tarefa é usualmente descrita em termos do sistema de coordenadas de referência, o problema cinemático inverso é usado com mais frequência.

O modelo cinemático de um braço robótico é freqüentemente construído a partir de matrizes de transformação homogêneas, formalmente descritas no Apêndice A. Através das matrizes de transformação, pode-se calcular a posição e a orientação da garra no espaço (problema cinemático direto), através de uma simples

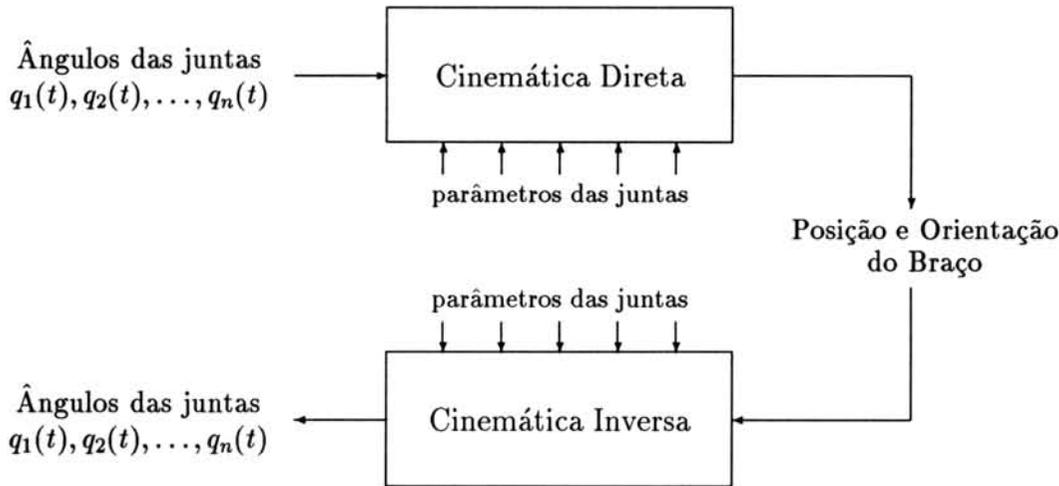


FIGURA C.1 — Relação entre os problemas cinemáticos inverso e direto

multiplicação destas matrizes.

Dado um braço manipulador, associado a cada segmento existe uma matriz que transforma um ponto do referencial fixo associado à extremidade deste segmento num ponto do referencial associado ao segmento anterior. Esta matriz é construída de acordo com as características geométricas do segmento e com o tipo de junta que une os dois segmentos (prismática ou rotacional). Recursivamente, temos um ponto do referencial da garra transformado num ponto do sistema fixo de referência. Num braço com N segmentos, o ponto do sistema fixo de referência é calculado da seguinte maneira:

$$\mathbf{p}_{ref} = (M_1 M_2 \dots M_N) \mathbf{p}_N$$

onde M_n , $n = 1, \dots, N$, é a matriz que transforma o referencial n no referencial $n - 1$ e \mathbf{p}_N é um ponto (normalmente a origem) do referencial N (ou referencial da garra).

C.3 Dinâmica

A dinâmica de um manipulador robótico diz respeito à formulação matemática das equações de movimento do manipulador. As equações dinâmicas do movimento de um manipulador são um conjunto de equações matemáticas que descrevem o seu comportamento dinâmico, levando em consideração todas as forças que atuam sobre ele. Estas equações de movimento são úteis para simulação computacional do movimento de um braço, para o projeto de sistemas de controle convenientes e para a avaliação do projeto cinemático e estrutural do robô.

O objetivo do controle dinâmico de um manipulador mecânico é manter a resposta dinâmica deste manipulador em conformidade com alguns pré-requisitos desejados sobre o desempenho do sistema e metas a serem atingidas. Geralmente, o desempenho dinâmico de um manipulador depende diretamente da eficiência dos algoritmos de controle e do seu modelo dinâmico. O problema do controle consiste, então, em obter modelos dinâmicos do sistema físico do braço e então formular as especificações das leis de controle ou estratégias correspondentes de modo a alcançar o desempenho e resposta esperados do sistema.

Anexo D

GSE

O *GSE* ou *Graphic Simulation Environment* foi o ambiente de simulação desenvolvido neste trabalho. Ele é composto de diversos módulos, independentes entre si, que tratam separadamente de aspectos da simulação propriamente dita. A figura D.1 esquematiza estes módulos e as relações existentes entre eles. Todos os módulos estão disponíveis para o utilizador final, seguindo-se uma breve descrição dos módulos do ambiente de simulação gráfica e de sua funcionalidade.

