UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Controle de Força e Posição de Robôs Manipuladores Utilizando Redes Neurais Artificiais

Dissertação submetida à Universidade Federal de Santa Catarina
como requisito parcial à obtenção do grau de

Mestre em Engenharia Elétrica

por

Sandro Battistella

Florianópolis, abril de 1999

Controle de Força e Posição de Robôs Manipuladores Utilizando Redes Neurais Artificiais

Sandro Battistella

Esta dissertação foi julgada adequada para a obtenção do título de **Mestre em Engenharia** na especialidade **Engenharia Elétrica**, área de concentração **Controle, Automação e Informática Industrial**, e aprovada em sua forma final pelo curso de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.

Florianópolis, 05 de abril de 1999.

Prof. Marcelo Ricardo Stemmer, Dr.
Orientador

Prof. Edson Roberto de Pieri, Dr.
Co-orientador

Prof. Ildemar Cassana Decker, D.Sc.
Coordenador do Curso de Pós-Graduação em Engenharia Elétrica

Banca Examinadora

Prof. Marcelo Ricardo Stemmer, Dr.
Orientador, UFSC - DAS

Prof. Edson Roberto de Pieri, Dr.
Co-orientador, UFSC - DAS

Prof. Werner Kraus Jr., Ph.D.
UFSC - DAS

Prof. Raul Guenther, D.Sc.
UFSC - EMC

Prof. Jorge Muniz Barreto, Dr.
UFSC - INF

Importa mais como se viveu do que quanto.
Viver bem não é viver muito e sim para além do tempo concedido, o que somente se obtém vivendo para o bem.
(Anônimo)

Agradecimentos

Aos meus pais, Ary e Josina, e meus irmãos, Saray e Paulo, por estarem sempre ao meu lado, incentivando-me em todos os momentos da minha vida. Não teria conseguido sem vocês.

Aos meus orientadores, Marcelo Stemmer e Edson de Pieri, pela orientação, sugestões e experiências passadas durante todo o desenvolvimento da dissertação.

Ao amigo e colega Leandro Coelho, cujas dicas foram muito valiosas para o enriquecimento desse trabalho.

Aos professores Augusto Bruciaplagia, Eugênio Castelan, Jorge M. Barreto e Raul Guenther pela oportunidade de trabalhar com pessoas extremamente compententes e profissionais, e pelo aprendizado, antes de tudo, humano, sempre uma fonte de apoio e incentivo.

À todos os colegas e professores do Departamento de Automação, pela rica convivência desses últimos anos, pelo companheirismo, pelo apoio nas fases difíceis do trabalho, em fim, por todas aquelas trocas de vivências que nos fazem aprender, tanto profissionalmente como ser humano.

Em especial gostaria de agradecer aos amigos Karina, Jorge, Pierre, Marcos M., Carlos V., Miguel, Howard, Ivana, Nicole, Carlos R., Amarylis, Alejandro, Kelly, Karen, Marcos V., Lau, Filisberto, Sobral, Fabiano, Eduardo, Júlio, Waldoir Jr., Alexandre, Simas, Emerson, Luís Henrique, Noeli, Charles e Dominick pela convivência, pela amizade, pelo apoio e pelas festas, e que, com certeza, marcaram uma época da minha vida. Aprendi muito com vocês.

E por último, ao CNPq e à Universidade Federal de Santa Catarina pela oportunidade de ter uma bolsa para estudar em uma das melhores universidades do país, em um país onde pouquíssimas pessoas tem acesso à educação. Oportunidades são responsabilidades.

Resumo

Este trabalho tem como objetivo verificar o emprego de redes neurais artificiais - RNA's - no controle da complexa dinâmica dos manipuladores robóticos rígidos, constituída por equações não-lineares acopladas, apresentando imprecisões paramétricas nos parâmetros e coeficientes do seu modelo, e estando ainda sujeito às perturbações externas do ambiente.

No trabalho são abordados dois problemas distintos: o controle de posição e o controle simultâneo de força/posição.

Para o controle de posição é proposta uma estrutura de controle neural com treinamento on-line, *constituindo-se de uma variação dos controladores apresentado em [ZM96] e [EL97]. O controle simultâneo de força/posição está baseado na abordagem de controle híbrido dinâmico apresentado por [Yos90]. São propostas duas estratégias de controle neural, originais para o caso do controle de força/posição, derivadas dos resultados obtidos para o controlador neural de posição. Essas estratégias de controle também são implementadas de forma* on-line.

O desempenho dos controladores neurais é verificado em simulações numéricas para o caso do controle de um manipulador SCARA, considerando a sua estrutura completa. Análises de performance são realizadas e comparadas com controladores clássicos do tipo PD e controle inverso, para o caso do controle de posição, e com o controlador híbrido, para o caso do controle de força/posição.

O controlador neural de posição é implementado no manipulador SCARA Inter do Laboratório de Robótica da Universidade Federal de Santa Catarina. Seu desempenho é comparado com o controlador clássico PD existente no manipulador.

Abstract

The objective of this work is the evaluation of the employment of Artificial Neural Networks - ANN - in the control of rigid robotic manipulators, whose complex dynamics is comprised by non-linear and coupled equations, with uncertainties on its coefficients and parameters and been affected by external disturbances existing in the surrounding environment.

In this work, two main problems are considered: the position control problem and the simultaneous control of force and position.

In the position control case, a neural controller based on an on-line *training is proposed, based on the works presented by [ZM96] and [EL97]. The force/position control problem is based on dynamic hybrid control presented by [Yos90]. Two new neural control schemes are proposed to solve the control of force/position, arising from the results obtained in the neural control structure use in the position control. The two new neural controllers are also implemented with an* on-line *training strategy.*

The performance of the neural controllers are verified in numerical simulations, in the case of control of a SCARA robot, taking into account its complete structure. Performance analyses are made and compared with classical PD controller and the well-known inverse controller, for position control problem, and with the hybrid scheme proposed in [Yos90], for the control of force/position case.

The position neuro-controller is implemented on the SCARA robot. This robot belongs to the Robotics Lab, at he Federal University of Santa Catarina. The neuro-controller performance is compared with the classical PD controller.

Sumário

Capítulo 1

Introdução

Os dispositivos robóticos, ou robôs, podem ser empregados para realizar um conjunto amplo de atividades, que envolvem desde a sua utilização em sistemas de manufatura, como manipulação de objetos, operações complexas de usinagem, pintura ou soldagem, até aplicações mais recentes, como encontradas na área de navegação (robótica móvel) ou aplicações na área médica [FS92] O estudo desse tipo de dispositivos constitui-se de uma importante área dentro da engenharia, chamada de *robótica.*

O emprego de tais manipuladores em tarefas relativamente simples, como tarefas do tipo *pick-and-place*, já encontra-se amplamente difundido dentro dos ambientes industriais, não necessitando de um estudo mais aprofundado do comportamento dinâmico do manipulador e das estruturas de controle empregadas.

Entretanto, em tarefas em que as especificações incluem seguimento de trajetórias, precisões mais elevadas, uma boa repetibilidade, velocidades elevadas ou tarefas em que o efetuador final do manipulador entra em contato com superfícies, torna-se necessário ampliar o conhecimento das características do manipulador. A análise de sistemas de controle mais complexos, que garantam as especificações de controle para as mais variadas e inesperadas situações, exige, também, o aprofundamento no conhecimento do comportamento dinâmico do manipulador. Porém, o conhecimento a respeito desse comportamento muitas vezes é impreciso e limitado, e, portanto, torna-se desejável o emprego de estruturas de controle que sejam capazes de apresentar bons desempenhos para situações diversas, mesmo com a pouca informação disponível na fase de projeto da lei de controle [FS92], [HSZG92], [ZM96], [Nar96], e [Aga97].

1.1 Problemática do Controle de Robôs Manipuladores

O processo de modelagem e identificação do comportamento dinâmico do manipulador robótico envolve um conjunto de fatores que aumentam a complexidade no momento de estabelecer um modelo adequado para o mesmo. Esse comportamento dinâmico é representado por um conjunto de equações não-lineares, com um alto grau de acoplamento entre as suas variáveis, além de corresponder a um modelo multivariável [dWSB96], [LAD93] e [AS86].

Inúmeros fatores devem ser considerados durante as fases de modelagem do sistema e projeto do sistema de controle: incertezas e variações paramétricas; modelagem dinâmica complexa; dificuldade de medição das inércias dos elos e dos coeficientes de atrito das juntas; flexibilidade existente nas juntas; contatos entre manipulador e ambiente de trabalho; folgas e zonas-mortas nos motores de atuação; manuseio de cargas com valores de massa desconhecidos; perturbações e ruídos externos; etc. Além disso, o modelo dinâmico adotado para descrever o comportamento do manipulador constitui-se de uma simplificação da estrutura real.

Tarefas mais simples, como do tipo *pick-and-place*, onde são desenvolvidas trajetórias do tipo ponto-a-ponto, ou seja, onde deseja-se controlar somente as posições iniciais e finais da trajetória, sem se preocupar com a trajetória intermediária entre esses pontos, as especificações de malha fechada podem ser satisfeitas com controladores clássicos de ganhos fixos [AS86] e [LAD93]. Porém em aplicações em que se deseja o seguimento de trajetórias contínuas de posição, que envolvam velocidades mais elevadas (influência relevante dos termos não-lineares de inércia, gravitação e torques de coriolis), as especificações de malha fechada podem não ser atendidas. Dessa forma, torna-se necessário a utilização de estruturas de controle mais elaboradas, exigindo um conhecimento maior a respeito do comportamento dinâmico do manipulador.

Inicialmente, foram empregados controladores clássicos do tipo PID para o controle de trajetória de robôs, por apresentarem uma certa simplicidade no seu projeto. Porém, tais controladores não são capazes de garantir performances de resposta adequadas na maioria das situações, apresentando erros no seguimento de trajetória. Além disso, o desempenho dessas estruturas de controle é prejudicado quando as incertezas, perturbações e ruídos são consideradas [dWSB96].

Controladores mais elaborados foram sendo apresentados, de forma a suprir as deficiências dos controladores do tipo PID's. Controlador PD com compensação de gravidade, controladores baseados em modelo, controladores robustos, adaptativos ou

baseados nas propriedades de passividade do manipulador começaram a ganhar espaço dentro do controle de manipuladores robóticos [dWSB96].

Como alternativa aos métodos clássicos e baseados em modelo, estruturas de controle que apresentam um comportamento adaptativo são extremamente interessantes em sistemas em que se dispõe de pouca informação a seu respeito, e cuja dinâmica é variável com o tempo. Controladores adaptativos clássicos e controladores neurais apresentam essas características, e vêm apresentando bons resultados no controle de posição, velocidade e força de robôs manipuladores. Essa capacidade de adaptação permite que as incertezas e imprecisões decorrentes de um modelo limitado utilizado na fase de projeto, sejam compensados durante a atuação do controlador. Entretanto, os controladores adaptativos necessitam do conhecimento explícito do modelo paramétrico do manipulador [ZM96], [KV96], [HSZG92] e [FS92].

O problema de controle de robôs manipuladores pode ser dividido em dois grande grupos:

- Controle de posição[1]
- Controle simultâneo de força[2]/posição.

No primeiro caso, são considerados os movimentos do manipulador em um *ambiente livre* de contato. Dessa forma, o sistema de controle deve ser capaz de garantir o seguimento das referências de posição e velocidade do efetuador final. O controle de força/posição consiste em considerar as forças que estão agindo sobre o manipulador, decorrentes da interação entre o efetuador final e uma superfície existente na área de trabalho. O objetivo do sistema de controle passa a ser, além de garantir o seguimento de posições e velocidades, a capacidade de realizar o controle das forças emergentes dos contatos físicos do manipulador em algumas direções. Nesse caso, diz-se que a movimentação do manipulador se dá em um *espaço restrito* [AS86], [LAD93] e [dWSB96].

Nos dois casos, controle de posição e controle simultâneo de força/posição, os problemas decorrentes da complexidade, a falta de precisão e a influência de fatores externos no modelo dinâmico do manipulador são fatores que influenciam consideravelmente o desempenho dos controladores. Mais especificamente no caso de controle de força/posição, torna-se necessário o conhecimento extra da geometria da superfície em contato ou do seu comportamento dinâmico, uma vez em que as estratégias de controle encontram-se baseadas nessas informações [Wil92], [dWSB96] e [Yos90].

[1] por posição entende-se posição e orientação
[2] por força entende-se força e torque

1.2 Redes Neurais em Robótica

Dentro da teoria de controle, as redes neurais artificiais podem ser vistas como um formalismo para modelagem ou representação do conhecimento [HSZG92]. As propriedades de mapeamento funcionais não-lineares tornam atraente o emprego de redes neurais artificiais (RNA's) em controle de processos.

Uma RNA é capaz de aprender, ou armazenar, o comportamento dinâmico de um sistema, baseado somente em um conjunto de dados do tipo entrada/saída do sistema. Dessa forma, é possível empregar um sistema de controle baseado em uma arquitetura neural sem necessariamente considerar o modelo analítico e as não-linearidades existentes na planta [FS92], [Tor92], [ZM96], [Nar96] e [Kli96]. Além disso, as RNA's apresentam uma forte capacidade de generalização e associação, que conferem à rede a possibilidade de extrapolar o conhecimento não contido explicitamente no conjunto de dados utilizado para o treinamento [CBL98], [FS91], [Nar96] e [KV93].

Outras características que também tornam interessante o emprego de RNA's na área de identificação e controle de processos consistem do alto grau de processamento paralelo distribuído que a sua estrutura apresenta, a capacidade de tratar problemas não-lineares e a robustez a ruídos que a rede apresenta [Tor92], [FS92], [Nar96] e [FS91].

Na área de robótica, as redes neurais artificiais vem sendo empregadas para resolver problemas de planejamento de tarefas; planejamento e controle de trajetória; controle de contatos; coordenação, manipulação, locomoção e navegação; sensoreamento e percepção; dentre outros [KV96], [Nar96], [FS92] e [ZM96].

Em controle de robôs manipuladores, a dificuldade encontrada na modelagem da equação dinâmica do manipulador vem motivando o emprego de estruturas baseadas em redes neurais. Pelas propriedades que a rede apresenta, principalmente a capacidade de realizar mapeamentos não-lineares funcionais, as RNA's podem ser empregadas em estratégias de controle de robôs manipuladores.

Dentro da estrutura de controle, a RNA pode assumir diversos papéis. Ela pode ser empregada para mapear a dinâmica direta ou inversa do manipulador; como um estimador de parâmetros para uma estrutura de controle adaptativo ou preditivo; para aprender a dinâmica do meio em contato com o manipulador; atuando como um compensador de não-linearidades em conjunto com uma estrutura de controle linear [KV96], [ZZ96], [FS92]. As RNA's ainda podem ser empregadas com técnicas mais avançadas de controle, como controle ótimo, estrutura variável, controle por alocação de pólos, controle *fuzzy-neural*, *self-tunning* PID, dentre outras [Aga97].

Em estruturas de controle *off-line*, o funcionamento da RNA encontra-se baseado na capacidade que esta apresenta de aprender, ou armazenar, a informação contida

nos exemplos do conjunto de treinamento. Nesse caso, necessita-se de uma fase prévia de levantamento dos dados do conjunto de treinamento, através do emprego de uma estrutura de controle convencional, uma vez em que o robô manipulador é instável em malha aberta. Após essa fase de treinamento prévio fora do processo, a rede passa a atuar em uma estrutura de controle de posição ou força/posição. De forma a obter uma boa capacidade de generalização, o conjunto de dados deve conter um conjunto suficiente de informações. Entretanto, como o manipulador apresenta uma dinâmica não-linear, acoplada e multivariável, tal mapeamento só é possível com um conjunto relativamente grande de dados, exigindo estruturas de redes mais complexas e algoritmos de treinamento mais eficientes que o tradicional *backpropagation.*

A outra alternativa consiste em empregar métodos *on-line* de treinamento. Nessa abordagem, a rede apresenta as fases de aprendizagem e atuação sobre o processo simultaneamente. A vantagem consiste na não necessidade de levantamento de um conjunto de treinamento, e na relativa simplicidade nas estruturas de redes obtidas. Entretanto, deve-se assegurar a estabilidade do sistema manipulador mais controlador nos instantes em que a rede ainda não se encontra suficientemente treinada.

Atualmente, o emprego de RNA's no controle de posição e velocidade encontra-se relativamente estabelecido [HSZG92], [KV96], [ZZ96], [EL97], [ZM96], dentre outros. Entretanto, o problema de controle de força/posição, utilizando controladores clássicos ou não, ainda necessita de pesquisas e estudos de forma a estabelecer resultados mais sólidos e consistentes [VGBT93], [GHW97] e [JH98].

1.3 Proposta da Dissertação

O presente trabalho procura analisar o emprego de redes neurais artificiais na área de controle de manipuladores robóticos rígidos. São tratados os problemas de controle de posição e de controle simultâneo de posição/força de um manipulador SCARA, considerando os seus quatro graus de liberdade.

Busca-se verificar as propriedades de robustez dos controladores neurais frente às imprecisões e variações paramétricas decorrentes da falta de informação a respeito da dinâmica do manipulador e às perturbações externas atuando sobre o mesmo.

As RNA's assumem um papel auxiliar à estrutura de controle, procurando compensar não-linearidades e perturbações que não são tratadas pela estrutura principal de controle. Dessa forma, obtém-se estruturas de rede mais simples, com um número pequeno de neurônios para a resolução de um problema complexo que se constitui o controle de posição e controle de força/posição de robôs manipuladores.

O desempenho dos controladores é analisado através de simulações numéricas e

implementações práticas, onde são realizadas comparações com outras estratégias "não-adaptativas" de controle.

1.4 Estrutura da Dissertação

O capítulo 2 busca dar uma breve introdução ao tema estudado pela robótica. Inicialmente é apresentado um pequeno histórico dos robôs manipuladores. Em seguida são apresentados os aspectos relevantes à modelagem cinemática e dinâmica dos manipuladores robóticos rígidos. Na seqüência, introduz-se as equações matemáticas que descrevem o comportamento do manipulador SCARA, que serão empregadas nesse trabalho. Por fim são apresentados os problemas de controle de posição e velocidade no espaço de juntas - controle cinemático - e no espaço da tarefa, e, também, estratégias de controle simultâneo de força/posição.

No capítulo 3 é dada uma descrição geral das características, topologias e métodos de treinamento de redes neurais, com um enfoque para as estruturas de redes empregadas nesse trabalho (redes do tipo MLP — *Multilayer Perceptron*). Estratégias de controle neurais encontradas na área de controle de processos são apresentadas de uma maneira geral. Os algoritmos de treinamento para redes MLP empregados durante o desenvolvimento desse trabalho encontram-se descritos no apêndice *B*.

No capítulo 4 é discutido o emprego de estruturas neurais em controle de manipuladores robóticos. Aspectos quanto à definição da topologia das redes e arquiteturas de controle são apresentados. Estratégias neurais de controle de posição e controle simultâneo de posição/força encontrados na literatura são discutidas. Também são apresentadas as estratégias convencionais de controle e as estratégias neurais de controle empregadas nesse trabalho: controle de juntas e o controle híbrido de força/posição. Para o caso de controle de posição são considerados os casos do emprego de controladores do tipo PD, a técnica de controle inversa e uma variação da estrutura de controle neural apresentada em [ZM96] e [EL97]. No caso do controle de força, considera-se a abordagem proposta por [Yos90]. São propostas duas estratégias de controle neural inéditas e comparadas com o controlador híbrido dinâmico de [Yos90].

No capítulo 5 são apresentados os resultados das simulações e das implementações experimentais referente ao controle de posição. A performance do controlador neural é comparada com os controladores "não-adaptativos" apresentados no capítulo 4.

No capítulo 6 analisa-se o caso do controle de força. Considera-se o problema de controle híbrido de força/posição. O desempenho dos controladores neurais desenvolvidos são comparados com o desempenho do controlador híbrido dinâmico proposto por [Yos90].

As conclusões e perspectivas são apresentadas ao final, indicando os resultados obtidos e apontando novas propostas e complementações ao trabalho desenvolvido.

No apêndice A encontram-se listados os parâmetros do manipulador do Laboratório de Robótica. No apêndice B são apresentados os algoritmos de treinamento de redes do tipo MLP. Por fim, no apêndice C, os módulos mais importantes implementados para o controle neural do manipulador do Laboratório de Robótica são apresentados.

Capítulo 2

Robótica

2.1 Introdução

Inovações tecnológicas na área de informática, eletrônica e dispositivos mecânicos, o emprego mais efetivo de robôs manipuladores pela indústria e o crescente interesse acadêmico pela pesquisa na área de robótica, vêm contribuindo para o desenvolvimento dessa especialidade da engenharia, cada vez mais ampla e abrangente. Envolvendo diversos ramos da engenharia (eletro-eletrônica, mecânica, controle de processos, comunicação, manufatura) e ciências afins, como a informática, a robótica apresenta uma variedade ampla de aplicação, podendo ir desde os robôs industriais, utilizados para movimentação de peças e ferramentas, até aplicações mais recentes, como é o caso da robótica móvel.

No presente capítulo, serão apresentados alguns conceitos que descrevem o comportamento dinâmico dos robôs manipuladores rígidos e suas propriedades. Inicialmente é apresentado um pequeno histórico da robótica. Logo após introduz-se o modelo dinâmico de um robô rígido. Como exemplo, é apresentado o modelo matemático do manipulador SCARA utilizado nesse trabalho. Em seguida, a problemática de controle de robôs manipuladores — controle de posição e controle simultâneo de força/posição — é formalmente apresentada. Concluindo o capítulo, são discutidos os aspectos relevantes e os problemas encontrados no controle de robôs. Também serão apresentadas algumas estratégias não clássicas de controle.

2.2 Visão Geral da Robótica

2.2.1 Histórico

Inúmeras foram as tentativas de se construir, e "programar", um dispositivo mecânico que viesse a substituir o homem na execução de tarefas mais complexas, perigosas ou tediosas. Porém foi só no final da década de 30 e início de 40 que os primeiros manipuladores remotos, dispositivos mestre-escravo utilizados para movimentar materiais perigosos - radioativos por exemplo, começaram a aparecer. No início da década de 50 é construído pelo Laboratório de Servomecanismos do Instituto de Tecnologia de Massachusetts (MIT) a primeira máquina-ferramenta controlada por comando numérico (máquina NC) [KCN89].

Historicamente, a *era da robótica* iniciou na década de 50 [AS86], [KCN89], quando George C. Devol patenteou um dispositivo para manuseio de materiais, com "memória programável". Esse dispositivo teve origem na combinação de duas novas tecnologias: o *comando numérico* e o *manipulador remoto*. A tecnologia NC permitia coordenar o funcionamento de uma máquina-ferramenta através da informação armazenada em uma base de dados, ou programa. A utilização dessas características das máquinas NC em um manipulador remoto permitiu, então, a Devol criar um manipulador que pudesse ser programado e que funcionasse sem a necessidade de uma pessoa comandar o mesmo remotamente.

Nas décadas seguintes inovações foram somando-se ao campo da robótica: robôs móveis, integração com sistemas de visão, sensores de tato e força, utilização de servomotores para acionamento, primeiro robô industrial controlado por minicomputador (1973), efetuador final com punhos flexíveis, etc. As áreas que mais contribuíram para o avanço da robótica foram eletrônica, com um avanço considerável no final da década de 60 e toda a década de 70, e a área de informática que, com processadores mais poderosos, permitiu a implementação de sistemas de comando e controle mais poderosos, sofisticados e eficientes.

Segundo o *RIA - Robot Institute of America*, um robô pode ser definido como sendo [KCN89]:

> " *Um manipulador reprogramável e multifuncional projetado para mover e manusear materiais, peças, ferramentas, ou dispositivos especiais capazes de desempenhar uma variedade de tarefas através de movimentos variáveis programados.*"

Os robôs podem ser enquadrados em dois grandes grupos: robôs fixos (rígidos ou flexíveis) e robôs móveis. A área de interesse desse trabalho encontra-se dentro do

primeiro grupo, mais precisamente, os robôs manipuladores rígidos.

Os robôs flexíveis diferem dos robôs rígidos, por apresentarem, em seu mecanismo, ou estrutura mecânica, elementos não-rígidos, ou seja, que apresentam um certo percentual de flexibilidade, como por exemplo, elasticidade nas transmissões e juntas, ou interação existente nos contatos entre o manipulador e o ambiente de trabalho, ou ainda os robôs para emprego em tarefas no espaço que são concebidos utilizando materiais leves, e por isso, apresentam flexibilidades nos seus elos.

2.2.2 Manipuladores Robóticos Rígidos

Robôs manipuladores rígidos podem ser encontrados em diversas indústrias, desempenhando as mais variadas tarefas e atividades. Podem ser utilizados em montagem: na indústria eletro-eletrônica são empregados para montagem de circuitos integrados; em sistemas de manufatura: linha de montagem de carros (soldagem, pintura), integração entre máquina CNC e sistema de transporte; na indústria aeroespacial; na indústria metal-mecânica em geral, dentre outras.

Na execução dessas atividades, o dispositivo robótico utilizado deve satisfazer uma série de especificações e requisitos de forma a atender as máximas forças envolvidas, alcançar precisões desejadas, levando em consideração o tipo de ambiente em que o robô se encontra inserido.

Atualmente existe um número expressivo de robôs, com as mais variadas estruturas e atendendo a diversas necessidades de mercado. Pelo menos uma boa parte deles, pode ser decomposta em seus sistemas primários ou constituintes, facilitando a sua análise e compreensão de funcionamento[1].

Um robô manipulador pode ser analisado, e classificado, conforme a suas três principais estruturas (figura 2.1): sistema de controle, acionamento e o seu mecanismo, ou estrutura mecânica [KCN89].

A sua estrutura mecânica é constituída pelo conjunto de elos conectados através das suas juntas, ou eixos. Ela é composta por um braço, por um punho e pelo efetuador final. Este último é o elemento que executa a tarefa propriamente dita. É o braço do manipulador que fornece ao robô a capacidade de movimentação, permitindo-lhe atingir as diversas regiões no espaço de trabalho e realizar tarefas específicas.

De acordo com as características do ambiente de trabalho e da tarefa a ser desenvolvida, a movimentação do robô pode se dar em um espaço livre, caso não haja contato físico entre o efetuador final e o meio, ou em um espaço restrito, caso exista interação

[1]Daqui para diante quando for mencionado o termo robô, estar-se-á referindo ao manipulador robótico rígido

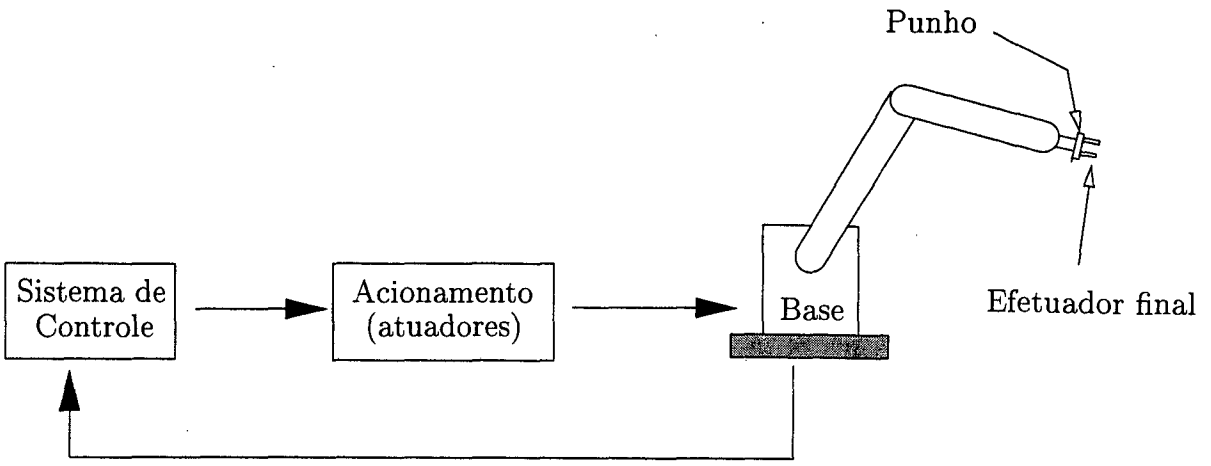

Figura 2.1: Elementos básicos do manipulador

do efetuador com o ambiente, gerando forças de contato, que devem ser consideradas [dWSB96]. O primeiro caso constitui-se do problema de *controle de posição e velocidade.* O segundo, por sua vez, corresponde ao problema de *controle simultâneo de força/posição.*

De acordo com a disposição dos elos e juntas (juntas de rotação apresentam deslocamentos angulares, e juntas de translação, deslocamentos lineares), os robôs podem ser classificados como: robôs de coordenadas cilíndricas, robôs de coordenadas esféricas, e robôs de coordenadas cartesianas ou prismáticos. O número de juntas de um robô define o número de graus de liberdade (*Degree of Freedom - DOF*) que o mesmo possui. Cada grau de liberdade corresponde a uma direção em que o manipulador pode se movimentar.

Quanto ao tipo de controle utilizado pelos manipuladores robóticos, pode-se enquadrá-los em dois grandes grupos [KCN89]: robôs com controle em malha-aberta e robôs com controle em malha-fechada.

O primeiro conjunto, também chamado de robôs não-servo controlados, apresentam uma estrutura de controle em malha aberta, empregando chaves de fim de curso e batentes para definir o final das trajetórias ou a sequência das operações, como por exemplo os robôs do tipo *pick-and-place.* Esse tipo de estrutura de controle era empregada nos primeiros manipuladores, em função da tecnologia disponível na época não poder suportar estratégias de controle mais sofisticadas.

Entretanto, em muitas tarefas torna-se necessário satisfazer certos requisitos de desempenho — como a precisão no posicionamento e a repetibilidade (capacidade do robô

atingir repetidamente uma mesma posição, dentro de uma certa precisão) — tornando-se necessário um controle mais refinado e preciso. Nesse caso, utiliza-se um sistema de controle em malha-fechada (robôs servo-controlados), comumente empregados em controle de posição, velocidade e força. Assim, para cada eixo, ou junta, existe uma malha de controle, responsável pela aquisição de sinais e atuação. O objetivo final do sistema de controle consiste em fornecer os sinais de comando para o sistema de acionamento do robô, de forma a garantir a correta posição e orientação do efetuador final, de acordo com as informações provenientes dos sensores.

De acordo com o tipo de trajetória desenvolvida pelo manipulador, as estruturas de controle empregadas podem se tornar mais complexas. Nas trajetórias do tipo ponto-a-ponto, onde as posições intermediárias entre os pontos iniciais e finais desejados não são relevantes, controladores com ganho fixo, ou até mesmo uma estrutura de controle em malha aberta podem ser suficientes. Entretanto nos casos de seguimento de trajetória e nas situações em que as dinâmicas não modeladas e a falta de conhecimento nos parâmetros do manipulador tornam-se mais significativas, o sistema de controle deve garantir que o manipulador execute o movimento de acordo com uma trajetória de referência, para todos os instantes de tempo da trajetória (controle de posição e controle simultâneo de força/posição).

O acionamento diz respeito aos elementos que transformam um sinal de comando, proveniente do sistema de controle, em um sinal de comando (drivers e amplificadores) para os atuadores (motores elétricos, dispositivos pneumáticos ou hidráulicos). Os atuadores elétricos são mais utilizados nas indústrias. Em operações que exijam cargas muito elevadas, empregam-se atuadores hidráulicos.

Em tarefas mais complexas, que envolvam sistemas de controle em malha-fechada, torna-se necessário um conhecimento mais profundo e preciso do comportamento do manipulador (sensores, atuadores e o meio onde o mesmo se encontra inserido). Para tanto, faz-se necessário o estudo de um modelo adequado que melhor descreva a dinâmica do manipulador. Tal estudo será apresentado na próxima seção.

2.3 Modelo Matemático do Robô Manipulador Rígido

O estudo da movimentação do robô manipulador pode ser dividido em duas partes: o estudo da cinemática e o estudo da dinâmica [dWSB96], [AS86], [LAD93]:

- *Modelagem cinemática*: estudo da movimentação do robô, em relação a um sistemas de coordenadas fixo, sem considerar as forças e torques que causam a

movimentação do robô. A modelagem cinemática de um manipulador envolve a cinemática direta e inversa, e a cinemática diferencial.

- *Modelagem dinâmica*: estudo das equações do movimento do manipulador, considerando as forças e torques que atuam sobre o mesmo. A cinemática é a base para a derivação das equações da dinâmica dos manipuladores.

As equações cinemática e dinâmica do manipulador empregam um conjunto de variáveis que é utilizada para descrever o seu comportamento.

Os modelos cinemático e dinâmico do manipulador podem empregar como variáveis, os valores dos deslocamentos das suas juntas de revolução e translação, para descrever o seu comportamento temporal. Esse conjunto de variáveis generalizadas forma um espaço de referência, denominado *espaço de juntas.*

No espaço de juntas, são considerados os deslocamentos, velocidades e acelerações em função das juntas do manipulador. O mapeamento entre os valores de torque e força nos atuadores e os deslocamentos nas juntas pode ser determinado diretamente pela equação dinâmica do manipulador, que pode ser derivada diretamente através dos métodos de Lagrange ou de Newton-Euler.

Outra possibilidade, consiste em utilizar um conjunto de variáveis que definem um espaço cartesiano. Esse espaço cartesiano pode corresponder ao sistema de coordenadas fixado na base, que é frequentemente empregado como sistema de referência para os demais sistemas de coordenadas do manipulador e das tarefas a serem executadas. O espaço de referência assim formado denomina-se *espaço operacional.*

Quando o espaço de trabalho é descrito pelo sistema de coordenadas situado diretamente no ambiente onde será desenvolvida a tarefa, como por exemplo o sistema de coordenadas situado na superfície de um objeto a ser manipulado pelo robô, o espaço de trabalho denomina-se *espaço da tarefa.*

Para que o robô possa desenvolver uma determinada operação, é necessário especificar e planejar a trajetória a ser seguida pelo manipulador. Essa trajetória pode ser especificada no espaço de juntas, ou no espaço cartesiano (tarefa ou operacional).

No espaço de juntas, a trajetória de referência fornece os valores desejados de deslocamento e velocidade para as variáveis das juntas. Nesse caso, a dinâmica do manipulador também é descrita em função das variáveis das juntas.

Entretanto, na maioria dos casos a tarefa a ser desenvolvida é descrita em função de um sistema de coordenadas cartesiano, e nesse caso, é necessário utilizar-se de uma transformação que leve os valores de posição e orientação do efetuador final para o espaço de juntas, e vice-versa. Dessa forma a análise da cinemática direta e inversa,

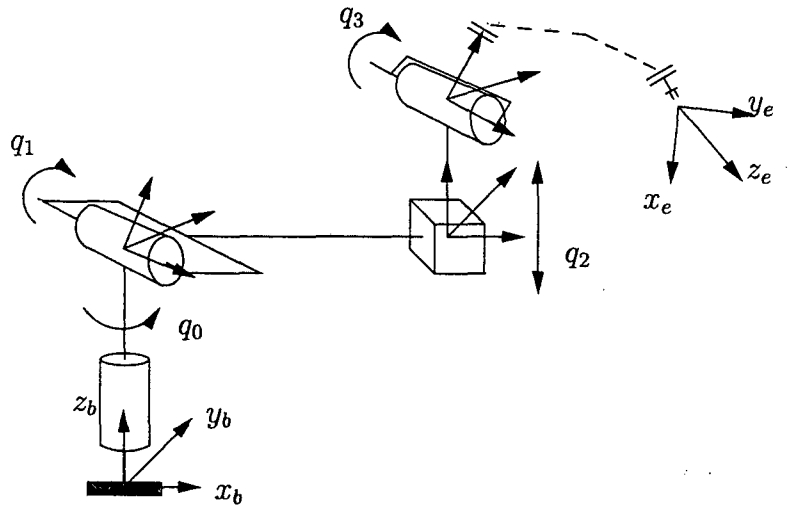

Figura 2.2: Sistema de coordenadas dos elos e juntas

que realizam os mapeamentos entre o espaço de juntas e o espaço cartesiano, torna-se relevante para o estudo da dinâmica do manipulador, e são apresentadas na sequência.

2.3.1 Cinemática dos Manipuladores

Cinemática Direta

A estrutura básica de um manipulador é constituída de vários elos conectados entre si através das suas juntas. A posição e a orientação do efetuador final, portanto, vão depender das posições, velocidades e orientações intermediárias de cada um dos elos.

Em cada junta é adotado um sistema de coordenadas, figura 2.2. O conjunto dos deslocamentos q_i das juntas do manipulador formam o espaço de juntas.

Usualmente podem ser utilizados mais dois sistemas de coordenadas auxiliares. O primeiro encontra-se situado no elo da base, definido pelos vetores ortonormais x_b, y_b e z_b. Esse sistema de coordenados é freqüentemente empregado como sistema de referência para todos os demais sistemas de coordenadas do manipulador. O segundo sistema de coordenadas é constituído pelos vetores x_e, y_e e z_e, também ortonormais, que correspondem ao sistema de coordenadas do efetuador final.

Para descrever o comportamento do efetuador final em função do sistema de coordenadas formado pelos vetores x_b, y_b e z_b é necessário transformar a posição e orientação do manipulador, medidos no sistema de coordenadas do efetuador final, para o primeiro sistema de coordenadas. Isso é obtido através do emprego das matrizes de transformação homogêneas. Dessa forma, é possível empregar as matrizes de transformações

para descrever a posição e a orientação de cada elo do braço do manipulador, em relação aos sistemas de coordenadas dos elos precedentes, e particularmente, em relação ao sistema de coordenadas da base.

A equação 2.1 apresenta a transformação da posição e orientação do efetuador final em relação ao sistema de coordenadas da base.

$$T(q) = \begin{bmatrix} R_e(q) & p_e(q) \\ 0 & 1 \end{bmatrix} \tag{2.1}$$

Na equação 2.1, $p_e \in \Re^{3x1}$ define a posição final do efetuador e $R_e = [n_e\ s_e\ a_e]$ é a matriz de rotação $\in \Re^{3x3}$ do efetuador final em relação ao sistema de coordenados da base. Observa-se que a matriz R_e é ortonormal, ou seja $R_e^T R_e = I$, e suas colunas são os vetores ortonormais x_b, y_b e z_b que formam uma base para o sistema de coordenadas fixado na base do manipulador.

Através da matriz de transformação homogênea é possível expressar a cinemática direta do manipulador, que faz o mapeamento das variáveis das juntas (posição e velocidade) para a posição e orientação do efetuador final.

Um método que auxilia na determinação das matrizes de transformação é o conhecido método de *Denavit-Hartenberg* [AS86] [dWSB96], [LAD93] e [AS86]. A notação de *Denavit-Hartenberg* utiliza um número mínimo de parâmetros para descrever completamente as relações cinemáticas do manipulador.

Cinemática Inversa

A cinemática inversa possibilita o conhecimento dos valores das variáveis das juntas baseado nos valores de posição e orientação do efetuador final. Isso possibilita encontrar as equações do movimento para as juntas, quando a tarefa é especificada no espaço cartesiano (operacional ou da tarefa).

Ao contrário do que acontece com a cinemática direta, que apresenta uma relação única entre a posição e orientação do efetuador final e as variáveis das juntas, a cinemática inversa podem apresentar múltiplas ou até mesmo infinitas soluções para suas equações. Além disso, as equações que definem a cinemática inversa são não-lineares, e dependem dos parâmetros de Denavit-Hartenberg, podendo até mesmo não existir soluções fechadas. Nesses casos somente soluções numéricas podem ser encontradas [dWSB96] e [AS86].

Cinemática Diferencial

O mapeamento entre as velocidades nas juntas $\dot{q}$, $\in \Re^{nx1}$, e a velocidade no efetuador final $\dot{x}$, $\in \Re(6x1)$, é dada pela equação cinemática diferencial

$$\frac{\partial x}{\partial t} = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J_g(q)\dot{q} \tag{2.2}$$

onde $\dot{p}$ são as velocidades lineares, ω são as velocidades angulares e J_g é uma matriz $\in \Re^{6xn}$, dita matriz Jacobiana do manipulador.

A matriz Jacobiana é responsável pelo mapeamento entre as velocidades angulares e lineares do manipulador. Ela representa o relacionamento infinitesimal entre os deslocamentos nas juntas δq e a localização e orientação do efetuador final do manipulador δx.

Quando o cálculo do Jacobiano está baseado na contribuição das velocidades lineares e angulares de cada junta do manipulador, obtém-se o *Jacobiano geométrico* - equação 2.2.

As três primeiras linhas do jacobiano geométrico estão associadas com as velocidades lineares, e as três últimas com as velocidades angulares. Dessa forma, pode-se escrever a matriz J_g como sendo

$$J_g = \begin{bmatrix} J_{l1} & J_{l2} & \cdots & J_{ln} \\ J_{a1} & J_{a2} & \cdots & J_{an} \end{bmatrix} \tag{2.3}$$

onde J_{li} consiste dos termos correspondentes às contribuições das velocidades lineares, J_{ai} à contribuição das velocidades angulares, e n é o número de graus de liberdade do manipulador.

Se a posição e a orientação do efetuador final forem especificadas em função de um conjunto mínimo de parâmetros, é possível obter a velocidade do efetuador final diferenciando a equação cinemática direta. Seja então $k : \Re_n \rightarrow \Re_n$ uma função de transformação diferenciável em q, e

$$x = \begin{bmatrix} p \\ \phi \end{bmatrix} \tag{2.4}$$

um vetor $\in R^{nx1}$, onde p é a posição final do efetuador e ϕ a sua orientação descrita por um conjunto de ângulos de Euler ou *row*, *pitch* e *yaw*. Assim é possível escrever a posição x como sendo $x = k(q)$. O valor de velocidade também pode ser obtido, então, através da seguinte equação:

$$\dot{x} = \begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \frac{\partial k(q)}{\partial q}\dot{q} = J_a(q)\dot{q} \tag{2.5}$$

A matriz jacobiana J_a assim obtida, consiste do *Jacobiano analítico* do manipulador.

Os jacobianos analítico e geométrico mantém uma relação de equivalência entre si, através da equação

$$J_g(q) = T_a(\phi)J_a(q) \tag{2.6}$$

onde

$$T_a(\phi) = \begin{bmatrix} I & 0 \\ 0 & T(\phi) \end{bmatrix} \tag{2.7}$$

com a matriz T sendo definida por

$$\omega = T(\phi)(\dot{\phi}). \tag{2.8}$$

O Jacobiano geométrico é empregado quando se está interessando no conhecimento do estado das grandezas físicas do manipulador e o Jacobiano analítico quando as grandezas no espaço da tarefa são consideradas [dWSB96].

2.3.2 Dinâmica dos Manipuladores

A equação dinâmica dos manipuladores robóticos, também chamada de equação do movimento, serve como base para o desenvolvimento de estruturas de controle. A modelagem dinâmica de um robô manipulador consiste em encontrar um mapeamento entre as forças exercidas pelos atuadores e agentes externos, e as posições, velocidades e acelerações nas juntas.

Ela pode ser obtida através de dois métodos [dWSB96], [AS86] e [LAD93]: pela formulação de Lagrange, baseada na medida das energias cinética, potencial e gravitacional agindo no sistema; e pela formulação de Newton-Euler, baseada na computação recursiva das posições, velocidades, forças e momentos entre os elos.

Dinâmica no Espaço de Juntas

Pela formulação de Lagrange, as variáveis das juntas $q = [q_1, \cdots, q_n]^T$ constituem um conjunto de coordenadas generalizadas do manipulador. A equação dinâmica do manipulador pode ser obtida através das equações de Lagrange:

$$\frac{\partial}{\partial t}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i \text{ com } i = 1, \ldots, n \tag{2.9}$$

onde

$$L = K - U \tag{2.10}$$

é o lagrangeano expresso em função da diferença entre a energia cinética K e a energia potencial U, e τ_i é uma força generalizada atuando sobre a junta i. Para o caso de juntas prismáticas, q_i é um deslocamento linear, e para o caso de juntas de revolução, o deslocamento q_i é do tipo angular.

A energia cinética do manipulador é dada pela soma da energia cinética de cada um dos elos, sendo que a energia cinética de um elo i considerado individualmente pode ser expressa por

$$K_i = \frac{1}{2} \text{ trace} \left[\sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial T_i}{\partial q_j} I_i \frac{\partial T_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right] \tag{2.11}$$

onde T_i é a transformação homogênea entre o sistema de coordenadas do elo e o da base, I_i é o tensor de inércia do elo e *trace* é a função matricial traço.

A energia cinética total do sistema é dada por

$$K = \sum_{i=1}^{n} K_i = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} h_{ij}(q) \dot{q}_i \dot{q}_j = \frac{1}{2} \dot{q}^T H(q) \dot{q} \tag{2.12}$$

com $H(q)$ sendo uma matriz $\in \Re^{nxn}$, simétrica e positiva definida. A matriz $H(q)$ é a matriz de inércia do manipulador.