D.1 Window Manager

O módulo *Window Manager* é construído sobre a funcionalidade do *X Window System*. É um módulo que tem como função oferecer uma interface fácil de utilizar na criação e manipulação de janelas. A principal razão pela qual este módulo é necessário é que as primitivas gráficas do *X Window System* não são muito fáceis de lidar, sendo necessário ter-se um profundo conhecimento da estrutura e funcionamento do sistema mesmo para se escrever um programa relativamente simples.

Este módulo fornece a classe *win*, cujas funções associadas permitem criar, escrever, desenhar, trocar as cores, e efetuar uma série de outras operações sobre janelas de um modo totalmente transparente ao sistema subjacente.

Juntamente com a classe *win*, este módulo fornece a classe *colormap*, referente ao tratamento de cores do programa. Esta classe foi construída devido à dificuldade do tratamento de cores do *X Window System*. Os *colormaps*, ou mapas de cores, podem ser privados ou partilhados e esta escolha depende unicamente do utilizador e dos propósitos da aplicação em desenvolvimento.

Cores são recursos limitados, e, apesar de o número de cores permitidas numa *workstation* ser grande, existe um número limitado de cores que podem realmente ser usadas ao mesmo tempo. A vantagem das cores partilhadas é a sobreposição no tempo do uso dessas cores por diferentes aplicações. Existem uma série de cores (branco, preto, vermelho, azul, etc.) que são freqüentemente usadas quando um processo precisa lidar com cores. Uma vez que cores são recursos limitados, e que raramente as aplicações necessitam mudar os valores (tonalidades) das cores usadas, não há necessidade da alocação de uma série de células azuis, vermelhas, etc., quando apenas uma de cada seria necessária. Cores privadas, por outro lado, permitem ao utilizador alterar dinamicamente os valores das cores armazenadas no mapa de cores. Esta estratégia é útil quando a aplicação necessita, por exemplo, que grupos de *pixels* tenham uma relação numérica entre eles. Isso é bastante útil no desenho de sombras e objetos de três dimensões. A desvantagem desta abordagem é que uma aplicação

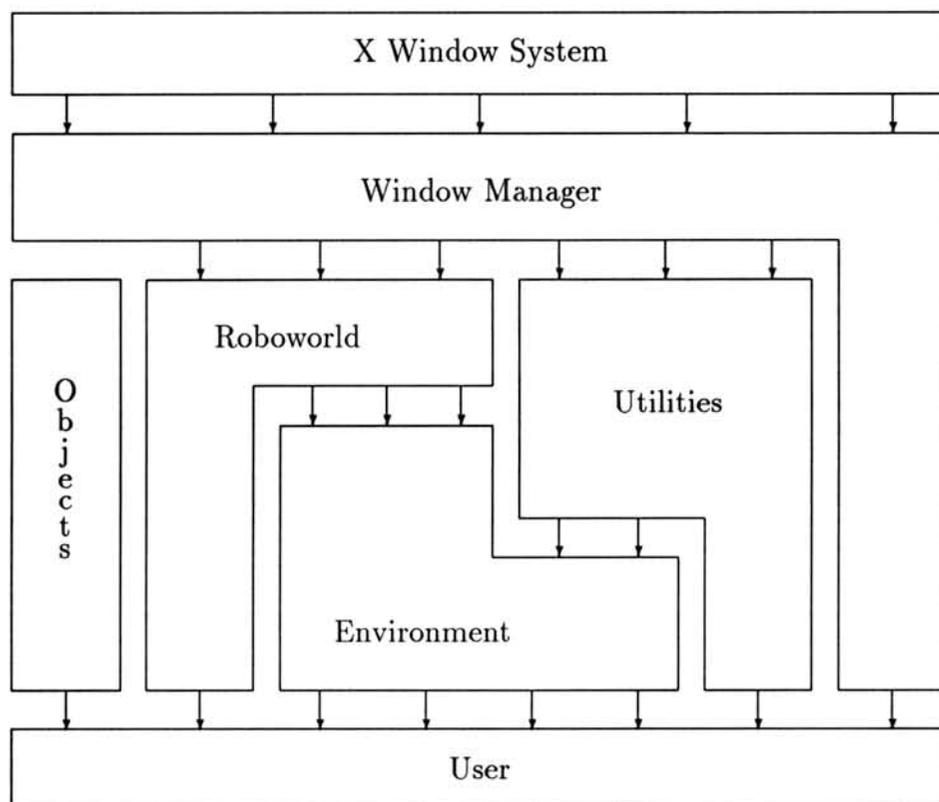


FIGURA D.1 — Interfaces do ambiente gráfico de simulação *GSE*

que utilize cores não compartilhadas pode facilmente monopolizar os recursos de cores de uma estação de trabalho.