Para o caso de robôs manipuladores com elos rígidos, a única fonte de energia potencial é a própria ação da gravidade. A energia potencial de um elo i de massa m_i, cujo centro de gravidade em relação ao sistema de coordenadas da base é r_i, é dada por

$$U_i = -m_i g^T T_i r_i \tag{2.13}$$

onde g é o vetor aceleração da gravidade, do tipo $g = [g_x \; g_y \; g_z \; 0]^T$ e T_i é a matriz de transformação homogênea. A energia potencial total será, portanto,

$$U = \sum U_i = -\sum_{i=1}^{n} q^T T_i I_i e_4 \tag{2.14}$$

com $e4 = [0\,0\,0\,1]^T$.

Os termos necessários para a resolução da equação lagrangeana são dados por

$$\frac{\partial L}{\partial \dot{q}} = \frac{\partial K}{\partial \dot{q}} = H(q)\dot{q} \tag{2.15}$$

$$\frac{\partial}{\partial t} \frac{\partial L}{\partial \dot{q}} = H(q)\ddot{q} + \dot{H}(q)\dot{q} \tag{2.16}$$

$$\frac{\partial L}{\partial q} = \frac{1}{2} \frac{\partial}{\partial q} (\dot{q}^T H(q) \dot{q}) - \frac{\partial P(q)}{\partial q} \tag{2.17}$$

e substituindo as equações 2.15, 2.16, e 2.17 na equação 2.9, obtém-se

$$H(q)\ddot{q} + \dot{H}(q)\dot{q} - \frac{1}{2}\frac{\partial}{\partial q}(\dot{q}^T H(q)\dot{q}) + \frac{\partial P(q)}{\partial q} = \tau. \tag{2.18}$$

Definindo o vetor de Coriolis e das forças centrífugas por

$$C(q,\dot{q}) = \dot{H}(q)\dot{q} - \frac{1}{2}\frac{\partial}{\partial q}(\dot{q}^T H(q)\dot{q}) \tag{2.19}$$

e o vetor de gravidade como sendo

$$G(q) = \frac{\partial P(q)}{\partial q} \tag{2.20}$$

chega-se a equação que descreve a dinâmica do manipulador rígido de n graus de liberdade:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{2.21}$$

A equação 2.21 é utilizada pela grande maioria dos autores na análise de estabilidade e no projeto de estruturas de controle. Entretanto, tal equação não inclui termos referentes às forças de atrito ou fricção dos elos e nem as perturbações externas ao manipulador.

Em [LAD93] é apresentada uma formulação mais geral. Na parcela correspondente ao atrito, pode-se considerar dois termos distintos: um referente ao atrito dinâmico e outro referente ao atrito viscoso. A modelagem e a estimação dos coeficientes de atrito das juntas do manipulador muitas vezes constitui-se de uma tarefa complexa [Gom95].

Sendo assim, a equação dinâmica do manipulador pode ser rescrita com os novos termos [LAD93]:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = \tau \tag{2.22}$$

onde $q \in \Re^{nx1}$ é o vetor de coordenadas generalizadas das juntas, $H(q) \in \Re^{nxn}$ é a matriz de inércia, $C(q,\dot{q}) \in \Re^{nx1}$ é o vetor de forças, momentos centrífugos e de Coriolis, $G(q) \in \Re^{nx1}$ é o vetor de forças gravitacionais, τ_d representa as perturbações sobre o sistema, $F(\dot{q}) = F_v\dot{q} + F_d$ corresponde a parcela referente ao atrito viscoso e estático, e $\tau \in \Re^{nx1}$ é o vetor de torque de controle sobre as juntas.

Dinâmica no Espaço da Tarefa

Na seção anterior, foram utilizadas as variáveis das juntas para descrever o sistema de coordenadas generalizadas utilizada pela formulação de Lagrange. Entretanto, muitas tarefas podem estar descritas de acordo com o sistema de coordenadas do espaço de trabalho do manipulador (espaço de tarefa ou espaço operacional), tornando-se interessante conhecer a equação dinâmica do manipulador em função desse novo sistema de coordenadas.

Como a formulação de Lagrange permite empregar qualquer sistema de coordenadas generalizadas para descrever a dinâmica do sistema, é possível através de uma transformação de coordenadas rescrever a equação dinâmica do manipulador 2.21 em função do sistema de coordenadas da tarefa.

Seja a posição e orientação do efetuador final dada pela equação 2.5

$$x = \begin{bmatrix} p \\ \phi \end{bmatrix} = k(q) \tag{2.23}$$

O sistema de coordenadas x, definido pela posição p e pela orientação ϕ, define o novo sistema de coordenadas. Derivando duas vezes a equação acima, obtém-se

$$\dot{x} = J_a(q)\dot{q} \tag{2.24}$$

e

$$\ddot{x} = J_a(q)\ddot{q} + \dot{J}_a(q)\dot{q} \tag{2.25}$$

onde J_a é o Jacobiano analítico do manipulador.

Isolando a derivada e a aceleração nas juntas nas equações 2.24 e 2.25, respectivamente,

$$\dot{q} = J_a^{-1}(q)\dot{x} \tag{2.26}$$

e

$$\ddot{q} = J_a^{-1}(q)\ddot{x} + \frac{d}{dt}(J_a^{-1}(q))\dot{x} \tag{2.27}$$

substituindo essas relações na equação da dinâmica do manipulador e pré-multiplicando todos os termos por $J_a^{-T}(q)$, obtém-se:

$$\bar{H}(q)\ddot{x} + \bar{C}(q,\dot{q})\dot{x} + \bar{G}(q) = F \tag{2.28}$$

onde

$$\bar{H}(q) = J_a^{-T} H(q) J_a^{-1} \tag{2.29}$$

$$\bar{C}(q,\dot{q}) = \left[J_a^T C(q,\dot{q}) J_a^{-1} + J_a^{-T}(q) \frac{\partial J_a^{-1}}{\partial t} \right] \tag{2.30}$$

$$\bar{G}(q) = J_a^{-T} G(q) \tag{2.31}$$

$$F = J_a^{-T} \tau \tag{2.32}$$

A equação 2.28 dessa forma, exprime o comportamento dinâmico do manipulador nas coordenadas generalizadas do espaço da tarefa dada por 2.21. Observa-se que as matrizes da equação 2.28 são dadas em função de q e $\dot{q}$.

Através da relação $\dot{q} = J_a^{-1}(q)\dot{x}$ é possível eliminar as variáveis q e $\dot{q}$ nas posições em que a inversa do Jacobiano existe, possibilitando rescrever a equação 2.28 somente em função de variáveis do espaço cartesiano.

Nos pontos no espaço cartesiano onde o Jacobiano se torna singular, chamados de singularidades, o robô manipulador apresenta uma perda de graus de liberdade, impossibilitando a realização de alguns movimentos.

Na fase de planejamento da trajetória deve-se contornar esse problema, evitando que a trajetória do efetuador final passe próxima às singularidades do sistema. Esse problema não ocorre no espaço de juntas, uma vez que as relações 2.24 e 2.25 são utilizadas somente para o espaço cartesiano de trabalho.

Na estratégia de controle híbrido de força/posição o problema de singulariedades também existe, pois a lei de controle exige a transformação de posição no espaço cartesiano para o espaço de juntas, através das equações 2.24 e 2.25. Dessa forma, deve-se garantir na etapa de planejamento da trajetória que o manipulador não venha a passar pelas proximidades dos pontos singulares.

Dinâmica com Movimento Restrito

Em determinadas tarefas o robô tem que entrar em contato com o meio e exercer uma força sobre esse ambiente, como em tarefas de esmirilhamento, polimento ou montagem de peças. Nessas situações, a dinâmica do robô é afetada pela interação meio/efetuador final, e se faz necessário o entendimento de como as forças resultantes dessa interação influenciam o comportamento do robô.

Além do controle da interação meio-efetuador final (controle de força), simultaneamente se faz necessário o controle do movimento do efetuador final no espaço da tarefa (controle de posição e velocidade) [dWSB96].

O cálculo do torque devido a uma força generalizada F, do tipo $F = [\gamma^T \mu^T]^T$, onde

γ é uma força externa e μ um momento externo aplicados sobre o efetuador final, pode ser obtido através do princípio do trabalho virtual [AS86] e [dWSB96]. Uma vez em que o manipulador rígido constitui-se de um sistema com restrições "holotômicas"e independentes do tempo, a sua configuração depende somente das coordenadas de juntas e não do tempo. Isto implica em que os deslocamentos virtuais coincidem com os deslocamentos elementares. Portanto, o trabalho desenvolvido pelo efetuador final e pelas juntas do manipulador no ponto de equilibro é dado por

$$\tau_e \partial q = \gamma^T \partial p + \mu^T \omega \partial t \tag{2.33}$$

onde ∂q é o deslocamento elementar nas juntas, ∂p e $\omega \partial t$ correspondem, respectivamente, aos deslocamentos elementares linear e angular do efetuador final. Levando em conta a equação 2.2 e a equação 2.33, o torque sobre as juntas resulta em

$$\tau_e = J_g^T F. \tag{2.34}$$

Dessa forma, a equação dinâmica do manipulador passa a conter mais um termo, referente aos torques externos provenientes do contato entre o efetuador final e uma superfície de contato - equação 2.35.

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + J_g^T F \tag{2.35}$$

Nas direções em que o movimento do manipulador é restrito, ocorre perda de graus de liberdade, limitando o movimento nessas direções. Para melhor compreender a atuação dessas restrições sobre a dinâmica do manipulador, faz-se necessário a modelagem das mesmas. Posteriormente, essa mesma modelagem será útil nas estratégias ditas híbridas, que separam o controle de movimento, nas direções em que o movimento do manipulador é livre, do controle de força nas direções onde o movimento é restrito.

2.3.3 Exemplo: Modelo Matemático do Manipulador SCARA

A seguir será apresentado o modelo matemático do manipulador SCARA, e corresponde a estrutura do manipulador existente no Laboratório de Robótica[2], utilizado para a realização dos experimentos dos capítulos 5 e 6.

[2] A dedução da equação dinâmica do manipulador SCARA empregada, corresponde àquela desenvolvida em [GWG98]

Cinemática do Manipulador

A forma construtiva de um robô manipulador do tipo SCARA, do inglês *Selective Compliant Articulated Robot Arm* é apresentada na figura 2.3. As duas primeiras juntas conferem movimentos de rotação, gerando deslocamentos angulares em torno de eixos verticais, trabalhando, portanto, em um plano horizontal. A terceira junta corresponde a um movimento de translação, realizando deslocamentos verticais ao longo do eixo z. E por último, a quarta junta executa movimentos de rotação sobre o eixo vertical, definindo a orientação do efetuador final do manipulador.

Como pode se observar, a estrutura do SCARA lhe confere uma grande rigidez no plano vertical, proporcionando flexibilidade no plano horizontal. Ao seu quarto elo pode ser fixado um efetuador final[3], permitindo o emprego do manipulador SCARA em um grande conjunto de aplicações.

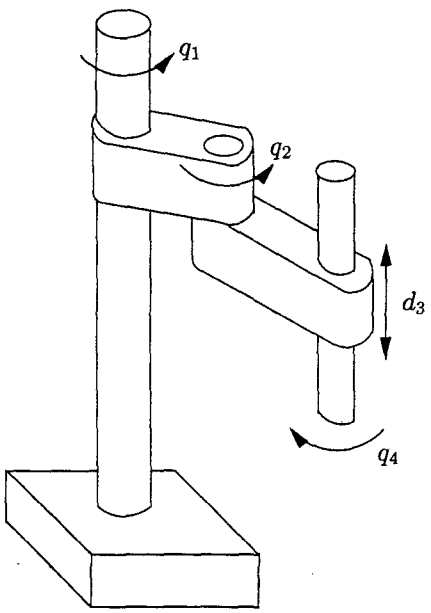

Figura 2.3: Robô SCARA

Para fazer o mapeamento das posições, velocidades e acelerações do espaço cartesiano para o espaço de juntas, emprega-se a cinemática direta, conforme visto na seção 2.3.1. A figura 2.4 mostra as variáveis de juntas e os parâmetros do manipulador.

Afixado ao elo da base, elo 0, define-se o sistema de coordenadas para o espaço operacional.

[3]Conjunto diverso de ferramentas e sensores

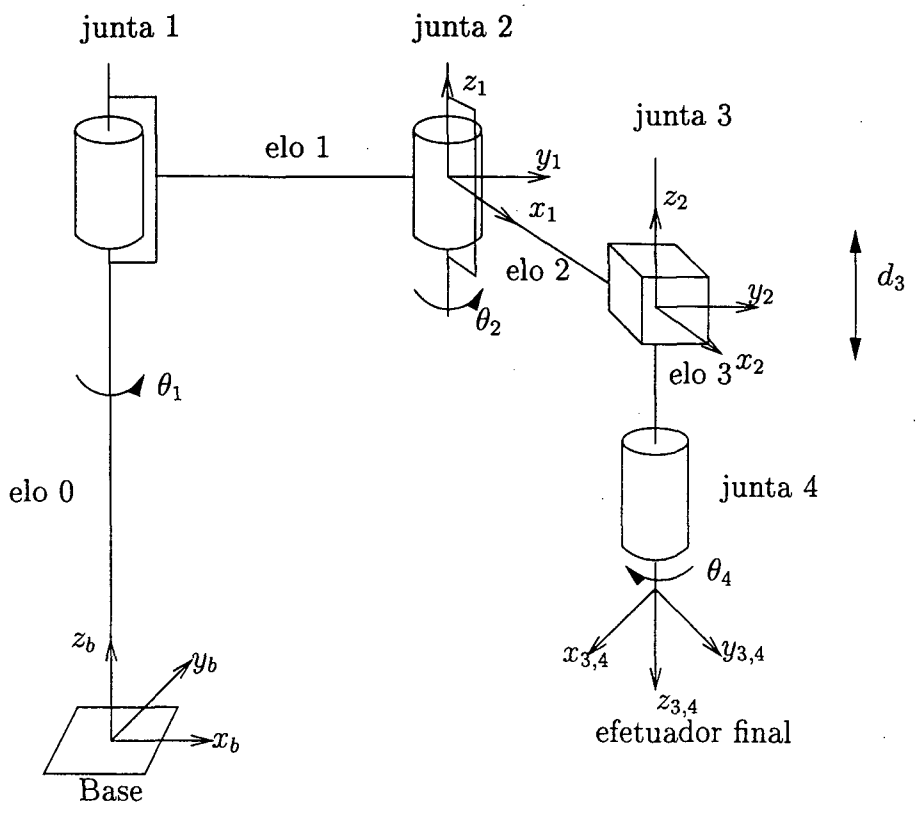

Figura 2.4: Parâmetros do SCARA

Os deslocamentos θ_i correspondem aos deslocamentos angulares nas juntas de rotação do manipulador, enquanto que a variável de junta d_3 representa o deslocamento linear na junta vertical de translação.

Os valores nominais de massa dos elos m_i, momentos de inércia I_i, comprimentos dos elos l_i e as distâncias dos centros de massa l_{ci} encontram-se no apêndice A. Valores de coeficientes de atrito nas juntas não estavam disponíveis, dessa forma, optou-se por não incluí-los no modelo.

De acordo com a figura 2.4, pode-se deduzir facilmente a cinemática direta do SCARA:

$$\begin{aligned} x &= l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ y &= l_1 \mathrm{sen}\theta_1 + l_2 \mathrm{sen}(\theta_1 + \theta_2) \\ z &= d_0 - d_3 \\ \theta &= \theta_1 + \theta_2 - \theta_4 \end{aligned} \tag{2.36}$$

onde $p = [x\ y\ z]^T$ são as posições do efetuador final no espaço operacional, de acordo com um sistema de coordenadas $[x_b\ y_b\ z_b]^T$, cuja origem se encontra no elo da base do manipulador. l_1 e l_2 correspondem aos comprimentos dos elos 1 e 2 do manipulador, respectivamente. As variáveis θ_1 e θ_2 são os deslocamentos angulares, medidos no sentido horário, das duas primeiras juntas de rotação, junta j_1 e junta j_2, respectivamente. O deslocamento vertical do manipulador é função do deslocamento da terceira junta, dado por d_3 e por d_0 que corresponde a origem do sistema de coordenadas afixado à terceira junta. A orientação do efetuador final é dada pela soma dos deslocamentos angulares nas juntas 1, 2 e 4.

Matriz Jacobiana

As velocidades podem ser obtidas através do Jacobiano analítico J_a, conforme seção 2.3.1. Dessa forma, empregando-se a equação 2.5, obtém-se:

$$\begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega \end{bmatrix} = J_a \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_3 \\ \dot{\theta}_4 \end{bmatrix} \tag{2.37}$$

O Jacobiano analítico do manipulador é expresso pela seguinte relação:

$$J_a = \begin{bmatrix} -l_1 \text{sen}\theta_1 - l_2 \text{sen}(\theta_1 + \theta_2) & -l_2 \text{sen}(\theta_1 + \theta_2) & 0 & 0 \\ l_1 \text{cos}\theta_1 - l_2 \text{cos}(\theta_1 + \theta_2) & l_2 \text{cos}(\theta_1 + \theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix}. \tag{2.38}$$

Equação Dinâmica

A dinâmica do manipulador pode ser expressa através da equação 2.21. As matrizes do seu modelo dinâmico encontram-se descritas na seqüência e atendem as propriedades apresentadas em [dWSB96].

A matriz de inércia $H(q)$ é dada por:

$$H(q) = \begin{bmatrix} h_{11} & h_{12} & 0 & I_4 \\ h_{21} & h_{22} & 0 & I_4 \\ 0 & 0 & m_3 + m_4 & 0 \\ I_4 & I_4 & 0 & I_4 \end{bmatrix} \tag{2.39}$$

onde os termos h_{ij} valem

$$\begin{aligned}
h_{11} &= I_1 + I_A + m_1 l_{c1}^2 + (m_2 + m_A) l_1^2 \\
&\quad + \left(\frac{m_2}{k} + m_A\right)(2 l_1 l_2 \cos\theta_2) + \left(\frac{m_2}{k^2} + m_A\right) l_2^2 \\
h_{12} &= I_A + \left(\frac{m_2}{k} + m_A\right) l_1 l_2 \cos\theta_2 + \left(\frac{m_2}{k^2} + m_A\right) l_2^2 \\
h_{21} &= I_A + \left(\frac{m_2}{k} + m_A\right) l_1 l_2 \cos\theta_2 + \left(\frac{m_2}{k^2} + m_A\right) l_2^2 \\
h_{22} &= I_A + \left(\frac{m_2}{k^2} + m_A\right) l_2^2
\end{aligned} \tag{2.40}$$

com $I_A = I_2 + I_3 + I_4$, $k = \frac{l_2}{l_{c2}}$ e $m_A = m_3 + m_4$.

A matriz de torques de Coriolís e torques centrífugos $C(q, \dot{q})$ é dada pela seguinte relação:

$$C(q,\dot{q})\dot{q} = \begin{bmatrix} c_{11} & c_{12} & 0 & 0 \\ c_{21} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_3 \\ \dot{\theta}_4 \end{bmatrix} \tag{2.41}$$

onde os coeficientes c_{ij} da matriz são dados por

$$\begin{aligned}
c_{11} &= -\left(\frac{m_2}{k} + m_A l_1 l_2\right) \dot{\theta}_2 \mathrm{sen}\theta_2 \\
c_{12} &= -\left(\frac{m_2}{k} + m_A\right) l_1 l_2 \mathrm{sen}\theta_2 \left(\dot{\theta}_1 + \dot{\theta}_2\right) \\
c_{21} &= \left(\frac{m_2}{k} + m_A\right) l_1 l_2 \dot{\theta}_1 \mathrm{sen}\theta_2.
\end{aligned} \tag{2.42}$$

O vetor de torques gravitacionais, por sua vez, é definido como sendo

$$G(q) = \begin{bmatrix} 0 \\ 0 \\ (m_3 + m_4) g \\ 0 \end{bmatrix} \tag{2.43}$$

onde g é o valor da aceleração da gravidade ($g = 9.81$ m/s^2).

2.4 Controle de Robôs Manipuladores

2.4.1 Controle de Posição e Velocidade

Quando o robô desempenha tarefas que não geram interações entre o meio e o manipulador — operações no espaço de tarefa livre — o objetivo do sistema de controle

consiste basicamente em comandar a posição, a velocidade, e eventualmente, a aceleração nas juntas do manipulador. O sistema de controle busca, então, que a trajetória desenvolvida pelo manipulador seja aquela especificada pela trajetória de referência. Para isso, o sistema de controle procura aplicar os torques necessários nas juntas de forma a garantir que o erro de posição e velocidade sejam nulos.

Os métodos de controle de trajetórias podem ser divididos em dois tipos: controle no espaço das juntas e controle no espaço da tarefa.

Controle no Espaço de Juntas

No controle no espaço de juntas, ou controle cinemático como é conhecido, a lei de controle é especificada em função das variáveis nas juntas. Entretanto, muitas das tarefas são especificadas de acordo com um sistema de coordenadas cartesiano, por exemplo, o sistema de coordenadas fixado ao elo da base.

Nesse caso, faz-se necessário mapear as trajetórias definidas no espaço cartesiano para o espaço de juntas através da cinemática inversa - figura 2.5. Porém, a utilização da cinemática inversa apresenta certas desvantagens intrínsecas aos mapeamentos não-lineares, tais como singulariedades e redundâncias, o que aumenta o grau de complexidade na geração da trajetória, e na especificação do controlador [dWSB96] e [Mar97].

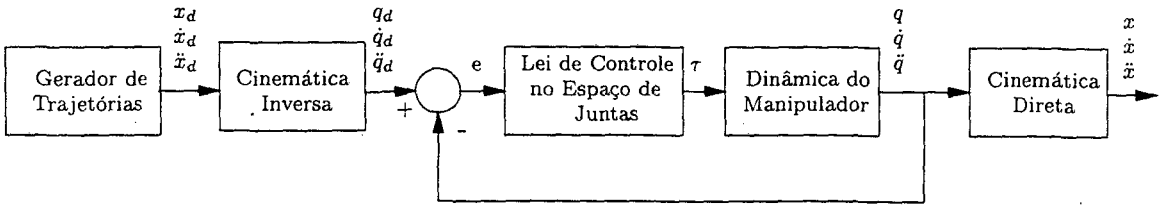

Figura 2.5: Controle Cinemático

No controle de espaço de juntas, os controladores são projetados baseados no modelo dinâmico do robô no espaço de juntas, ou seja, utilizando a informação das variáveis das juntas (posição e velocidade) para o cálculo da lei de controle.

Controle no Espaço da Tarefa

O controle no espaço da tarefa ocorre quando o controlador utiliza diretamente o valor das variáveis do sistema de coordenadas cartesiano adotado como referência. Esse método não apresenta o inconveniente do controle cinemático, que exige a necessidade

do conhecimento da cinemática inversa. Porém, ele não permite lidar facilmente com os efeitos das singularidades e redundâncias [dWSB96].

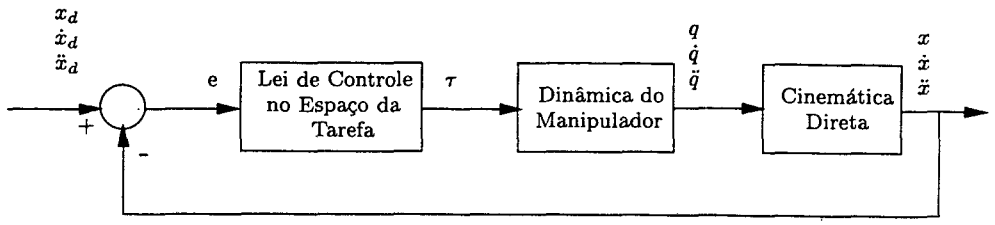

Figura 2.6: Controle no Espaço da Tarefa

As variáveis do espaço da tarefa são obtidas tendo como base as variáveis das juntas, através da cinemática direta. Dificilmente são empregados sensores para a medição direta das variáveis do espaço da tarefa, sendo mais usual a medida das variáveis das juntas.

2.4.2 Controle de Força

Em tarefas em que o robô venha a entrar em contato com o ambiente, o controle de movimento não é mais suficiente, sendo necessário controlar as forças e torques resultantes da interação robô-meio, sob pena de causar danos ao ambiente ou ao manipulador (figura 2.7). Entretanto, nem todas estratégias de controle necessitam da medida do valor de força para a sua compensação pelo sistema de controle.

No controle de força, existem duas situações que devem ser resolvidas: as situações de colisões inesperadas, cuja dinâmica, além de ser rápida, envolve forças elevadas; e as situações em que o robô deva seguir uma referência de força a ser exercida sobre o meio. Nesse trabalho é considerado o segundo caso.

O projeto do controlador de força se torna mais complexo, a medida em que a performance total do sistema não depende somente da dinâmica do manipulador, mas também das considerações feitas em relação à interação entre o manipulador e o ambiente: meio rígido ou flexível, definição das direções de controle de força e de posição. Além disso, a interação meio-robô varia de acordo com o tipo de material que compõe a superfície de contato [Wil92]. Todas essas características fazem do controle de força de robôs manipuladores uma área de pesquisa ainda com muitos pontos a serem pesquisados e esclarecidos.

Para medição dos valores de força e torque aplicados ao efetuador final do manipulador, são utilizados *sensores de força*. Sensores de força podem ser empregados

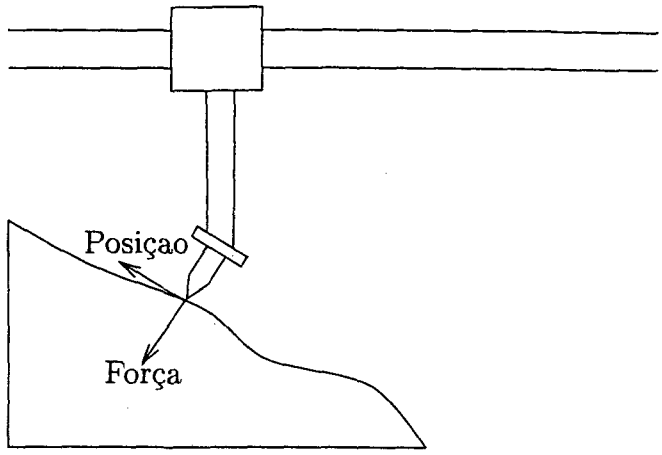

Figura 2.7: Robô Interagindo com Superfície

com as mais diferentes finalidades: avaliar o peso de uma determinada carga, seguir o contorno de uma superfície desconhecida, montagem de peças, robôs cooperando em uma mesma tarefa, etc.

O princípio básico dos sensores de força está na medição de deformações ou deslocamentos dos elementos elásticos que compõem o mesmo. Podem ser transdutores de deslocamento (potenciométricos, fotoelétricos, indutivos, capacitivos), semicondutores de efeito *Hall*, piezoelétricos, dentre outros [GFS97].

As estratégias empregadas para o controle simultâneo de força/posição podem ser divididas em [dWSB96], [AS86], [LAD93], [Yos90]: controle de rigidez, controle de impedância, controle paralelo e controle híbrido.

A seguir são apresentadas brevemente cada uma das estratégias de controle de força/posição. Salienta-se que o presente trabalho utiliza a estratégia de controle híbrido, mais precisamente a abordagem apresentada em [Yos90].

Controle de Rigidez

Na estratégia de controle rígido, o efetuador do manipulador é visto como sendo uma mola que exerce uma força sobre o ambiente, cuja rigidez pode ser ajustada para se adequar a diferentes meios. O valor dessa rigidez estipula o comportamento dinâmico do contato robô-meio. Através do controle da posição alcançada pelo manipulador, é possível controlar a força aplicada sobre o meio, uma vez que a o valor da força é dada pela relação:

$$f = K_e(x - x_e) \tag{2.44}$$

onde $K_e \in \Re^{nxn}$ é uma matriz diagonal, constante e positiva semi-definida, que descreve a rigidez oferecida pelo meio, $x_e \in \Re^{nx1}$ é um vetor referente à posição do meio, medido no espaço da tarefa, e $x \in \Re^{nx1}$ é um vetor que fornece a posição atual do efetuador final, também medido no espaço da tarefa.

No controle rígido, a força é controlada através do controle da posição x do efetuador final, não existindo uma leitura direta da força de contato entre o manipulador e o meio. Nessa caso, tem-se o controle indireto de força.

O controle rígido de força pode ser visto como um caso particular do controle de impedância, onde a equação 2.44 pode ser vista como uma impedância simplificada.

Controle de Impedância

O controle de impedância está baseado no conceito de que o controlador deve procurar regular o comportamento dinâmico existente entre o movimento do manipulador e as forças exercidas sobre o ambiente, ao invés de considerar os problemas de controle de força e movimento separadamente. A performance desejada é especificada por uma impedância dinâmica generalizada, sem existir, explicitamente, uma malha de controle de força - a força é controlada através do controle da posição.

Nesse tipo de controle, a dinâmica desejada é escolhida através da seleção das matrizes de inércia, amortecimento e rigidez aparente, atribuindo um comportamento de segunda ordem, diferente do controle rígido, onde supunha-se uma interação do tipo mola rígida - primeira ordem - entre o robô e o meio [dWSB96].

Apesar de poder tratar diversos tipos de contatos, essa estratégia de controle não permite o controle simultâneo das variáveis força e posição, e também não permite especificar a força aplicada sobre o meio. Isso se deve ao fato da lei de controle estar baseada na especificação de um comportamento dinâmico desejado para a interação entre o manipulador e o meio.

Controle Paralelo

Esse método de controle consiste em fornecer ao controlador baseado em impedância, a capacidade de fazer o controle de posição e força simultaneamente. A estratégia consiste em fechar mais uma malha de controle de força, externa a malha de posição, e com o emprego de um controlador adequado, torna-se possível regular a força de contato em um valor desejável. A lei de controle efetiva resulta em duas malhas de controle atuando em paralelo, a malha de força e a de posição.

Nas direções em que a força deve ser controlada, o controle de força deve ser preponderante sobre o controle de posição. Isso pode ser resolvido com a utilização de um

termo integral na malha de controle de força [dWSB96].

Esse método apresenta desvantagem de que o erro de força somente é nulo caso a força resultante da interação efetuador final/meio esteja alinhada com o vetor normal à superfície de contato.

Controle Híbrido de Força/Posição

O objetivo básico no controle híbrido consiste em particionar o espaço da tarefa em dois subespaços: o subespaço de controle de força e o subespaço de controle de posição. Esses espaços podem variar com o tempo, entretanto, as ações de controle de posição e força serão sempre feitas em subespaços complementares.

Os subespaços de controle e força são obtidos através de matrizes de seleção, que são complementares, e derivam da geometria do espaço onde o manipulador se encontra inserido. Portanto, na abordagem híbrida, a especificação da lei de controle depende do conhecimento da geometria do meio, que define as matrizes de seleção. Caso o meio, ou a tarefa, seja alterada, uma nova lei de controle tem que ser definida.

Existem, ainda, abordagens que mesclam as estratégias de controle híbrido com as propriedades do controle de impedância [dA97], [AS86], [LAD93], dentre outros.

A estratégia de controle híbrida será utilizada nesse trabalho. No capítulo 4 será apresentada a estratégia de controle híbrida baseada em [Yos90].

2.5 Conclusão

O aumento da complexidade nas tarefas desempenhadas pelos robôs manipuladores, exigiram um aprofundamento no conhecimento do comportamento dinâmico de tais sistemas. Um conjunto de equações diferenciais não-lineares e acopladas são empregadas para compor o modelo matemático, chamado de modelo dinâmico do manipulador (equação 2.21). Esse modelo constitui a base para o estudo das características e propriedades dos manipuladores, e são empregados nos projetos dos sistemas de controle.

O estudo do comportamento dinâmico, visto na seção 2.3.2, está baseado na cinemática do manipulador, equações que descrevem o seu movimento livre, e na sua dinâmica, que descreve o comportamento do manipulador em relação às forças e torques atuantes sobre o mesmo durante a sua movimentação.

Os problemas de controle encontrados nos manipuladores rígidos são divididos em dois grupos: controle de posição e controle de força. No primeiro preocupa-se com a movimentação do manipulador em tarefas em que não existam interações entre o efetuador final e o meio. O controle de força, por sua vez, procura resolver os problemas

que surgem quando o manipulador entra em contato com uma superfície. Atualmente, o problema de controle de posição e velocidade já encontra-se bem explorado, entretanto, o controle de força ainda constitui uma área com muitos avanços a serem feitos.

O presente trabalho se concentra no estudo de técnicas de redes neurais artificiais para o controle de posição e para o controle híbrido de força/posição de robôs manipuladores. No próximo capítulo serão apresentados os aspectos mais relevantes no emprego de redes neurais artificiais em controle.

Capítulo 3

Redes Neurais Artificiais

3.1 Introdução

Há muito tempo o homem vem tentando implementar dispositivos e sistemas que apresentem um comportamento inteligente. A área que se preocupa com esse estudo, denomina-se *Inteligência Artificial - (IA)* e divide-se em dois grandes ramos: *IA Simbólica* e *IA Conexionista*[1]. Enquanto a IA simbólica se preocupa em entender e descrever a lógica que descreve o processo inteligente adotado para resolver um determinado problema, a IA conexionista busca reproduzir a estrutura da qual emerge esse comportamento inteligente — o cérebro humano — sem se preocupar com os aspectos que definem a lógica adotada por esse mecanismo.

Inspiradas na estrutura física do cérebro humano, as redes neurais artificiais (RNA's) tentam reproduzir as funções das redes biológicas, buscando implementar seu comportamento básico e sua dinâmica. As RNA's começaram a serem utilizadas na resolução de problemas em diversas áreas, como controle de processos, condicionamento de sinais, reconhecimento de padrões, dentre outras.

Uma rede neural artificial pode ser definida como um

> *"sistema paralelo e distribuído composto por unidades de processamento simples (nodos ou neurônios), que computam determinadas funções matemáticas* [CBL98]".

Essas unidades de processamento individual, ou neurônios, são os elementos básicos de uma RNA, e encontram-se ligados entre si através de conexões, uma analogia às

[1] Alguns autores propõe agrupar as técnicas de Redes Neurais, Computação Evolutiva e Lógica Fuzzy sob uma nova designação: *Inteligência Computacional.*

conexões sinápticas na estrutura cerebral — essa é a origem do termo conexionismo. Essas conexões, por sua vez, possuem valores, ou pesos, que reforçam ou inibem a comunicação entre os neurônios e entre as entradas e saídas da rede.

A RNA apresenta uma forte capacidade de representação interna de problemas. Esse conhecimento é armazenado pela rede através dos valores dos pesos nas suas conexões. Contudo, esse conhecimento encontra-se disperso pelas conexões da rede, não sendo possível localizar fisicamente onde se situa um determinado conhecimento ou procedimento lógico. Esse é o motivo pelo qual muitos autores comparam as RNA's com caixas-pretas, não sendo necessário (e nem possível), preocupar-se com o funcionamento interno da rede ou com a forma em que o conhecimento vai ser armazenado ou estruturado, mas tão somente com a relação entrada-saída da rede propriamente dita.

Outra importante característica apresentada pelas RNA's reside no paralelismo inerente à sua estrutura, formada por um conjunto de unidades de processamento independentes, os neurônios, interligados pelas suas conexões. Decorrente desse fato, observa-se que os neurônios por si só não são poderosos em termos de representação de conhecimento, mas sim o grau de conectividade que a estrutura da rede apresenta [FS91], [KV93] e [CBL98]. Esse paralelismo tem motivado vários pesquisadores em empregar RNA's como uma alternativa à computação sequencial - a *neurocomputação*.

3.2 Modelagem da Rede Neural

As equações que descrevem o funcionamento de RNA tiveram origem quando McCulloch e Pitts [MP43] propuseram o primeiro modelo matemático para o neurônio. Desde aquela época, muitas inovações foram alcançadas, novos modelos de comportamento foram propostos, inspirados no comportamento dinâmico e nas reações eletroquímicas que ocorrem entre os neurônios existentes no cérebro humano.

3.2.1 Conceituação Biológica

Composto por uma rede com cerca de 10^{11} neurônios ligados entre si por aproximadamente 10^{13} conexões sinápticas, o cérebro humano serviu de inspiração para as RNA's.

O neurônio biológico consiste em uma célula especializada, responsável pela transmissão de impulsos eletroquímicos provenientes das suas terminações nervosas, conectadas a um órgão ou sensor específico (temperatura, tato, visão, etc.), ou aos demais neurônios da estrutura cerebral.

Um neurônio é composto pelo corpo celular, ou *soma* e diversas ramificações - figura 3.1.

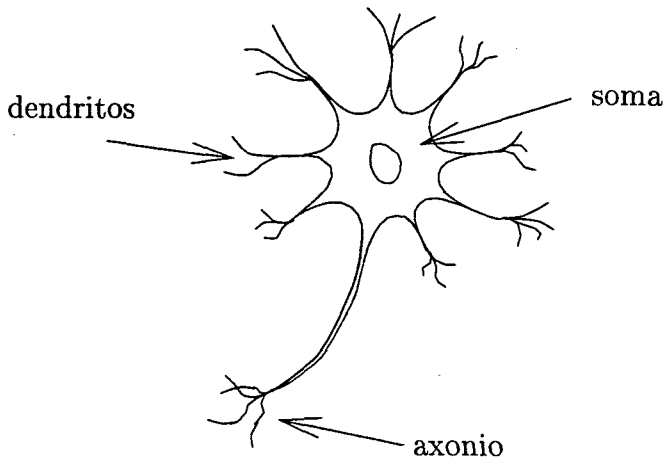

Figura 3.1: Neurônio biológico

As ramificações são compostas pelas terminações que conduzem os impulsos elétricos até o corpo celular, os *dendritos*, e também por uma ramificação, geralmente única, que transmitem o sinal do corpo celular para fora do neurônio, o *axônio*. Se o percentual de sinais provenientes dos dendritos, em um intervalo curto de tempo, for suficientemente alto, a célula dispara, produzindo um impulso nervoso através do axônio. Também é possível um axônio estar conectado a outros axônios ou diretamente com o corpo de outro neurônio.

Os impulsos nervosos são transmitidos através das conexões existentes entre o axônio de um neurônio e os dendritos de outros neurônios. Essas conexões são conhecidas como *sinapses*. Um neurônio pode ter entre 10^3 à 10^4 sinapses, e pode receber pulsos nervosos de cerca de 10^3 neurônios.

Nas sinapses existem substâncias químicas, chamadas de neurotransmissores, que propagam ou inibem o impulso nervoso. A transmissão da informação entre os neurônios dependem do tipo do neurotransmissor, da sua abundância no espaço sináptico e da sensibilidade da membrana a excitações eletroquímicas. O comportamento global da rede sináptica depende da intensidade que cada um dos seus neurônios tem de excitar ou inibir outros neurônios. Portanto, o "conhecimento", ou informação, que uma rede apresenta está intimamente associada aos processos eletroquímicos que ocorrem nas sinapses, e as alterações que ocorrem no comportamento dessas conexões influem o comportamento global da rede: é o processo de aprendizado.

Assim, observa-se que o conhecimento está na forma e na intensidade que os

neurônios ligam-se uns aos outros, ou seja, a informação está armazenada nas conexões sinápticas entre os mesmos.

3.2.2 Neurônio Artificial

O modelo proposto por McCulloh e Pitts [MP43], consistia em uma simplificação do neurônio biológico. O modelo resultante consistia em uma estrutura com n terminais de entrada, que representam os dendritos, um terminal de saída, que representa o axônio. Cada um dos terminais de entrada tem um peso associado, que representa o valor da conexão sináptica. A saída da rede, é ativada quando a soma de cada sinal de entrada multiplicada pelo respectivo peso supera um determinado valor, chamado de *threshold* ou *bias*, tendo a saída da rede um comportamento binário.

Além dessa simplificação, McCulloh e Pitts também assumiram que todos os neurônios disparam sincronamente, ou seja, disparam no mesmo tempo e que suas entradas avaliadas no instante t produzem uma saída no instante $t+1$. Tal comportamento síncrono não ocorre nos neurônios biológicos.

A partir desse modelo, várias outros modelos apareceram. Um modelo mais geral pode ser visto na figura 3.2 [Bar97].

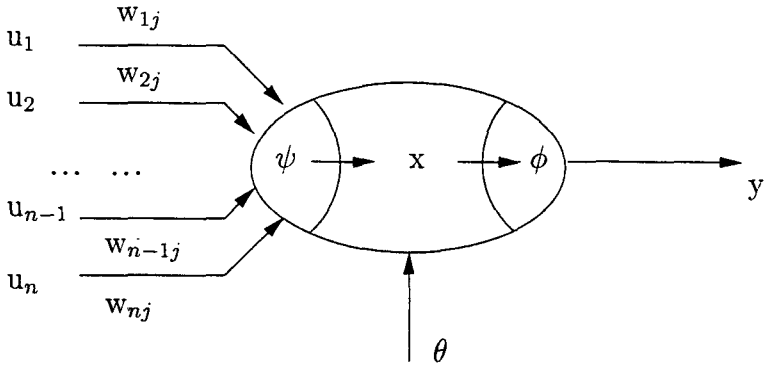

Figura 3.2: Neurônio Artificial

As entradas u_i são multiplicadas pelos respectivos pesos w_i e combinadas para produzir um estado de ativação do neurônio, através da função de entrada do neurônio Ψ. De acordo com uma função de ativação Φ, a partir do estado de ativação x, o neurônio produz a saída y. O termo θ corresponde ao *bias* do neurônio.

Geralmente são empregadas como função Ψ de entrada do neurônio a soma das entradas $u_i w_i$. Na literatura, podem ser encontradas redes que empregam outros tipos de funções de entrada, como por exemplo a função produto, mas que na prática não

são muito utilizadas [KV93].

A função de ativação Φ do neurônio pode ser implementada através de várias funções. As funções que são mais empregadas normalmente correspondem as funções lineares, rampa, degrau e sigmoidais.

A função rampa é geralmente empregada como uma função não-linear simplificada. As funções sigmoidais representam um conjunto de equações semilineares, limitadas e monotônicas, e freqüentemente empregadas em implementações de RNA's. Uma das funções sigmoidais mais comuns em RNA's, consiste da função logística, definida pela equação

$$\Phi(x)_i = \frac{1}{1 + e^{-x/T}} \tag{3.1}$$

onde o parâmetro T define a inclinação da curva.

Outros tipos de funções sigmoidais também bastante empregadas são as funções tangente hiperbólica e arcotangente. As funções logística e tangente hiperbólica apresentam uma curva de resposta muito semelhantes, com a diferença de que a primeira função apresenta uma saída variando entre 0 e 1, enquanto que a segunda, de -1 a 1.

A classe da função de ativação utilizada pouco influencia a saída da rede; entretanto, ela altera consideravelmente a velocidade do treinamento das RNA's [Kli96].

A equação do funcionamento de um neurônio que tem como função de entrada o somatório dos pesos multiplicados pelas entradas é dada por:

$$\begin{aligned} x &= \sum_i u_i w_i + \theta \\ y &= \Psi(x) \end{aligned} \tag{3.2}$$

Esse tipo de estrutura de neurônio é comumente empregada em redes do tipo MLP (*Multilayer Perceptron*), que serão apresentadas na seção 3.2.3.

Nas redes de função de base radial, *RBF - radial basis function* - a função de ativação do neurônio é definida por uma função do tipo $\Phi_i(x) = \phi(||x - c_i||)$, onde c_i correspondem ao centro da função de base [ZM96]. A função ϕ pode ser definida de várias maneiras, como $\phi(r) = r^2 \log(r)$, ou pela função Gaussiana $\phi(r) = exp[-r^2/\sigma^2]$, onde r é a distancia Euclidiana do vetor de entrada x até o centro da função de base c_i. Esse tipo de estrutura apresenta uma melhor capacidade de aproximações de funções não-lineares em comparação com as redes multicamadas [HSZG92], [ZM96] e [KV96].

Apesar de estarem fortemente inspiradas no modelo biológico, as RNA's estão ainda muito distante de representar satisfatoriamente, do ponto de vista físico, o comportamento real das redes neurais naturais. Entretanto, tal inspiração biológica serve como

fonte de referência na construção de estruturas mais sofisticadas de RNA's, e essas, por sua vez, auxiliam no entendimento das estruturas complexas que formam o cérebro humano.

3.2.3 Redes Multicamadas

Uma rede neural artificial é definida pela sua topologia, que consiste do tipo de neurônios empregados e a forma como estes encontram-se interligados.

Uma RNA pode conter uma ou várias camadas de neurônios. Suas conexões podem ser do tipo *feedforward*, ou acíclica, quando a saída de um neurônio da camada i somente pode ser conectada a um neurônio de uma camada adjacente (mais próximo da saída), ou do tipo *feedback*, ou cíclica, quando as saídas dos neurônios podem ser conectados em neurônios de qualquer camada, oferecendo um comportamento dinâmico à rede neural.