Os mapas de cores fornecidos pela classe *colormap* podem ou não ser utilizados juntamente com a classe *win*, que possui um mapa como membro. Apesar desta facilidade, recomenda-se sempre o uso de cores através de uma janela, pois a maioria das estações de trabalho não conseguem manter mais de um mapa aberto ao mesmo tempo, e a classe *win* faz uso do mapa de cores da própria estação. A tentativa de usar mais de um mapa ao mesmo tempo pode gerar efeitos colaterais não desejados, o que não acontece com o uso da classe *win*.

D.2 Utilities

O módulo *Utilities* contém diversas facilidades para construção de programas com interface gráfica. Estas facilidades estão encapsuladas em classes que possuem uma interface de construção e manipulação extremamente simples. Uma vez que as classes pertencentes a esta biblioteca são genéricas, elas podem ser utilizadas em qualquer programa de aplicação.

As classes contidas nesta biblioteca provêm facilidades na manipulação de arquivos de configuração, temporizadores, arquivos de figuras, histórico de comandos e elementos pertencentes a janelas gráficas, tais como botões, seletores, menus e leitores de caracteres interativos.

D.3 Objects

Quando tratam-se objetos no espaço tridimensional, freqüentemente incorre-se no mesmo problema anteriormente mencionado sobre sistemas de simulação, ou seja, muito esforço para a construção de primitivas de manipulação de objetos e nenhuma preocupação na reutilização de código já escrito.

A biblioteca de objetos fornece objetos tais como pontos, vetores, linhas, triângulos, retângulos, cubos e planos tridimensionais. Estes objetos podem ser criados, destruídos ou modificados diretamente pelo utilizador. Primitivas de manipulação de objetos, tais como rotação, translação, intersecção, soma, subtração e escala também são fornecidas por este módulo.

D.4 Environment

O módulo *Environment* é o módulo mais direcionado para aspectos de controle e simulação. Existem duas classes principais nesta biblioteca, as classes *environment* e *RobotArm*. A classe *environment* consiste em um ambiente completo de simulação, com visualização completa do ambiente, câmaras móveis, gravação de filmes, fotografias, editoração de objetos e uma série de outras capacidades, tudo com uma interface gráfica simples de usar. A classe *RobotArm* foi construída tendo como objetivo a possibilidade de uso de um braço genérico definido pelo utilizador no ambiente de visualização definido pela classe *environment*. Esta classe permite ao utilizador definir um braço mecânico genérico, seja através de um arquivo ou do próprio construtor da classe.

D.5 RoboWorld

RoboWorld [SUD 94] é um ambiente de simulação com visualização gráfica de objetos tridimensionais. Este ambiente fornece a classe *World* que mantém um número de entidades ou objetos. Para cada instância desta classe é possível a criação de um número arbitrário de objetos¹. Estes objetos, além de sua posição e orientação no espaço tridimensional, possuem ainda um certo número de atributos – descrevendo cor, visibilidade, tamanho, etc. – que podem ser acessados e modificados. A cinemática de um objeto é realizada através de operações de rotação e translação. Os objetos existentes podem ser agrupados de modo a formar outros objetos mais complexos. Operações sobre um grupo de objetos afetam todos os elementos do grupo como se eles formassem apenas um objeto. Grupos de objetos podem ser, eles mesmos, parte de um outro grupo de objetos.

Além de objetos simples, RoboWorld fornece ainda uma classe especial de objetos, chamados *robôs*. Estes objetos representam um braço robótico composto de segmentos ligados por juntas rotacionais. Cada braço pode possuir ainda uma garra localizada num de seus extremos. Estes objetos eram fixos e de dois tipos diferentes, que eram usados para simulação do sistema JANUS. Ao abrigo deste trabalho foram realizadas extensões no sistema de modo a possibilitar a definição de um braço arbitrário pelo utilizador, tornando o sistema verdadeiramente genérico.