Quanto ao funcionamento temporal, as redes podem ser estáticas, quando a saída atual da rede depende somente do seu estado atual, ou dinâmica, quando a saída da rede depende dos valores anteriores de tempo. As redes dinâmicas podem ser utilizadas em aplicações em que se faz necessário um processamento temporal. A dinâmica pode ser introduzida na rede através das realimentações entre os neurônios, através do emprego de *delays* na entrada da rede ou através da utilização de uma função de ativação dinâmica para os seus neurônios [FS91], [KV93] e [CBL98].

Uma topologia muito empregada consiste das redes multicamadas, também conhecidas como redes *Multilayer Perceptrons*, ou simplesmente, redes MLP. O emprego de redes MLP ressurgiu em 1986, com a publicação de [RM86], onde o emprego do algoritmo *backpropagation* foi "redescoberto".

As redes MLP conseguem resolver problemas conhecidos como não-linearmente separáveis, os quais não eram resolvidos por redes compostas por uma camada de neurônios [FS91] e [KV93]. As redes MLP tem a capacidade de realizarem mapeamentos não-lineares complexos, sendo, portanto, muito úteis em aplicações como reconhecimento de padrões e controle de processos.

Nesse tipo de rede, as entradas estão conectadas aos neurônios de entrada. Os neurônios intermediários entre os neurônios de entrada e os de saída, correspondem aos neurônios das camadas escondidas. A figura 3.3 representa uma rede *feedforward* multicamadas:

As redes MLP são redes compostas por várias redes do tipo *Perceptron*, que são redes de uma camada de neurônios, com neurônios do tipo apresentado pela figura 3.2. Nas redes MLP, os neurônios estão arranjados de forma a receber sinais de neurônios

da camada anterior, e fornecerem sinais para os neurônios da camada adjacente, não existindo comunicação entre neurônios da mesma camada. Além disso, as redes MLP correspondem a estruturas estáticas.

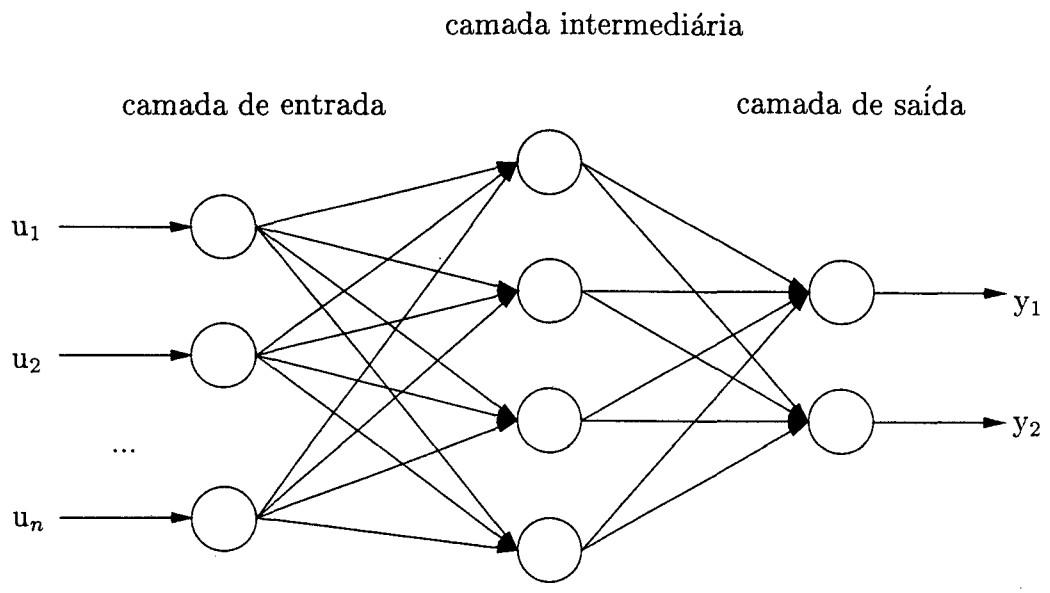

Figura 3.3: Rede multicamadas.

A definição da topologia da rede é ainda uma tarefa complexa, baseada mais em regras heurísticas do que formais. Os passos para a definição da topologia da rede envolvem uma série de fatores, como a escolha do tipo da rede propriamente dita (*feedforward*, com ciclos, dinâmica ou estática, etc.), a definição do número de camadas, a definição do número de neurônios, a escolha de um método de treinamento, etc., que devem ser considerados na fase de projeto da arquitetura e topologia de rede neural.

3.2.4 Aprendizado

Em RNA's, o procedimento usual para o seu emprego em resolução de problemas inicia pela fase de *aprendizagem*. Através de um método de aprendizagem é possível fornecer à rede um determinado comportamento, como por exemplo, a capacidade de funcionar como uma memória, reconhecer certos padrões, ou aprender a resolver um problema específico.

A capacidade de aprender através de exemplos e de generalizar a informação aprendida são uns dos maiores atrativos apresentados pelas RNA's e tem motivado a sua utilização nas mais variadas aplicações. Com o forte processamento em paralelo e com as capacidades de aprendizado, associação, generalização e auto-organização, a RNA

é capaz de extrair informações não apresentadas de forma explícita através de seus exemplos [FS91], [CBL98] e [KV93].

O procedimento básico de aprendizagem em uma RNA consiste em ajustar os valores dos pesos das suas conexões, utilizando um método ou algoritmo apropriado.

Os métodos de aprendizagem podem ser divididos em dois grandes grupos: *aprendizado supervisionado* ou *aprendizado não supervisionado.*

No aprendizado supervisionado, existe a figura do *professor*, um agente externo à rede, responsável por fornecer a saída desejada para a rede, quando da aplicação de um conjunto de padrões na entrada da rede. O método mais conhecido de aprendizado supervisionado para redes do tipo MLP, baseado na *regra delta generalizada*, constitui-se do algoritmo *backpropagation* [FS92], [FS91], [KV96], [Nar96] e [CBL98]. No apêndice B encontram-se descrito com maiores detalhes esse algoritmo, bem como outros algoritmos acelerados, derivados do algoritmo *backpropagation.*

O método de aprendizagem não-supervisionado, caracteriza-se pela ausência de exemplos (professor). A partir do momento em que a rede estabelece uma relação entre os dados de entrada, desenvolve-se nela uma capacidade de formar representações para codificar as características das entradas e criar novas classes automaticamente. Essa forma de aprendizado está essencialmente baseada na informação local do sistema, e em sinais internos do mesmo, sem a necessidade de uma referência externa. O aprendizado *hebbiano* e o aprendizado competitivo são exemplos de métodos de aprendizado não-supervisionado [CBL98].

3.3 Identificação e Controle utilizando Redes Neurais

No contexto da teoria de controle, as RNA's são vistas como um formalismo para modelagem de processos; ou uma estrutura para representação do conhecimento: o conhecimento da planta ou o mapeamento das suas características são armazenadas implicitamente nas RNA's. As propriedades de mapeamentos funcionais não-lineares tornam atraente o emprego de RNA's em controle [HSZG92].

As características de generalização do conhecimento, capacidade de aprendizado e o forte paralelismo apresentado pelas RNA's, tem motivado a utilização de tais estruturas nas mais variadas áreas de interesse, como identificação e controle de processos; processamento de sinais para controle; gerenciamento de alarmes; detecção e correção de falhas em sensores; reconhecimento de padrões em sistemas especialistas; robótica (visão, reconhecimento de voz, controle de braços mecânicos); dentre outras aplicações

[Tor92].

3.3.1 Vantagens das RNA's

Além das características de generalização, aproximação funcional e mapeamento não-linear, existem outras características mais específicas que tornam as RNA's atrativas para o emprego em sistemas de controle [FS93], [Kli96], [HSZG92], [Nar96], [FS92] e [Aga97].

Processamento Paralelo Distribuído

O processamento obtido por uma RNA é relativamente elevado devido a sua estrutura paralela. Uma vez treinada, uma RNA apresenta uma velocidade de processamento ideal para aplicações em tempo real. Outro aspecto interessante consiste nas implementações em *hardware* de RNA's. Nessas implementações são obtidas altas velocidades de processamento devido a sua estrutura paralela - (*Neurocomputação*). Além disso, o paralelismo pode oferecer um grau de tolerância a falhas maior que o apresentado por técnicas convencionais (seqüenciais).

Aproximador universal

Existem estruturas de rede que possuem a capacidade de aproximar qualquer mapeamento não-linear arbitrário, como por exemplo as redes MLP e as redes do tipo RBF [Nar96], [HSZG92]. Em controle, o interesse na capacidade da RNA realizar mapeamentos, pode ser visivelmente observada na área de identificação e modelagem de processos.

Do ponto de vista matemático, não existe uma vantagem óbvia das RNA's em relação aos outros métodos (métodos polinomiais, séries trigonométricas, funções ortogonais) empregados para modelagem de sistemas [HSZG92]. Entretanto, por apresentarem uma robustez maior à ruídos, por serem tolerantes a falhas e mais fáceis de ser implementadas em *hardware*, as RNA's vem sendo empregadas cada vez mais na modelagem e identificação de processos.

Necessidade de conhecimento mínimo do processo

A vantagem dessa característica reside no fato que em técnicas de controle convencionais, faz-se necessário o conhecimento interno (mais detalhado) do processo em questão. Através de uma RNA é possível levantar um modelo, sem necessariamente conhecer o seu funcionamento "interno", ou seja, sem o conhecimento implícito do

comportamento dinâmico do processo. Nesse caso, as RNA's necessitam somente do conhecimento "externo", como as características e comportamento dos sinais de entrada e saída, faixa de atuação desses sinais; regime e condições de funcionamento; conhecimento qualificado das influências da entrada sobre a saída.

Esse tipo de abordagem se torna possível, uma vez em que a RNA pode aprender o comportamento do sistema, baseada somente nos sinais de entrada e saída da planta, que vão se constituir dos exemplos para o treinamento da rede [Tor92].

Controle não-linear

Uma razoável parte dos processos existentes pode ser tratado como um sistema linear trabalhando em um ou vários pontos de operação. Em torno de cada um desses pontos de operação, existe um modelo linearizado que descreve razoavelmente bem o comportamento dinâmico do sistema. Baseada no teoria de sistemas lineares, foi possível, então, desenvolver um conjunto poderoso de técnicas para análise e projeto de sistemas de controle.

Com o aumento da complexidade do sistema (pouca informação a respeito do processo, modelagem complexa, sensores e atuadores distribuídos, alto nível de ruído, sistemas multivariáveis), o emprego de técnicas ditas convencionais de controle não conseguem atender satisfatoriamente, em muitos casos, as especificações desejadas para o sistema [Nar96]. Em grande parte, isso se deve ao fato do sistema apresentar um conjunto de não-linearidades que não permite a utilização de técnicas lineares de controle.

Entretanto, não existe uma teoria geral e sistemática para sistemas não-lineares. Além disso, nem sempre é possível obter uma representação matemática única de um sistema não-linear, sendo necessária uma análise individual de cada caso. Dessa forma, a capacidade de realizar mapeamentos não-lineares complexos que as RNA's apresentam, vem impulsionando o seu emprego na síntese de controladores em sistemas não-lineares.

Entretanto, não se justifica o emprego de RNA's em sistemas onde as técnicas convencionais apresentam uma performance satisfatória. O emprego de tal técnica representaria uma perda de tempo, onde o aprendizado de uma nova técnica não agregará muito valor ao processo [Aga97] e [Chi97].

Sistemas multivariáveis

Devido a flexibilidade na sua estrutura, com múltiplas entradas e saídas, as RNA's podem ser facilmente empregadas em sistemas multivariáveis.

Eliminação natural de ruídos

Uma vez treinada a rede fica insensível a pequenas variações na entrada [FS93]. Essa característica torna-se importante uma vez que em muitos sistemas de controle, os sinais são medidos contendo uma certa parcela de ruídos.

3.3.2 Estruturas de Controladores Neurais

O emprego de RNA's em controle é bastante variado. Uma rede neural pode assumir diversos papéis em uma estrutura de controle [Chi97]. Ela pode ser empregada como um elemento auxiliar ao controlador da planta, como por exemplo atuar somente como um o modelo de estimação para o processo em questão(estimador ou identificador), ou atuando como um controlador propriamente dito.

O papel que a rede vai assumir na malha de controle vai depender de uma série de fatores [Aga97]:

Conhecimento disponível do sistema. A informação disponível do sistema é suficiente para síntese de controladores mais simples, ou é necessária a identificação ou modelagem da planta? É necessário o levantamento total do modelo do sistema, ou somente alguns de seus parâmetros?

Faixa de operação. O sistema opera em diferentes pontos de operação? A sua dinâmica modifica consideravelmente nesses diferentes pontos de operação (não-linearidades)?

Características não-lineares. As não-linearidades existentes no processo inviabilizam o emprego de técnicas mais simples de controle (controladores convencionais)?

Incremento da performance. A melhora no desempenho do sistema justifica o estudo e emprego de técnicas mais avançadas de controle?

Se pudermos responder afirmativamente as questões acimas, então o emprego de uma RNA fica bem justificado.

Na parte de estimação e modelagem uma rede neural pode desempenhar diversas funções. Uma rede neural pode ser empregada para mapear a *dinâmica direta* da planta. Dessa forma, a rede assim treinada é capaz de reproduzir a relação entrada/saída do processo mapeado.

Outra possibilidade de utilização de RNA's em modelagem, consiste em buscar mapear a relação inversa entre entrada e saída do processo, conhecida como *dinâmica*

inversa. Uma RNA que mapeia o inverso da planta pode ser utilizada em estruturas de controle do tipo *controle inverso*, [Aga97], [Nar96] e [FS92].

Além dessas possibilidades, uma RNA pode ser utilizada para prever valores futuros da saída do processo (controle preditivo); para estimar os parâmetros do sistema (nesse caso conhece-se a ordem das equações dinâmicas que descrevem o modelo); ou ainda, como no caso do mapeamento da dinâmica direta, utilizar os pares de entrada/saída da planta para mapear a dinâmica completa da planta, sem se preocupar com a sua dinâmica interna [Aga97] e [HSZG92].

Na literatura de controle neural, algumas das estratégias mais freqüentemente encontradas são [Aga97], [Krö94], [Nar96], [FS92] e [Tor92]:

Controle Supervisionado. A rede neural é treinada de forma a imitar as ações de um mecanismo (operador humano ou controlador) atuando sobre o processo.

Controle Direto Inverso. Uma RNA no papel de controlador da planta, é treinada para mapear a dinâmica inversa da planta, de forma que o produto em cascata com o processo seja próximo a um, em todas as faixas de operação.

Controle Inverso *Feedforward*. São empregadas duas redes neurais. A primeira rede, em paralelo com a planta, é treinada de forma *off-line* com a dinâmica direta do processo. A segunda RNA, na malha direta com a planta, assume o papel de controlador, sendo treinada com dinâmica inversa, ao modo do controlador do item anterior.

Controle pelo Modelo Interno. Nesta estrutura um modelo inverso e um modelo direto são empregados. O modelo direto é colocado em paralelo com a planta, e a diferença entre a saída da planta e da rede é realimentada. Esse sinal é processado por um filtro e introduzido na rede que desempenha a dinâmica inversa (rede controladora). Essa implementação é limitada em sistemas estáveis em malha-aberta.

Controle Adaptativo. Emprego de uma RNA para sintonia de um controlador adaptativo, ou fornecendo os parâmetros estimados utilizados pela lei de controle.

Controle Preditivo. Neste método uma RNA fornece os valores futuros da planta dentro de um horizonte especificado. Esse sinal é passado ao controlador, que pode ser uma RNA ou não, que busca minimizar uma determinada função custo.

Além das estratégias acima citadas, pode-se utilizar RNA em conjunto com outros métodos de controle, como controle ótimo, estrutura variável, controle por alocação de pólos, controle *fuzzy-neural* [dBBB98], *self-tunning* PID, dentre outras [Aga97].

A utilização de RNA em controle de processos, entretanto, não se constitui de uma tarefa trivial. A escolha da melhor estrutura de controle, do papel da RNA dentro da lei de controle, da definição da própria estrutura da rede são questões que devem ser observadas atentamente, pois um bom desempenho depende essencialmente desses quesitos.

Quando as RNA's são utilizadas em identificação de processos, ou em controle *off-line* (onde a rede é treinada, em primeira instância, fora da planta), são empregados conjuntos de exemplos, constituídos dos sinais de entrada e saída da própria planta, utilizados para o treinamento da rede. Entretanto, a escolha desse conjunto de exemplos deve atender a determinados critérios de forma a que a aproximação do processo obtida pela rede seja fiel ao processo propriamente dito. Deve-se observar a coerência dos padrões com o comportamento normal do sistema, e a necessidade de distribuição uniforme das várias entradas e saídas do domínio de atuação do sistema, abrangendo toda a faixa de operação e funcionamento do processo [Kli96] e [FS91].

3.4 Conclusão

O emprego de estruturas das quais, aparentemente, emerge um comportamento inteligente é um dos objetivos do estudo da IA Conexionista. Tais estruturas, tendo como modelo a rede sináptica formada pelo cérebro humano, vem encontrando seu espaço nas mais diversas áreas, abrangendo aplicações como o reconhecimento de padrões, sistemas de controle, previsão de falhas, dentre outras. A essas estruturas nomearam-se *Redes Neurais Artificiais* - RNA.

Na área de controle, as RNA's encontraram seu lugar principalmente devido a capacidade que estas apresentam de realizar mapeamentos não-lineares arbitrários e de aprender através de exemplos. As propriedades de paralelismo e rejeição de ruídos das RNA's também as tornam bastante atrativas para o seu emprego em controle.

O projeto de uma rede neural em controle requer que o usuário decida sobre o tipo de estratégia de controle a ser empregada, e também sobre a topologia que a rede vai apresentar. Tipo de rede, número de camadas, tipo de função de ativação, controle direto ou indireto, rede atuando como estimador ou como controlador, etc., são questões que devem ser resolvidas pelo projetista do sistema de controle, baseando-se mais em conhecimentos empíricos e heurísticos, do que em uma teoria formalmente bem definida. Além disso, a falta de critério na utilização das RNA's pode levar a uma subutilização das suas reais capacidades.

Neste trabalho, a técnica de Redes Neurais Artificiais será adotada para o controle de robôs manipuladores, como será visto no próximo capítulo.

Capítulo 4

Controle Neural de Robôs Manipuladores

4.1 Introdução

Em tarefas onde os principais parâmetros do manipulador são conhecidos, onde o comportamento dinâmico do manipulador não se encontra sujeito à perturbações e efeitos do tipo flexibilidade, e onde desenvolvem-se tarefas relativamente simples, como funções do tipo *pick-and-place*, controladores clássicos com ganhos fixos são suficientes. Entretanto nas aplicações em que são exigidas seguimento de trajetória contínua, altas velocidades de deslocamento (influência mais significativa dos termos não-lineares de gravitação, inércia, coriolis e torques centrífugos) ou ainda, manipulação de diferentes cargas, tais controladores não apresentam um desempenho satisfatório. Nesse caso, torna-se necessária a utilização de estruturas mais complexas de controle, exigindo um conhecimento mais aprofundado a respeito do comportamento dinâmico do robô manipulador [KV96], [ZZ96], [ZM96].

A grande dificuldade, entretanto, consiste em levantar os valores reais dos parâmetros da equação que descreve a dinâmica do manipulador e que está sujeita às incertezas, alterando seus valores de acordo com as condições do ambiente de trabalho. Além disso, tal equação é uma simplificação do modelo real, desprezando condições que influenciam o comportamento do manipulador, como a flexibilidade existente nas transmissões, atrito nas juntas, perturbações e ruídos externos, dentre outros. No caso do controle de força, ainda existe o problema do entendimento da interação existente entre o meio e o efetuador final, e que ainda não se encontra suficientemente conhecida.

As dificuldades inerentes ao processo de modelagem e identificação do comportamento do manipulador, bem como da especificação e projeto da estrutura de controle

são [IFOU92], [LAD93], [dWSB96], [KV96], [ZZ96], [EL97] e [JH98]:

- variação nos parâmetros
- dinâmica complexa
- vários modelos propostos
- sistema multivariável, não-linear e acoplado
- dificuldade de medir os coeficientes não-lineares de atrito
- dificuldade de medir a inércia dos elos
- flexibilidade nas juntas
- dinâmica do atuador variável
- contatos entre efetuador final e ambiente de trabalho
- folgas e zonas-mortas nos atuadores
- cargas diferentes e desconhecidas
- impactos e colisões
- simplificações no modelo
- modos de alta-freqüência
- atrasos de transporte negligenciados

Como alternativa aos métodos clássicos e baseados no conhecimento preciso do modelo, estruturas de controle que apresentam um comportamento adaptativo são extremamente interessantes em sistemas em que se dispõe de pouca informação a seu respeito, e cuja dinâmica é variável com o tempo. Os controladores adaptativos clássicos e os controladores neurais apresentam essa característica, e vem sendo empregados em controle de posição, velocidade e força de robôs manipuladores.

4.2 Redes Neurais Artificiais em Controle de Robôs

Os controladores clássicos baseados em modelos (aqueles que incluem termos da equação do manipulador em sua lei de controle) podem não apresentar uma solução razoável, pois encontram-se baseados em um modelo com informação incompleta, com conhecimento parcial ou impreciso dos parâmetros do manipulador. Estes algoritmos também são extremamente sensíveis à falta de informação nos sensores, eventos não planejados (como impactos) e situações não familiares no ambiente de trabalho [KV96].

Buscando contornar a pouca informação disponível sobre o sistema e as variações paramétricas que o mesmo apresenta, torna-se necessário utilizar métodos de controle que sejam eficientes sob essas condições. Os controladores robustos e adaptativos apresentam essas características [ZM96]. As capacidades de adaptação e aprendizado que os controladores adaptativos e neurais possuem, motivam a sua utilização para o controle de manipuladores robóticos.

Entretanto, os algoritmos de controle adaptativo, como o STR (*Self-Tuning Regulator*) e o MRAC (*Model Reference Adaptive Control*), duas das estruturas de controle adaptativo mais utilizadas, necessitam de uma especificação explícita do modelo ou da estrutura da planta. Apesar dos controladores adaptativos compensarem bem as *incertezas estruturais* — incertezas nos parâmetros devido a imprecisão nas propriedades do elo do manipulador, cargas desconhecidas, imprecisão nos torques, etc. — ainda não está bem claro se eles conseguem resolver sozinhos as *incertezas não-estruturais* — provenientes da presença dos modos de alta-freqüência, atrasos de transporte negligenciados, atritos não-lineares, etc. [EL97], [KV96] e [ZM96].

Os algoritmos adaptativos e neurais baseiam-se nas informações obtidas do processo para adaptarem seus parâmetros. Entretanto, seus objetivos são bem diversos. Os primeiros estão baseados na atualização do comportamento dinâmico durante a fase de funcionamento do sistema, apresentando um comportamento *adaptativo*. Os controladores neurais, por sua vez, desenvolvem um modelo global, associando comportamentos com situações — processo denominado de *aprendizado* [ZM96].

Um sistema que trata cada situação distinta de operação como sendo uma situação nova, está limitado a um comportamento adaptativo, enquanto que um sistema que correlaciona experiências passadas com situações passadas, e que pode evocar e explorar tais situações em sua memória, é capaz de aprender. Além disso, uma vez que um sistema com capacidade de aprendizado é capaz de ajustar sua memória para acomodar novas experiências, ele também não deixa de apresentar um comportamento adaptativo. Por essa razão, os controladores neurais apresentam uma vantagem no que tange ao aspecto da não necessidade do conhecimento do funcionamento interno do sistema

[ZM96].

Na área de robótica, as redes neurais artificiais vem sendo empregadas para resolver problemas de planejamento de tarefas; planejamento e controle de trajetória; controle de contatos; coordenação, manipulação, locomoção e navegação; sensoreamento e percepção; dentre outros [KV96], [Nar96] e [FS92].

Em controle de robôs, quatro tipos de estruturas de redes vem sendo empregadas com mais sucesso: redes MLP (*Multilayered Perceptrons*); redes RBF (*Radial Basis Function*; redes CMAC (*Cerebellar Model Articulation Controller*); e as redes de *Hopfield* e as variações recorrentes das redes MLP [KV96].

As redes CMAC apresentam propriedades de generalização local, enquanto que as redes MLP apresentam uma capacidade de generalização global, estando as redes RBF entre essas. As estruturas que tem uma capacidade de aprendizado local apresentam uma velocidade de treinamento ou aprendizagem maior, entretanto não conseguem generalizar o aprendizado tão bem quanto uma rede com capacidade de aprendizagem global [KV96].

Existem diversas formas de utilizar as capacidades de aproximação, generalização e paralelismo de uma RNA em uma estrutura de controle. Uma RNA pode ser empregada para mapear a dinâmica inversa do manipulador; para atuar em paralelo com uma estrutura de controle clássico linear, compensando as não-linearidades do robô; como um estimador de parâmetros para um controlador adaptativo ou preditivo; ou ainda, aprendendo a dinâmica do meio no caso do controle de força (impactos e colisões) [KV96], [ZZ96] e [FS92].

Uma rede neural pode ser utilizada para aprender a cinemática inversa, cuja resolução analítica é relativamente complexa. A vantagem da utilização de RNA's para o mapeamento da dinâmica do manipulador (direta e inversa) é a capacidade de aprender essa dinâmica, incluindo as incertezas, sem a necessidade da modelagem analítica, bastando somente um conjunto de vetores de treinamento (exemplos) obtidos do processo real.

Na área de robótica, uma rede neural pode ser treinada de várias formas. Muitas vezes, o processo de treinamento está intimamente relacionado com o tipo de estrutura de controle a ser empregada [FS92] e [KV96]. Três são as mais empregadas:

Aprendizado Generalizado: Inicialmente a RNA é treinada fora do processo - treinamento *off-line*, ou seja, através de um conjunto de exemplos obtidos da planta, a rede é treinada de forma a mapear a sua relação entrada/saída. Posteriormente a rede é colocada em uma estrutura de controle, atuando como modelo (dinâmica direta), ou como controlador (dinâmica inversa, controle *feddforward*), sendo re-

treinada de forma *on-line*. Entretanto, a formação do conjunto de exemplos deve seguir uma série de critérios, buscando abranger toda a faixa de operação do manipulador (várias cargas acopladas, diferentes trajetórias) de forma a permitir que a rede consiga generalizar razoavelmente o aprendizado para situações para as quais não foi treinada.

Aprendizado Especializado: A RNA é colocada como controlador do processo. As fases de aprendizado e controle são simultâneas, ou seja, a rede aprende a controlar o processo, atuando diretamente sobre o mesmo em tempo real - treinamento *on-line*. Para ajustes dos pesos, torna-se necessário o conhecimento do Jacobiano da planta (variação da saída em relação à entrada), que pode ser calculado analiticamente, ou utilizando um modelo ou estimador para o planta, empregando-se, por exemplo, uma segunda RNA. O problema desse tipo de estrutura reside na possibilidade de instabilização do sistema nas fases iniciais do treinamento. Outro ponto que dificulta a sua utilização é o conhecimento analítico do Jacobiano da planta. Caso seja empregada uma segunda RNA (controle inverso *feedforward*), é necessário treiná-la de forma *off-line*.

***Feedback Error Learning*:** A RNA é colocada em paralelo com uma estrutura de controle *feedback*. Dessa forma, a rede pode atuar como um controlador *feedforward*, ou como um compensador das não-linearidades da planta. O sinal de saída do controlador *feedback* é utilizado como sinal de treinamento para a rede neural. A vantagem dessa estrutura é que ela não necessita de um conjunto de treinamentos *off-line* ou do conhecimento do Jacobiano do manipulador. Além disso, o controlador *feedforward* garante a estabilidade nas fases iniciais do treinamento. É um método exclusivamente *on-line*. Posteriormente, o controlador *feedback* pode ser retirado, ficando a rede neural encarregada do controle do processo. A desvantagem consiste na necessidade do projeto do controlador *feedback*, de forma que a inclusão da RNA na malha de controle não prejudique o desempenho do mesmo (a RNA não pode ser vista como uma perturbação pelo controlador).

Em estruturas de treinamento *off-line*, o conjunto RNA-controlador apresenta uma boa performance de controle para aquelas trajetórias implicitamente contidas no conjunto de treinamento. Porém, o mesmo pode não acontecer para uma nova trajetória. Isso significa que o conjunto de treinamento utilizado não era suficientemente completo, existindo situações não previstas e que a capacidade de generalização da rede não é suficiente para garantir um bom desempenho. Outra possível causa desse problema,

consiste no fato da RNA realizar somente um mapeamento entrada/saída dos dados, e não da dinâmica do manipulador propriamente dita, não sendo capaz de prever o comportamento do manipulador para situações não-previstas no conjunto de exemplos.

Para o mapeamento da dinâmica inversa de robôs com mais de três juntas, a quantidade de dados que formam o conjunto de exemplos é muito grande, tornando o treinamento da rede um processo muito custoso e lento. Um mapeamento geral exige um conjunto amplo de valores, tornando a estrutura das redes mais complexa [ZM96], [ZZ96]. Além disso, as equações do robô manipulador são não-lineares, e o sistema é multivariável, o que dificulta mais ainda o processo de aprendizagem da inversa da planta. Por essa razão, as RNA's tem assumido mais um papel auxiliar no controle de robôs, do que efetivamente tentar mapear toda a dinâmica do manipulador (como a estrutura *Feedback Error Learning*), ou atuar como controlador principal. Essa característica se deve ao fato de que as RNA's aparentam ter uma performance melhor quando não necessitam aprender ou guardar muita informação a respeito do processo [IFOU92].

4.3 Controle de Posição de Robôs Manipuladores

Conforme exposto na seção 2.4.1, o controle de posição consiste em seguir uma referência de posição e velocidade para as juntas, de forma que o efetuador-final do manipulador execute uma determinada trajetória. A lei de controle encarrega-se de encontrar os torques adequados para cada junta, de forma a garantir as especificações de controle (tempo de resposta, rejeição à perturbação, robustez à ruídos, valores máximos de picos, etc.). A utilização da cinemática inversa se faz necessária para mapear a trajetória de referência no espaço cartesiano do efetuador final para o espaço das juntas.

O problema de controle de posição envolve a parte de *regulação* e *controle de trajetória*. O problema de regulação, consiste em manter uma posição fixa no espaço de juntas, independente das perturbações atuando sobre o manipulador.

No controle de trajetória, deseja-se que o robô acompanhe uma trajetória de referência para posição e velocidade, variante no tempo. A lei de controle deve ser capaz de atender as especificações de controle durante toda a trajetória.

Serão apresentados três estruturas de controle de posição de robôs manipuladores, no espaço de juntas[1]: o controle convencional PD, a estrutura de controle inverso e

[1]Nesse trabalho assume-se a configuração do controle cinemático, empregando as equações dinâmicas do manipulador no espaço de juntas. As estruturas de controle também são especificadas no espaço de juntas. Também assume-se que a referência já está sendo especificada no espaço de juntas, não se preocupando com a complexidade da cinemática inversa.

uma proposta de controle neural do tipo *Feedback Error Learning.*

4.3.1 Controle Clássico de Posição

Controle PD/PID

Seja a equação dinâmica do manipulador definida pela equação 2.21:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{4.1}$$

onde $q \in Re^{nx1}$ é o vetor de coordenadas generalizadas das juntas, $H(q) \in \Re^{nxn}$ é a matriz de inércia, $C(q,\dot{q}) \in \Re^{nx1}$ é o vetor de forças e momentos centrífugos e de Coriolis, $G(q) \in \Re^{nx1}$ é o vetor de forças gravitacionais, e $\tau \in \Re^{nx1}$ é o vetor de torque de controle sobre as juntas.

Dois dos controladores mais simples, e que foram empregados inicialmente para o controle no espaço de juntas, foram os controladores clássicos PD e PID - figura 4.1[2]. Através de uma lei de controle relativamente simples 4.2 e 4.3 é possível atender os requisitos de controle:

$$\tau = K_p(q_d - q) - K_d\dot{q} \tag{4.2}$$

correspondendo a um controlador analógico do tipo PD, e

$$\tau = -K_d\dot{q} + K_p(q_d - q) + K_i\int_0^t (q_d - q)\partial t \tag{4.3}$$

caracterizando uma estrutura de controle, também analógica, do tipo PID. Os termos $K_{(.)}$ correspondem as matrizes de ganho dos controladores, da ordem $\in \Re^{nxn}$, positivas e geralmente diagonais, e q_d é o valor de referência para o deslocamento nas juntas q.

Apesar do controlador PD não conseguir resolver o problema de regulação, apresentando valores de erros em regime permanente, este controlador garante estabilidade global ao manipulador. O controlador PID, por sua vez, é localmente estável, resolve o problema de regulação mas não apresenta uma uniformidade no regime transitório [dWSB96].

Uma alternativa muito empregada, que melhora o desempenho do controlador PD, consiste em incluir um termo de compensação (geralmente de gravidade, e mais raramente, de atrito). Entretanto para obter um desempenho razoável, é necessário o conhecimento exato do termo $G(q)$ da equação 4.1, que é incluída na equação 4.2 [dWSB96], [LAD93].

[2]D/A e A/D referem-se, respectivamente, aos conversores digital/analógico e analógico/digital.

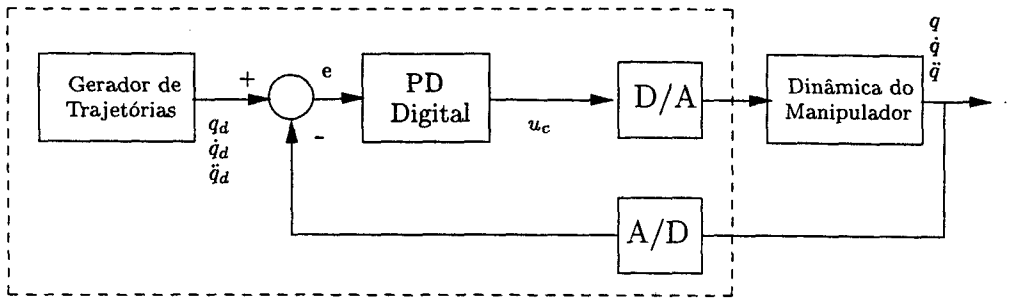

Figura 4.1: Controle PD Digital de Posição

Controle Dinâmico Inverso

Outra possibilidade, muito conhecida, consiste em utilizar o próprio modelo do manipulador para tornar o sistema manipulador + controlador, um conjunto de n duplos integradores, desacoplados e lineares. Esse método é conhecido como *controle inverso*, *controle da dinâmica inversa*, *controle linearizante*, ou ainda *método de controle via torque-computado* - figura 4.2. O controle inverso consiste em utilizar o modelo da planta para o seu controle. Entretanto essa estrutura apresenta uma configuração em malha aberta, além de não garantir um bom funcionamento em função das limitações e imprecisões no modelo adotado. Dessa forma, torna-se necessário utilizar uma outra estratégia de controle, em malha fechada, para garantir um melhor desempenho de controle[3].

A lei de controle da dinâmica inversa é dada pela expressão 4.4:

$$\tau = \hat{H}(q)u_0 + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) \tag{4.4}$$

onde $\hat{H}$, $\hat{C}$ e $\hat{G}$ são as matrizes estimadas do modelo dinâmico e u_0 é um sinal de controle. Para gerar o sinal u_0, tipicamente são utilizados controladores do tipo PD ou PID.

Substituindo a lei de controle 4.4 na equação 4.1, considerando que os valores estimados sejam idênticos aos reais, e que também não existam influências de perturbações externas e ruídos atuando sobre a dinâmica do manipulador, a equação em malha fechada resulta em

$$H(q)(\ddot{q} - u_0) = 0. \tag{4.5}$$

[3]Nesse trabalho adota-se a terminologia *controle inverso* para a lei de controle compreendida pelo modelo da planta mais o controlador PD, PI ou PID.

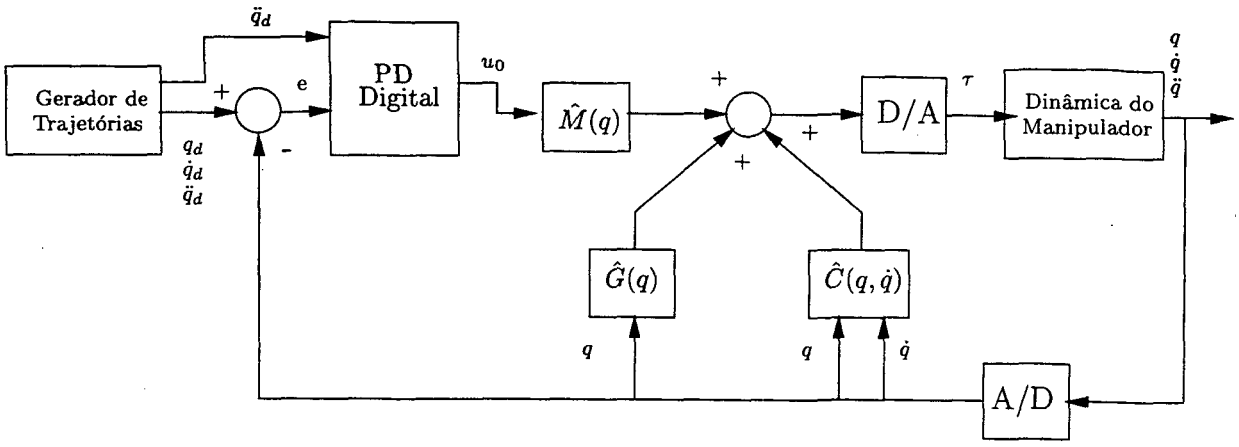

Figura 4.2: Controle Inverso de Posição

Como $H(q)$ é definida positiva, a inversa da matriz de inércia existe, e portanto

$$\ddot{q} = u_0 \tag{4.6}$$

Como u_0 é o sinal proveniente de um controlador PD ou PID

$$u_0 = \ddot{q}_d + K_d(\dot{q} - \dot{q}) + K_p(q_d - q) \tag{4.7}$$

$$u_0 = \ddot{q}_d + K_d(\dot{q} - \dot{q}) + K_p(q_d - q) + K_i \int_0^t (q_d - q)\partial t \tag{4.8}$$

pode-se provar que o sistema é assintoticamente estável [Mar97], [dWSB96].

Observa-se que o controle inverso apresenta a inconveniência de exigir o exato conhecimento da equação dinâmica do manipulador. Caso os termos estimados não correspondam aos valores reais, a lei de controle pode não ser capaz de seguir a referência de posição com erro nulo.

Algumas estratégias de controle buscam melhorar a performance da estrutura de controle inverso, empregado técnicas adaptativas para estimação dos parâmetros do manipulador, como é o caso dos controladores adaptativos [Mar97], ou das redes neurais artificiais [IFOU92], [SdPB98] e [NG97].

4.3.2 Controle Não-Clássico de Posição

Outro conjunto de controladores estão baseados nas propriedades de passividade, ou ainda, são projetados de forma a garantir os critérios de estabilidades segundo Lyapunov [dWSB96]. A vantagem desses últimos métodos sobre o controle inverso,

reside no fato de ambos não estarem baseados na linearização e desacoplamento da equação dinâmica do manipulador, apresentando uma robustez maior.

Também existem os métodos de controle robusto, que buscam definir as leis de controle baseado no conhecimento dos limites das variações paramétricas e das perturbações que existem no sistema, e os métodos de controle adaptativo, que conseguem garantir o seguimento de trajetória mesmo sem a necessidade do conhecimento dos parâmetros do processo. Em [Mar97] podem ser verificadas provas de estabilidade para o emprego de controladores adaptativos e controladores baseados em passividade em controle de posição de robôs manipuladores.

Os controladores adaptativos e neurais podem ser empregados para suprir essa deficiência do controle inverso: a falta de conhecimento a respeito dos parâmetros do manipulador. Como discutido na seção 4.2, os controladores adaptativos, entretanto, necessitam do conhecimento de um modelo paramétrico para a planta. Esse problema pode ser contornado com o emprego de estruturas de controle utilizando redes neurais, uma vez que essas não necessitam do conhecimento implícito do processo.

Controle Neural de Posição

O emprego de redes neurais para controle de posição em robôs manipuladores já é um assunto bem explorado, podendo ser encontradas várias estruturas de controle na literatura. Abrangem desde as arquiteturas que utilizam uma RNA para realizar o mapeamento completo da dinâmica do manipulador, como o controle inverso, até estratégias mais simples, e muitas vezes mais eficientes, que empregam uma rede para compensar as não-linearidades ou como um estimador de parâmetros para uma estrutura de controle adaptativa ou baseada em passividade [Mar97], [Fla97], [KSS93], [EL97] e [ZM96].

Seja a equação 4.1 que descreve a dinâmica do manipulador. Como exposto na seção anterior, um controlador clássico tipo PID não consegue atender as especificações de malha-fechada para o seguimento de trajetória, quando existe incertezas paramétricas e perturbações externas. O método do torque computado (equação 4.4) consegue garantir uma boa resposta, porém, necessita do exato conhecimento do modelo do manipulador. Uma alternativa, seria empregar uma RNA para mapear a dinâmica do manipulador, de forma que a dinâmica estimada seja a mais próxima possível dos valores reais.

Existem várias formas de se realizar essa estimação. [ZM96] emprega uma estrutura complexa, para mapear cada um dos termos da equação do manipulador, utilizando para isso uma rede para cada matriz da equação 4.1. Entretanto, a estrutura de rede obtida é extremamente complexa.

[EL97] emprega o modelo *Feedback Error Learning*, visto na seção 4.2, com a rede fazendo o papel de um controlador *feedforward*, cancelando as não-linearidades nominais da planta em regime permanente. Entretanto, essa estrutura necessita de uma fase de treinamento *off-line*. Em [SdPB98] também é utilizada uma estrutura do tipo *Feeback Error Learning*. A rede inicialmente é treinada *off-line* com a dinâmica direta do manipulador, e em seguida é colocada na estrutura de controle. A RNA continua sendo treinada em *background*, e atualizada periodicamente.

[NG97] propõem uma alternativa ao mapeamento completo da dinâmica do manipulador. São empregadas duas redes, uma para mapear a matriz de inércia $H(q)$ e outra para mapear os termos de coriolis, centrífugos e gravitacionais $C(q,\dot{q})$ e $G(q)$. E em [SdPB97], comenta-se da utilização de uma rede neural treinada em cascata com um controlador clássico (PD ou PID), também realizando o mapeamento da dinâmica inversa (treinada *off-line*).

O emprego de uma estrutura que necessita de treinamento *off-line*, ou baseada no *aprendizado generalizado*, necessita do levantamento do conjunto de exemplos. O conjunto de exemplos deve abranger todo os pontos de funcionamento do manipulador, de forma a permitir uma capacidade de generalização razoável (tanto interpolação como extrapolação). Além disso, são necessários muitos dados para formar esse conjunto de exemplos.

Para um robô com n graus de liberdade, são necessárias n valores de posição, velocidade e aceleração por instante de tempo. Como o robô apresenta uma dinâmica relativamente rápida (da ordem de milisegundos), então, para uma trajetória da ordem de segundos, dentre as várias possíveis, são necessários muitos pontos. Isso prejudica não só a escolha adequada de um conjunto de exemplos, mas também aumenta a complexidade da rede, além de influir no tempo de treinamento da mesma.

Muitas das alternativas encontradas na literatura empregam uma rede neural como um estimador para o modelo da planta. Dessa forma, necessitam de uma quantidade muito grande de dados para realmente aprender a dinâmica do manipulador, e não somente fazer um mapeamento entre as entradas e saídas do mesmo [ZM96]. Para poder contornar o problema do treinamento *off-line* quando o conjunto de dados é muito grande, pode-se buscar utilizar as RNA's de uma forma alternativa. Em [Fla97] o autor propõe que o erro a ser minimizado pela rede seja composto pelo erro entre a saída desejada e a saída da rede e também por uma parcela que inclui a ação de controle. Dessa forma, a rede irá buscar minimizar tanto o erro de aproximação da planta e o esforço de controle. [ZM96] apresenta uma proposta baseada em minimização da ação do controlador. Gradualmente, uma RNA vai assumindo o papel de um controlador clássico utilizado para estabilizar o sistema nos momentos iniciais de

operação. [IFOU92] propõe uma rede para compensar as imprecisões em um modelo paramétrico fixo. Entretanto a carga computacional exigida por essa proposta é relativamente grande.

Em estruturas em que a rede neural assume um papel mais de auxiliar ao controlador do manipulador, a complexidade na sua utilização diminui, dispensando, muitas vezes, o treinamento *off-line* [IFOU92]. A rede também não precisa ser retreinada para novas trajetórias, pois a função dela, não é mais aprender o comportamento global do manipulador frente a uma nova referência, mas sim complementar ou auxiliar a operação do controlador principal.

Estrutura de Controle Neural de Posição Proposta

A estrutura de controle neural de posição proposta nesse trabalho encontra-se baseada nas estruturas descritas em [ZM96] e [EL97], consistindo de uma variação do modelo *Feedback Error Learning*, apresentado na seção 4.2.