¹Objetos geométricos tridimensionais disponíveis no ambiente

Bibliografia

- [ATK 90] ATKESON, Christopher G. & REINKENSMEYER, David J., *Using Associative Content-Addressable Memories to Control Robots*. In: *Neural Networks for Control*. Editado por: W. Thomas Miller III, Richard S. Sutton e Paul J. Werbos, MIT Press, 1990.
- [BAR 92] BARTO, Andrew G., *Reinforcement Learning and Adaptive Critic Methods*. In: *Handbook of Intelligent Control – Neural, Fuzzy and Adaptive Approaches*, capítulo 12, 1992.
- [BAR 90] BARTO, Andrew G., *Connectionist Learning for Control*. In: *Neural Networks for Control*, Editado por: W. Thomas Miller III, Richard S. Sutton e Paul J. Werbos, MIT Press, 1990.
- [BEE 90] BEER, Randall D., *Intelligence as Adaptive Behavior*. Academic Press Inc., 1990.
- [BRO 91] BROOKS, Rodney A., *Intelligence Without Reason*. Relatório Técnico, Artificial Intelligence Laboratory, MIT, 1991.
- [COR 94] CORDO, Paul & HARNAD, Stevan (editores), *Movement Control*. Cambridge University Press, 1994.
- [ENG 96] ENGEL, Paulo M., *Attentional Mode Neural Network: A New Approach for Real-Time Self-Learning*. *IEEE International Symposium on Circuits and Systems*, Atlanta, Georgia, May 12-15, 1996.
- [FER 86] FERRATÉ, G. et al., *Robotica Industrial*. Marcombo Editores, Barcelona, 1986.
- [FEA 96] FERREIRA, Ana Paula L. & ENGEL, Paulo M., *Positioning a Robot Arm: an Adaptive Neural Approach*. to appear 1996 *International Workshop on Neural Networks for Identification, Control, Robotics and Signal/Image Processing*, Veneza, Itália, Agosto 1996.
- [FEA 95] FERREIRA, Ana Paula L., *GSE – User’s Guide*. Relatório Técnico, GMD, Alemanha, 1995.
- [FEA 94] FERREIRA, Ana Paula L., *Controle Inteligente de um Braço de Robô*. Trabalho Individual TI-461, UFRGS, 1994.
- [FRA 90] FRANKLIN, Judy A. & SELFRIDGE, Oliver G., *Some New Direction for Adaptive Control Theory in Robotics*. In: *Neural Networks for Control*, Editado por: W. Thomas Miller III, Richard S. Sutton e Paul J. Werbos MIT Press 1990

- [FRE 91] FREEMAN, James A. & SKAPURA, David M., *Neural Networks — Algorithms, Applications and Programming Techniques*. Addison-Wesley Publishing Company, 1991.
- [FUU 87] FU, K.S., GONZALEZ, R.C. e LEE, C.S.G., *Robotics — Control, Sensing, Vision and Intelligence*. McGraw-Hill International, 1987.
- [HOU 90] HOUK, James C. *et al*, *An Adaptive Sensorimotor Network Inspired by the Anatomy and Physiology of the Cerebellum*. In: *Neural Networks for Control*, Editado por: W. Thomas Miller III, Richard S. Sutton e Paul J. Werbos, MIT Press, 1990.
- [KAF 84] KAFRISSEN, Edward & STEPHANS, Mark, *Industrial Robots and Robotics*. Reston Publishing Company, Inc., USA, 1984.
- [KAW 90] KAWATO, Mitsuo, *Computational Schemes and Neural Network Models for Formation and Control of Multijoint Arm Trajectory*. In: *Neural Networks for Control*, Editado por: W. Thomas Miller III, Richard S. Sutton e Paul J. Werbos, MIT Press, 1990.
- [LEW 95] LEWIS, Frank L., LIU, Kai e YESILDIREK, Aydin, *Neural Net Robot Controller with Guaranteed Tracking Performance*. *IEEE Transactions on Neural Networks*, Vol. 6, no. 3, 1995.
- [KEL 94] KELBER, Ulrich, ŚMIEJA, Frank e BEYER, Uwe, *Cooperation between Two Simulated Robot Arms using a Reflective Team Architecture*. Relatório Técnico, German National Research Center for Computer Science (GMD), Alemanha, 1994.
- [MAT 94] MATARIC, Maja J., *Interaction and Intelligent Behavior*. Tese de Doutorado, MIT, 1994.
- [MAT 92] MATARIC, Maja J., *Behavior-Based Systems: Key Properties and Implications*. *IEEE International Conference on Robotics and Automation*, Workshop on Architectures for Intelligent Control Systems, pp. 46-54, 1992.
- [MEL 90] MEL, Bartlett W., *Vision-Based Robot Motion Planning*. In: *Neural Networks for Control*, Editado por: W. Thomas Miller III, Richard S. Sutton e Paul J. Werbos, MIT Press, 1990.
- [MIN 86] MINSKY, Marvin, *The Society of Mind*. Simon and Schuster, 1986.
- [MUE 94] MUEHLENBEIN, Heinz, *Open Worlds, Reflective Statistics and Stochastic Modelling*. Relatório Técnico, German National Research Center for Computer Science (GMD), Alemanha, 1994.
- [NAR 90] NARENDRA, Kumpati S., *Adaptive Control Using Neural Networks*. In: *Neural Networks for Control*, Editado por: W. Thomas Miller III, Richard S. Sutton e Paul J. Werbos, MIT Press, 1990.