O sistema de controle procura gerar o sinal τ da equação 4.1, empregando uma RNA atuando em paralelo com um controlador convencional. A RNA é responsável por compensar as não-linearidades, perturbações e imprecisões existentes no modelo. A ação de controle principal é dada pelo controlador convencional. Dessa forma, a estabilidade nos instantes iniciais da execução da trajetória do robô é garantida, evitando um comportamento transitório indesejável nos momentos iniciais de atuação da rede neural. A figura 4.3 mostra a estrutura de controle proposta.

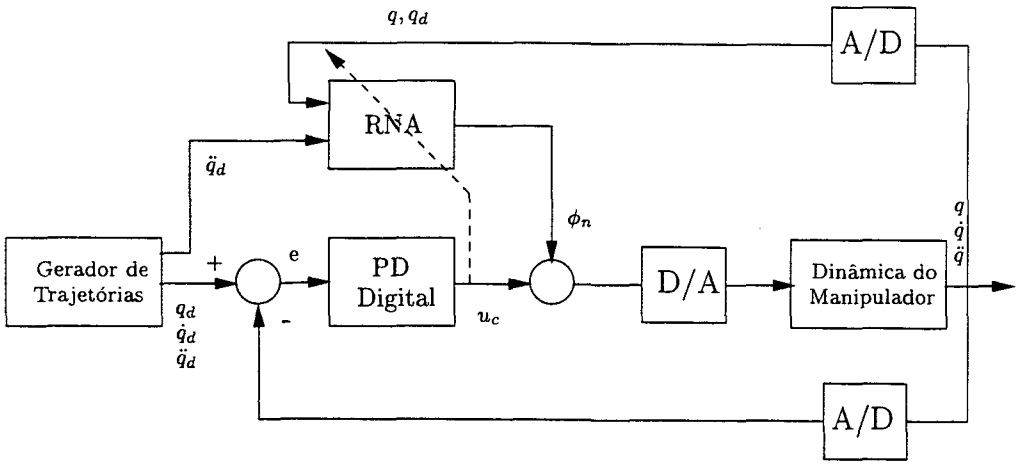

Figura 4.3: Proposta de Controle Neural de Posição

A lei de controle é dada por:

$$\tau = u_c + \phi_n \tag{4.9}$$

onde u_c é dado por um controlador digital do tipo PD. O sinal u_c é empregado para treinar a rede.

O projeto do controlador PD é feito diretamente no plano Z. O PD é projetado de forma a obter-se os melhores índices de performance possíveis, para diferentes pontos de operação, considerando as condições nominais do manipulador. Uma vez tendo-se realizado essa primeira etapa, inclui-se a rede neural, devidamente projetada (algoritmo de treinamento, número de camadas e neurônios, etc.). A rede atua diretamente no processo sem necessitar de uma etapa prévia de treinamento *off-line*.

A diferença dessa estrutura com as demais propostas [ZM96], [EL97] consiste no fato da mesma não necessitar de uma etapa de treinamento *off-line*. Além disso, a estrutura de controle considera somente uma rede neural, tratando todas as juntas em conjunto, ao contrário dos trabalhos de [ZM96] e [EL97], reduzindo o esforço computacional da lei de controle.

O papel da rede neural ϕ_n na estrutura de controle pode ser interpretada de várias formas, influenciando a escolha da topologia da rede, além de definir a estratégia de controle e os algoritmos de treinamento utilizados para treinamento da rede neural.

Caso seja utilizada uma estrutura do tipo *Feedback Error Learning*, a rede ϕ_n atuará como um controlador *feedforward*. Para tanto, ela pode ser treinada *off-line* de forma a compensar as não-linearidades nominais da planta [SdPB98], [EL97] e [ZM96].

Na estratégia de controle inverso, ou torque computado (equação 4.4), a dinâmica do manipulador pode ser substituída por uma rede neural que tenha sido treinada previamente com a dinâmica do manipulador [SdPB97].

A proposta desse trabalho consiste em utilizar o sinal de controle u_c para treinar a rede (figura 4.3). A rede interpreta esse sinal como sendo o erro de treinamento, e busca ajustar seus pesos de forma a minimizar o esforço de controle do controlador PD, fornecendo um valor de torque ϕ_n nas suas saídas (uma para cada junta). Dessa forma, obtém-se, indiretamente, a minimização do erro de posição e velocidade, uma vez que o controlador PD constitui-se da soma dos erros de posição e velocidade multiplicados por ganhos. Portanto, a saída do PD é nula se e somente se os erros de posição e velocidade também forem nulos.

Assim, a RNA atuando como indicado pela estrutura da figura 4.3 busca minimizar a ação de controle do PD, assumindo o papel de controlador em situações em que o PD por si só não consegue atuar, tais como compensação de gravidade, variações de carga, variações paramétricas e perturbações atuando sobre o manipulador.

A dificuldade maior nessa estratégia de controle, entretanto, consiste em provar a estabilidade do sistema robô + PD + RNA, sendo essa, talvez, uma das grandes desvantagens do seu emprego em controle de processos.

Porém, existem dois caminhos, que apontam para a solução do problema da análise de estabilidade de redes neurais em controle: análise da estrutura de controle considerando a rede neural como uma perturbação ou como uma matriz de realimentação de estados com ganhos não lineares.

Controladores do tipo PD apresentam uma forte propriedade no controle de posição de robôs manipuladores: garantia de estabilidade global para o sistema em malha fechada. Por sua vez, a saída da rede neural constitui-se de um sinal limitado no tempo, mesmo não ocorrendo a convergência do seu treinamento. Assim, a estrutura de controle neural de posição proposta, também apresenta a propriedade de estabilidade global, uma vez que, para efeitos de análise de estabilidade, o sinal da rede neural pode ser interpretado como uma perturbação com amplitude limitada, sendo ainda válidas as propriedades apresentadas pela estrutura clássica de controle PD. Dessa forma, a primeira proposta consiste, então, em analisar o desempenho da estrutura de controle, considerando a rede neural como uma perturbação.

Pode-se observar, também, que no esquema de controle neural de posição proposto, utiliza-se como entrada da rede os valores de posição, velocidade e aceleração nas juntas do manipulador. Dessa forma, a rede pode ser interpretada como uma matriz de realiamentação de estados com ganhos não-lineares. Ou seja, com o conjunto de entradas da rede acima definido, tem-se acesso a todo o espaço de estados do manipulador, sendo possível, assim, empregar uma rede neural estática, incorporando um comportamento dinâmico do processo em questão. Além disso, o estudo mais detalhado dessa "interpretação" para o comportamento da rede neural, aponta um possível caminho para uma teoria mais consistente e formal (menos "heurística"), como base para a área de estudo de redes neurais artificiais.

4.4 Controle Híbrido de Força/Posição

Conforme visto na seção 2.4.2, quando o efetuador final entra em contato com uma superfície, a dinâmica do robô passa a sofrer influência das forças e torques resultantes desse contato. Esse contato pode ser intencional (controle de força), ou não (tratamento de colisões). A equação 2.35 representa a dinâmica do manipulador sob essas condições. As principais estratégias de controle são desenvolvidas para o controle de força. O tratamento de colisões, apesar de ser um tópico de pesquisa importante, tem merecido menor atenção na literatura. Nesse trabalho a aplicação de técnicas de controle são desenvolvidas para o primeiro caso.

No controle híbrido, seção 2.4.2, o objetivo da lei de controle consiste em garantir que as referências de posição e força sejam seguidas. Para que tal requisito ocor-

ra, o controlador híbrido busca particionar o espaço de trabalho em dois subespaços: *subespaço de movimento livre* e *subespaço de movimento restrito.* No subespaço de movimento livre, o efetuador final pode se mover sem encontrar restrição ao seu movimento. Nesse subespaço faz-se o controle de posição. Nas direções em que o movimento é restrito, faz-se necessário controlar as forças e torques de forma a não danificar tanto a estrutura do manipulador, quanto a superfície em contato, além de procurar atingir as especificações de controle.

De acordo com [Yos90], o controle híbrido pode ser implementado de duas formas. A primeira, conhecida como *controle híbrido via compensação de realimentação*[4], baseia-se no conhecimento do jacobiano da superfície em relação ao espaço de juntas para compor a lei de controle. Esse jacobiano é responsável por mapear os erros de posição e força para o espaço de juntas. Através das matrizes de seleção, baseadas na geometria da superfície, faz-se a separação dos subespaços de controle de força e posição. São empregados, usualmente, controladores do tipo PD, PI e PID.

Uma outra possibilidade, de acordo com [Yos90], consiste em utilizar a dinâmica do manipulador na lei de controle - *controle híbrido dinâmico*[5]. As matrizes de seleção estão baseadas em hiperplanos que descrevem as restrições ao movimento do manipulador.

4.4.1 Controle Híbrido Dinâmico

Seja uma restrição ao movimento do efetuador-final, referenciada em relação a um sistemas de coordenadas r fixo, de seis graus de liberdade, dado pela equação:

$$p_i(r) = 0, \quad i = 1, 2, \ldots, m \tag{4.10}$$

com $m \leq 6$. As funções $p_i(r)$ devem ser duplamente diferenciáveis e mutuamente independentes em um subconjunto S do espaço dimensional R^6. Quando tem-se $p_i(r) > 0$ o manipulador entra em contato com a superfície interna do objeto, e $p_i(r) < 0$, quando for a superfície externa.

Diferenciando a equação 4.10, obtém-se:

$$E_F \dot{r} = 0 \tag{4.11}$$

onde

$$E_F = [e_{7-m}, e_{8-m}, \ldots, e_6]^T \tag{4.12}$$

[4]Do inglês *hybrid control via feedback compensation*

[5]O termo vem do inglês *Dynamic Hybrid Control.* Será mantido o termo em português, para diferenciar da primeira estratégia de controle híbrido proposto por [Yos90]

$$e_{6-m+i} = \frac{\partial p_i(r)}{\partial r}, \text{ com } i = 1, 2, \ldots, m. \tag{4.13}$$

Para expressar a trajetória do efetuador-final nas superfícies de restrição dadas pela equação 4.10, assume-se que existem $(6 - m)$ funções

$$y_P = s(r) = [s_1(r), s_2(r), \ldots, s_{6-m}(r)]^T \tag{4.14}$$

tal que $s_j(r)$, com $j = 1, 2, \ldots, 6 - m$, sejam funções duplamente diferenciáveis e que o conjunto $\{p_i(r), i = 1, 2, \ldots, m; s_j(r), j = 1, 2, \ldots, 6 - m\}$ sejam mutuamente independentes na região S. Definindo

$$E_P = [e_1, e_2, \ldots, e_{6-m}]^T \tag{4.15}$$

onde

$$e_j = \frac{\partial s_j(r)}{\partial r}, \; j = 1, 2, \ldots, 6 - m \tag{4.16}$$

e

$$\dot{y}_P = E_P \dot{r}. \tag{4.17}$$

é possível especificar as direções onde o efetuador pode se movimentar livremente, dado pelo conjunto $\{e_1, e_2, \ldots, e_{6-m}\}$, e onde o efetuador encontra-se sujeito às restrições $p_i(r)$, dado pelo conjunto $\{e_{7-m}, e_8, \ldots, e_6\}$.

A lei de controle híbrido para um manipulador sem redundância $(n \leq 6)$ é dada por [Yos90]:

$$\tau = \tau_P + \tau_F \tag{4.18}$$

$$\tau_P = \hat{H}(q)\ddot{q}_d + \hat{C}(q, \dot{q})\dot{q} + \hat{G}(q) \tag{4.19}$$

$$\ddot{q}_d = J^{-1}\left\{E^{-1}\left(\begin{bmatrix} u_1 \\ 0 \end{bmatrix} - \dot{E}J\dot{q}\right) - \dot{J}\dot{q}\right\} \tag{4.20}$$

$$\tau_F = J^T E_F^T u_2 \tag{4.21}$$

onde $E = [E_F \; E_P]^T$, e u_1 é um vetor de dimensão $(6 - m)$ e u_2 possui dimensão m. As matrizes $(\hat{\cdot})$ são as matrizes estimadas da equação dinâmica do manipulador.

Caso as matrizes sejam exatas, se $y_P(0) = 0$ e $\dot{y}_p(0) = \dot{y}_{Pd}(0)$, e se o sistema não sofrer a influência de perturbações externas, então as trajetórias desejadas de posição (y_{Pd}) e de força (f_{Fd}) serão obtidas através de

$$u_1 = \ddot{y}_d \text{ e } u_2 = f_{Fd}. \tag{4.22}$$

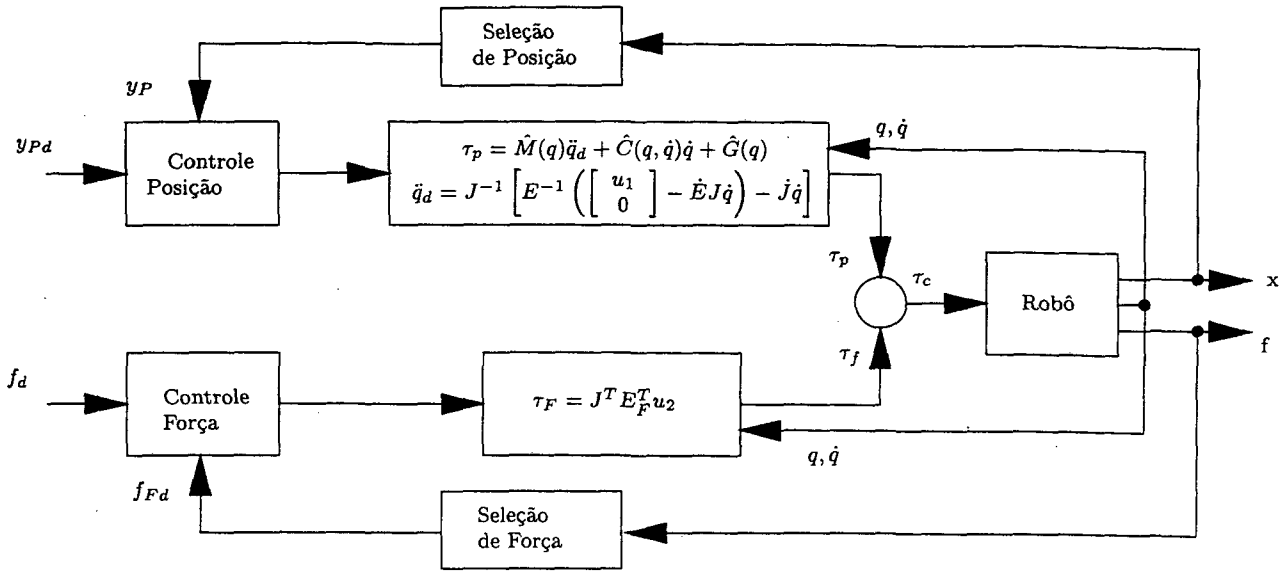

Figura 4.4: Controle Híbrido Dinâmico de Força/Posição

Entretanto, com o emprego da equação 4.22 para produzir os sinais de comando para o controlador híbrido, o sistema de controle fica em malha aberta. Uma performance de controle mais eficiente pode ser obtida, quando empregam-se controladores do tipo PID para produzir os sinais u_1 e u_2. Dessa forma, obtém-se uma lei de controle em malha-fechada, como pode ser vista na figura 4.4.

O valor de força f_F no subespaço de força é obtido através da relação

$$f_F = [0\ I_m]E^{-T}Q_\tau^T \tag{4.23}$$

com $v = Q_r\dot{r}$, onde Q_r é uma matriz de transformação $\in \Re^{6x6}$, que é função de r, v é a velocidade do efetuador final em relação a um sistema de coordenadas de dimensão 6 e $\dot{r}$ é a velocidade do efetuador-final em relação ao sistema de coordenadas empregado para descrever as restrições.

Em baixas velocidades as equações 4.18, 4.19 e 4.20 podem ser simplificadas, resultando em uma lei de controle mais simples:

$$\tau = H(q)J^{-1}E^{-1}\begin{pmatrix} u_1 \\ 0 \end{pmatrix} - J^T E_F^T u_2. \tag{4.24}$$

4.4.2 Controle Neural de Força/Posição

A área de controle de força constitui-se de um ramo da robótica onde várias pesquisas vem sendo realizadas e as aplicações industriais ainda encontram-se em estado

emergente. O estudo da interação entre sensor/efetuador-final e meio ainda é objetivo de muitas pesquisas [Wil92]. Dentre a bibliografia disponível sobre controle de força, existem poucas referências ao emprego de RNA´s na solução deste problema.

RNA's vem sendo utilizada como uma ferramenta poderosa no auxílio ao controle de força. Além das dificuldades também existentes no problema de controle de posição, como não-linearidades não modeladas, variações paramétricas e perturbações, o problema de controle de força apresenta uma dificuldade maior em função do contato do manipulador com uma superfície, geralmente, não conhecida e submodelada.

No controle de impedância, a necessidade de estabelecer um comportamento dinâmico entre o efetuador-final e a superfície de contato é prejudicada pela falta de conhecimento a respeito da superfície. Redes neurais artificiais vem sendo empregadas para se adaptarem às incertezas quanto à dinâmica do contato robô/meio, garantindo o comportamento dinâmico desejado [JH98] e [VGBT93]. Também são empregadas RNA's em situações de impacto ou colisões não planejadas (controle de contato) [Nar96], [FS92].

Em [GHW97] são empregas redes RBF para mapear cada elo da estrutura do manipulador. Esse mapeamento é feito diretamente no espaço da tarefa, evitando a necessidade das inversões do Jacobiano para o cálculo das leis de controle. Os resultados obtidos podem ser estendidos para o caso de força. Em [FSTM92] é descrito o emprego de uma rede neural para compensar os controladores clássicos em uma estrutura de controle híbrido de um robô de dois graus de liberdade.

RNA's também podem ser empregadas para estimar ou mapear a dinâmica do manipulador. Entretanto, tal estratégia demanda um certo tempo de treinamento *off-line* e depende da escolha adequada do conjunto de treinamento. Além disso, esse conjunto de exemplos deve incluir um número relativamente grande de possibilidades de forma a garantir a capacidade de generalização da rede neural.

Como muitas vezes a dinâmica do contato robô/meio não se encontra totalmente disponível, essa falta de informação prejudicará o desempenho da rede. Dessa forma, as forças envolvidas durante o contato, se não forem bem controladas, podem danificar a superfície do objeto em contato, ou até mesmo a própria estrutura do robô.

O emprego de uma rede neural como compensador dinâmico no caso de controle híbrido de forca/posição, parece ser mais indicado, uma vez que existe uma estrutura de controle (os próprios controladores clássicos da lei de controle híbrido, equações 4.18, 4.19, 4.20, 4.21) sendo capazes de evitar forças ou torques elevados. A rede atua somente como um compensador das dinâmicas não-modeladas, sem necessidade de treinamento *off-line*.

Nos próximos itens serão apresentadas duas novas estratégias de controle neural

para o problema de controle híbrido de força/posição, originalmente propostas nesse trabalho.

Controlador Híbrido Neural

A estrutura de controle híbrido neural, consiste em empregar duas redes neurais na estratégia de controle híbrido dinâmico apresentada na seção anterior. Uma das redes é empregada na malha de posição, enquanto que a outra é utilizada na malha de força.

A lei de controle híbrido encontra-se baseada nos valores nominais dos coeficientes do modelo dinâmico do manipulador - equações 4.18 à 4.21.

Incertezas, variações paramétricas e perturbações podem prejudicar o desempenho do controlador híbrido dinâmico. Com o emprego de uma estrutura adaptativa, como uma RNA, é possível aumentar a robustez do sistema de controle frente ao pouco conhecimento da estrutura dinâmica do manipulador. Dessa forma, a rede passa a funcionar como um compensador das não-linearidades, incertezas e perturbações não previstas pelo modelo utilizado na lei de controle híbrido (equação 4.18).

Os controladores clássicos comumente utilizados na malha de controle de força são do tipo PI ou PID. A proposta do esquema de controle híbrido neural é utilizar tanto na malha de posição, quanto na malha de força, um conjunto PD + RNA, de forma tal que a rede neural procure minimizar o sinal de comando do PD. A idéia principal reside no fato de que se a saída do controlador PD for nula, isso necessariamente implica em que os erros de posição e força sejam nulos. Dessa forma, as considerações e resultados obtidos no controle neural de posição, são estendidos para o caso de controle neural de força/posição.

Para a malha de posição o emprego de um controlador PD não representa uma diferença muito grande, uma vez que o controlador híbrido tradicional pode empregar também um controlador PD [Yos90]. Na malha de força, porém, usualmente emprega-se controladores com ação integral, de forma a garantir um sinal constante na saída do controlador, permitindo que o manipulador exerça uma determinada força sobre a superfície em contato. Com a substituição do PID pela estrutura PD mais rede neural, a rede, além de compensar a imprecisão no levantamento do modelo e as perturbações externas, também passa a atuar como o termo integral, garantindo erro nulo de força.

Como aconteceu para o caso de controle de posição, provas de estabilidade de redes neurais ainda não se encontram totalmente estabelecidas, prejudicando o estudo da estabilidade do sistema robô + controlador clássico + rede neural.

A figura 4.5 apresenta a estrutura de controle neural de posição proposta.