- [NGU 90] NGUYEN, Derrick & WIDROW, Bernard, *The Truck Backer-Upper: An Example of Self-Learning in Neural Networks*. In: *Neural Networks for Control*, Editado por: W. Thomas Miller III, Richard S. Sutton e Paul J. Werbos, MIT Press, 1990.
- [RIT 92] RITTER, H., MARTINETZ, T. e SCHULTEN, K., *Neural Computation and Self-Organized Maps — An Introduction*. Addison-Wesley Publishing Company, 1992.
- [SHA 94] SHADMEHR, Reza & MUSSA-IVALDI, Ferdinando A., *Geometric Structure of the Adaptive Controller of the Human Arm*. AI Memo 1437, Department of Brain and Cognitive Sciences, MIT, 1994.
- [SMI 94a] ŚMIEJA, Frank & BEYER, Uwe, *JANUS: A Robot Manipulator System Implemented on a Blackboard Architecture*. Relatório Técnico, German National Research Center for Computer Science (GMD), Alemanha, 1994.
- [SMI 94b] ŚMIEJA, Frank & BEYER, Uwe, *JANUS: A Society of Agents*. Relatório Técnico, German National Research Center for Computer Science (GMD), Alemanha, 1994.
- [SMI 94c] ŚMIEJA, Frank, BEYER, Uwe e RICHTER, Gernot, *Adaptive Control of a Robot Arm Using Driver Programs*. Relatório Técnico, German National Research Center for Computer Science (GMD), Alemanha, 1994.
- [SMI 92] ŚMIEJA, F.J. & MUHLENBEIN, H., *Reflective Modular Neural Network Systems*. Relatório Técnico, German National Research Center for Computer Science (GMD), Alemanha, 1992.
- [SMI 90] ŚMIEJA, F.J. & MUHLENBEIN, H., *The JANUS Architecture for a Robot Brain*. Relatório Técnico, German National Research Center for Computer Science (GMD), Alemanha, 1990.
- [SON 94] SONTAG, Eduardo D., *Some Topics in Neural Networks and Control*. Relatório Técnico, Department of Mathematics, Rutgers University, 1994.
- [SPI 77] SPIEGEL, Murray R., *Probabilidade e Estatística*. Coleção Schaum, Ed. McGraw-Hill do Brasil, 1977.
- [STA 87] J., Staugaard & ANDREW, C., *Robotics and AI — An Introduction to Applied Machine Intelligence*. Prentice Hall Inc., USA, 1987.
- [SUD 94] SUDBRAK, Thomas, BEYER, Uwe e ŚMIEJA, Frank J., *RoboWorld — A 3-dimensional Simulation Environment*. Relatório Técnico, German National Research Center for Computer Science (GMD), Alemanha, 1994.
- [VAL 92] VALAVANIS, Kimon P. & SARIDIS, George N., *Intelligent Robotic Systems: Theory, Design and Applications*. Kluwer Academic Publishers, USA 1992.

- [WER 90] WERBOS, Paul J., *Overview of Designs and Capabilities*. In: *Neural Networks for Control*, Editado por: W. Thomas Miller III, Richard S. Sutton e Paul J. Werbos, MIT Press, 1990.