Ao contrário da estrutura proposta em [FSTM92], optou-se por empregar duas

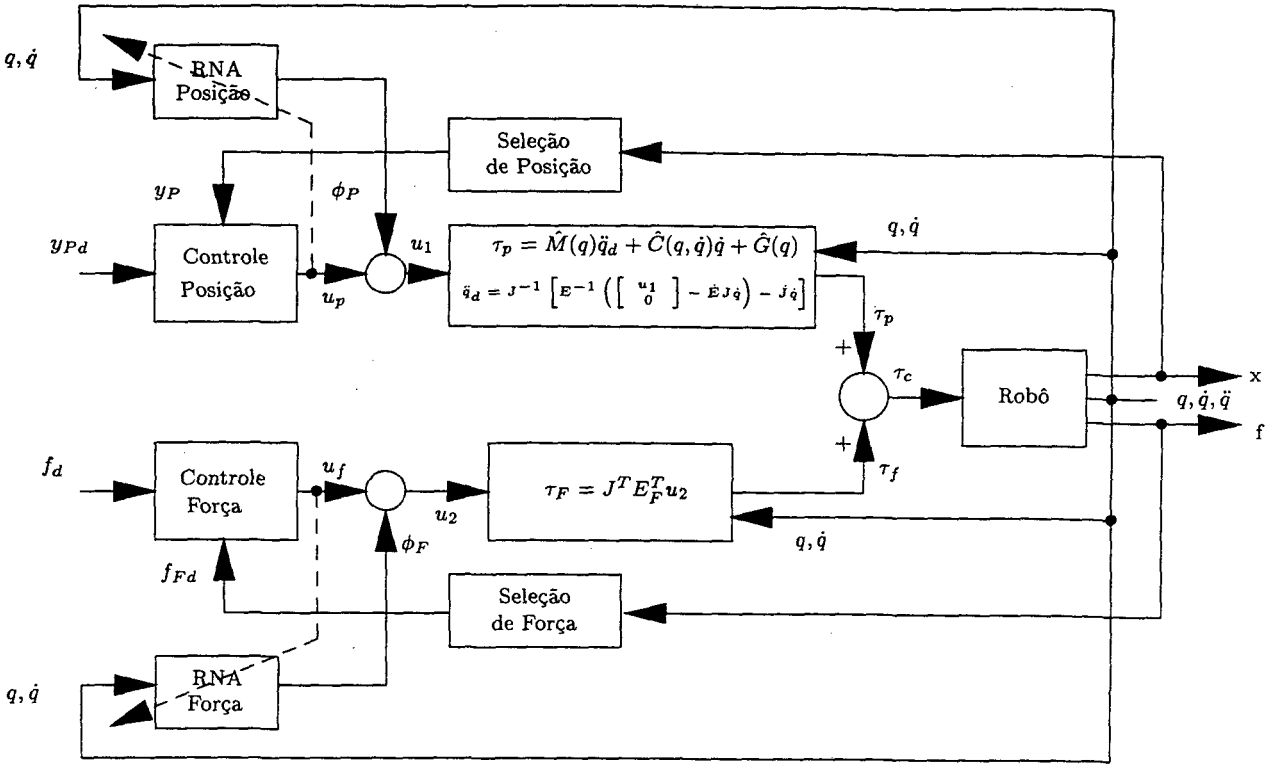

Figura 4.5: Controle Híbrido Neural de Força/Posição

redes. Uma rede para a malha de posição ϕ_P e outra rede para a malha de força ϕ_F. Os controladores neurais assim estabelecidos, permitem um projeto mais simplificado, separando o problema de controle em duas situações distintas e diversas.

Dessa forma, os sinais de controle u_1 e u_2 passam a ser definidos por

$$\begin{aligned} u_1(k) &= K_{Pp}e_y + K_{Dp}\frac{\partial e_y}{\partial t} + \phi_P \\ u_2(k) &= K_{Pf}e_f + K_{Df}\frac{\partial e_f}{\partial t} + \phi_F. \end{aligned} \tag{4.25}$$

onde e_y e e_f são os erros de posição e força, respectivamente.

A vantagem dessa estrutura neural consiste na simplicidade, na não necessidade de treinamento *off-line*, uma vez que a tarefa das redes neurais consiste basicamente em minimizar o esforço de controle, atuando em situações em que os controladores clássicos não conseguem oferecer uma boa performance.

A grande desvantagem desse método é a necessidade do conhecimento da dinâmica do manipulador. A equação 4.19 exige o conhecimento das matrizes de inércia, e dos vetores de forças de coriolís e torques centrífugos, e do vetor gravidade. Entretanto,

com valores nominais conhecidos, mas não exatos, a rede consegue compensar tais imprecisões, corrigindo o modelo dinâmico. Tal fato não acontece com a estrutura de controle híbrida dinâmica convencional.

Controlador Híbrido Neural sem Modelo Dinâmico

Uma desvantagem apresentada nos dois métodos anteriores, consiste na necessidade da dedução analítica do modelo completo do manipulador. As equações 4.18 à 4.21 e 4.25 empregam as matrizes de inércia, Coriolís e torques gravitacionais no modelo, de forma a permitir a implementação da lei de controle híbrido. Ainda que a lei de controle neural necessite somente dos valores nominais dos parâmetros do manipulador, com a rede compensando as não-linearidades e variações paramétricas não representadas pelo modelo paramétrico, o conhecimento desse modelo se faz necessário.

Procurando contornar a necessidade do levantamento das matrizes dinâmicas do manipulador, pode-se empregar uma estrutura de controle mais simples, que não inclua tais matrizes na sua lei de controle. A equação 6.21 propões essa nova lei de controle.

$$\begin{aligned} \tau &= \ddot{q}_d + \tau_F \\ \ddot{q}_d &= J^{-1}\left\{E^{-1}\left(\begin{bmatrix} u_1 + \phi_P \\ 0 \end{bmatrix} - \dot{E}J\dot{q}\right) - \dot{J}\dot{q}\right\} \\ \tau_F &= J^T E_F^T(u_2 + \phi_F) \end{aligned} \tag{4.26}$$

Os sinais u_1 e u_2 sendo definidos por controladores do tipo PD e ϕ_P e ϕ_F são, respectivamente, as redes para a malha de posição e força. Essa lei de controle é derivada do controle de posição, e também está baseada no conceito de que se a rede conseguir minimizar o sinal de saída de um controlador, necessariamente, o erro de posição é nulo.

Entretanto, como acontece com as demais estruturas de controle neural propostas nesse trabalho, seria necessário estabelecer considerações quanto as propriedades de estabilidade do sistema robô + controlador neural.

O mapeamento dos erros no espaço cartesiano para o espaço de juntas é feito pelo sinal $\ddot{q}_d$.

Porém, observou-se durante as simulações que essa estrutura não era capaz de compensar o termo de gravidade, pois a ação de controle da rede neural não era suficientemente rápida para corrigir o efeito gravitacional. Dessa forma, introduziu-se um termo de compensação, dado por $(m_3 + m_4)g$ que representa a ação da gravidade na junta de

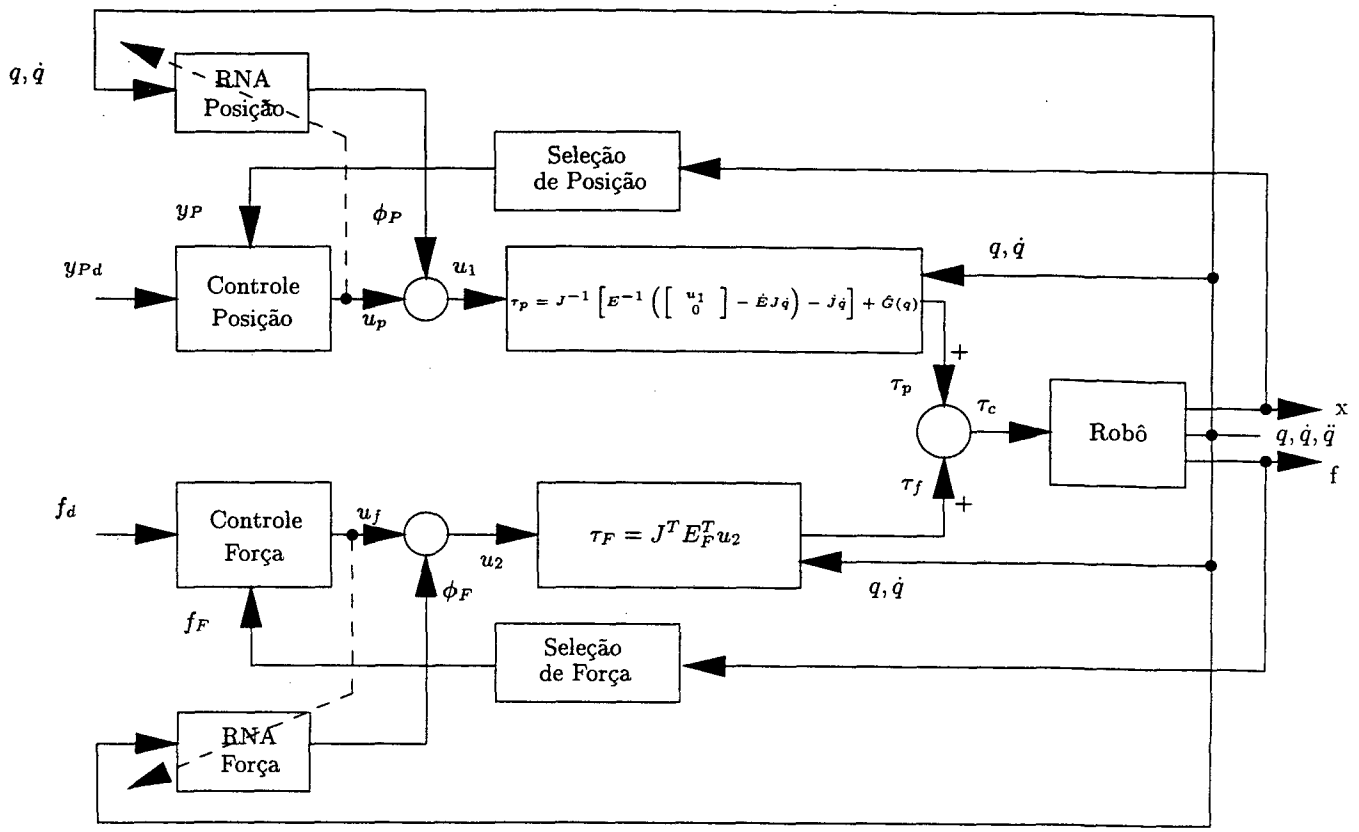

Figura 4.6: Controle Híbrido Neural sem Modelo Dinâmico

translação, conforme será visto no capítulo 6. Os valores de massa correspondem aos valores nominais do manipulador, que não necessariamente correspondem aos valores reais.

A equação de controle passa então a ser dada por:

$$\begin{aligned} \tau &= \ddot{q}_d + \tau_F + \hat{G}(q) \\ \ddot{q}_d &= J^{-1}\left\{E^{-1}\left(\begin{bmatrix} u_1 + \phi_P \\ 0 \end{bmatrix} - \dot{E}J\dot{q}\right) - \dot{J}\dot{q}\right\} \\ \tau_F &= J^T E_F^T (u_2 + \phi_F) \end{aligned} \tag{4.27}$$

O objetivo das redes ϕ_P e ϕ_F passam a ser a compensação de toda a dinâmica do manipulador. A grande vantagem nessa abordagem reside na não necessidade da dedução analítica do modelo do manipulador.

4.5 Conclusões

O emprego de estruturas adaptativas em controle de robôs é uma alternativa às técnicas de controle convencionais, que necessitam do conhecimento dos parâmetros do modelo do manipulador. Controladores adaptativos vêm sendo empregados com sucesso na área de controle de posição, porém apresentam a desvantagem de necessitarem de um modelo paramétrico do robô. Além disso, não se conhece muito bem o desempenho de tais controladores frente as incertezas não-estruturais do sistema. A vantagem do emprego de redes neurais reside justamente na não necessidade do conhecimento da planta, sendo capaz de aprender o modelo dinâmico do manipulador, incluindo as não-linearidades e também as incertezas estruturadas e não-estruturais.

Entretanto, a necessidade de um treinamento prévio *off-line*, com o emprego de um conjunto de exemplos obtidos do sistema real, pode ser um entrave a sua utilização em robótica. Primeiro porque, de acordo com a qualidade do conjunto de exemplos utilizado, a forma de treinamento empregada, que está intimamente associada ao tipo de estrutura de rede (MLP, CMAC, RBF, redes recorrentes), não é possível prever o comportamento da rede para situações em que ela não foi treinada. Isto é, a capacidade de generalização da rede pode não ser suficiente para garantir uma performance satisfatória do sistema de controle. Segundo, o processo de treinamento geralmente é lento e exige um custo computacional elevado. Em manipuladores robóticos de mais de três graus de liberdade, a quantidade de dados é relativamente alta, aumentando o tempo de treinamento.

A forma como uma rede neural é empregada em uma estrutura de controle pode definir a sua eficiência. Estruturas de controle em que a rede assume um papel mais de coadjuvante, não necessitando aprender um conjunto de informações relativamente elevada, como mapear a dinâmica não-linear, multivariável e acoplada de um manipulador de n graus de liberdade, o desempenho apresenta uma performance melhor, e com uma simplicidade maior no projeto do controlador e da rede.

O emprego de redes neurais na área de controle de posição de robôs manipuladores já encontra-se bem estruturado com resultados consistentes. Entretanto, na área de controle de força, onde as dificuldades inerentes às interações entre o manipulador e as superfícies dos objetos em contato com o manipulador ainda não encontram-se definitivamente conhecidas, a utilização de redes neurais ainda não está amplamente difundida, necessitando de muitas pesquisas e trabalhos experimentais.

O controlador neural de posição, apresentado na seção 4.3.2, e os controladores neurais de força, propostos na seção 4.4.2, foram implementados no controle de um manipulador SCARA, e os resultados das simulações numéricas e experimentos práticos

são apresentados nos capítulos que seguem.

Capítulo 5

Resultados: Controle de Posição

5.1 Introdução

Nesse capítulo são apresentados os resultados obtidos através de simulações e de uma implementação prática do controle de posição do robô manipulador do tipo SCARA, existente no Laboratório de Robótica da Universidade Federal de Santa Catarina.

As equações que descrevem o comportamento cinemático e dinâmico do manipulador correspondem ao apresentado na seção 2.3.3. Os dados do manipulador encontram-se no apêndice A e mais detalhes técnicos podem ser encontrados em [GWG98]. Foi considerada a estrutura completa do manipulador, compreendendo os seus quatro graus de liberdade, em lugar do modelo simplificado de dois graus de liberdade utilizado na maioria dos trabalhos disponíveis na literatura.

Para os exemplos de simulação são analisados o desempenho de três estruturas de controle: o controle PD clássico sem compensação de gravidade, o controlador dinâmico inverso e o controlador neural proposto nesse trabalho. Na implementação prática do controle de posição analisa-se o desempenho de duas estruturas de controle: controle PD e a proposta de controle neural de posição.

Os controladores PD e PID utilizados nas estratégias de controle apresentadas no capítulo 4 foram discretizados.

Para realização das simulações foram empregados os softwares MATLAB 4.0 e SIMULINK 1.3c.

5.2 Simulações

5.2.1 Trajetórias de Referência e Situações Consideradas

Nesse trabalho, buscou-se especificar as trajetórias de referência diretamente no espaço de juntas. Dessa forma, não foi necessário resolver o problema da cinemática inversa - seção 2.3.1.

As funções que definem os valores desejados de deslocamento nas juntas do manipulador, devem ser funções que descrevem trajetórias suaves para posição, velocidade e aceleração. Ou seja, a função que especifica a referência deve ser contínua e duplamente diferenciável, correspondendo a funções uniformemente limitadas. Funções polinomiais, ou compostas por senos e cosenos apresentam essas propriedades, e podem ser empregadas para gerar os sinais de referência. Outro tipo de curva que pode ser empregada para gerar a trajetória de referência, constituem as curvas do tipo *spline*.

Foram verificados os desempenhos dos três controladores de posição apresentados no capítulo anterior: o controlador PD sem compensação de gravidade, a estrutura de controle pela dinâmica inversa e a proposta de controle neural.

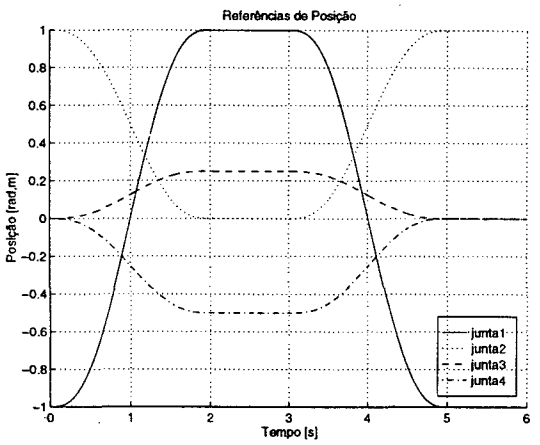

Figura 5.1: Referência de Posição nas Juntas

As trajetórias de referência empregadas constituem-se de curvas do tipo *spline* e são apresentadas na figura 5.1. O objetivo do sistema de controle consiste em garantir o seguimento da trajetória, durante toda a sua execução.

A trajetória consiste de um movimento suave, de forma a não excitar os modos de alta freqüência do manipulador. Nas simulações realizadas, o manipulador é movimentado a partir da sua posição inicial até um ponto intermediário, dentro de um intervalo de tempo de dois segundos. Tendo o manipulador atingido essa posição, o controlador deve ser capaz de manter o manipulador nessa posição intermediária durante um se-

gundo. Em seguida, o manipulador deve voltar a posição original, novamente dentro de um intervalo de tempo de dois segundos, e assim manter-se nessa posição por mais um segundo.

Procurando verificar a robustez dos controladores, verificou-se o desempenho de cada uma das estruturas de controle em três situações distintas:

Condições Nominais: Supõe-se que o modelo seja exato, sem existência de imprecisões e perturbações atuando sobre o manipulador.

Variação Paramétrica e Perturbações: Assume-se que somente os valores nominais dos parâmetros da equação dinâmica do manipulador são conhecidos, existindo imprecisões nos parâmetros do manipulador. Considera-se também a existência de perturbações.

Variação de Carga: Uma carga é acoplada ao efetuador-final do manipulador, com valores de massa e inércia desconhecidos.

O período de amostragem empregado foi de 2.5 ms.

5.2.2 Projeto dos Controladores

Para o ajuste dos ganhos dos controladores duas foram as possibilidades levantadas. A primeira consiste em obter o melhor ajuste de ganhos para cada estrutura de controle tomadas individualmente, levando em consideração somente as condições nominais de operação. Dessa forma, a análise das estratégias de controle enfoca, principalmente, o desempenho individual dos controladores.

Uma segunda possibilidade, consiste em empregar as mesmas matrizes de ganho dos controladores PD para as três estratégias de controle. Dessa forma, a análise do desempenho dos controladores pode ser feito diretamente, comparando-se os valores das ações de controle, erros de posição e velocidade. Entretanto, a análise assim realizada, enfoca o desempenho de uma estrutura de controle do tipo PD em três situações distintas: controle PD puro, controle PD com o modelo nominal da planta (controle inverso) e controlador PD com um sistema de compensação "adaptativa" (rede neural).

Durante a realização dos experimentos, foram simuladas as duas propostas acima. No caso da segunda proposta, verificou-se o comportamento das três estratégias de controle empregando-se os valores de ganho reais utilizado para o controle do manipulador SCARA Inter do Laboratória de Robótica, e também com os valores de ganhos do melhor controlador PD puro atuando em condições nominais. Entretanto, observou-se

que com tais valores de ganhos, as estratégias de controle inversa e neural apresentaram saturações na ação de controle, pelo fato do ajuste dos ganhos dos controladores PD empregado nessas estratégias, não estarem baseados em um critério que leve em consideração a existência de uma estrutura auxiliar de controle, atuando em paralelo com o controlador PD. Sendo assim, optou-se por escolher, para as três estratégias de controle, o conjunto de parâmetros que apresentasse o melhor desempenho.

Todos os controladores foram projetados levando em consideração somente as condições nominais de operação. Nas situações em que são consideradas as incertezas e variações paramétricas sobre o modelo, bem como a presença de perturbações e cargas desconhecidas acopladas ao efetuador final, utiliza-se o controlador previamente sintonizado, não sendo realizados novos ajustes.

Controlador PD

O controlador PD digital foi ajustado de forma a apresentar um bom desempenho para o caso nominal. Não foram levadas em consideração, durante a fase de ajuste dos ganhos do controlador, a existência de perturbação e assumiu-se desconhecidas as variações paramétricas.

O controlador digital empregado foi:

$$u_0(k) = \ddot{q}_d(k) + K_P e(k) + \frac{K_D}{T}\left[e(k) - e(k-1)\right] \tag{5.1}$$

onde q_d corresponde aos valores desejados para as juntas, $e(k)$ corresponde ao erro de posição nas juntas. As matrizes de ganhos obtidas correspondem à

$$K_P = \begin{bmatrix} 175 & 0 & 0 & 0 \\ 0 & 126 & 0 & 0 \\ 0 & 0 & 350 & 0 \\ 0 & 0 & 0 & 21 \end{bmatrix}$$

$$K_D = \begin{bmatrix} 66 & 0 & 0 & 0 \\ 0 & 24 & 0 & 0 \\ 0 & 0 & 150 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}. \tag{5.2}$$

Controle Dinâmico Inverso

Para gerar o sinal u_0 na equação 4.19, foi empregado um controlador do tipo PID. Conforme mencionado anteriormente, o projeto do controlador dinâmico inverso foi

feito somente levando em consideração as condições nominais do manipulador, sendo negligenciada as perturbações e variações paramétricas, durante a fase de projeto[1].

A equação do controlador PID pode ser obtido através do emprego da aproximação retangular para o termo integral. Dessa forma, obtém-se

$$u_0(k) = \ddot{q}_d(k) + u_0(k-1) + p_0 e(k) + p_1 e(k-1) + p_2 e(k-2) \tag{5.3}$$

com

$$\begin{aligned} p_0 &= K_P\left(1 + \frac{K_D}{T}\right) \\ p_1 &= -K_P\left(1 + 2\frac{K_D}{T} - K_I T\right) \\ p_2 &= \frac{K_P K_D}{T} \end{aligned} \tag{5.4}$$

onde T é o valor do período de amostragem.

Os ganhos do PID utilizado correspondem a

$$\begin{aligned} K_P &= \begin{bmatrix} 8 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \\ K_D &= \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \\ K_I &= \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}. \end{aligned} \tag{5.5}$$

Controle Neural

O projeto do controlador digital neural consiste de duas fase distintas: o ajuste do controlador PD e a definição da topologia da rede neural.

[1] Observa-se que, para o caso onde o modelo empregado pela lei de controle corresponde exatamente ao modelo real, o erro de posição e velocidade é nulo. Entretanto, devido aos efeitos das imprecisões e limitações existentes nas simulações numéricas, os resultados obtidos para o caso nominal, apresentam erros não "previstos" pela teoria

A utilização da rede neural na estrutura de controle envolve a definição da quantidade de camadas de neurônios, o número de neurônios em cada camada, o algoritmo de aprendizagem e o ajuste de seus parâmetros.

Muitas vezes uma rede com um número elevado de camadas ou neurônios pode apresentar o problema de sobretreinamento. Nesses casos, a rede apresenta um erro de treinamento relativamente baixo, se considerado os exemplos contidos no conjunto de treinamento, porém o mesmo não acontece com um conjunto de teste escolhido para verificar a capacidade de generalização da rede [Kli96]. Assim, existe um compromisso entre o número de camadas e neurônios e as capacidades de aprendizagem e generalização da rede.

A escolha do número de camadas, algoritmo de treinamento, número de neurônios, valores iniciais das conexões foram feitas heuristicamente, buscando manter um compromisso entre desempenho durante regimes transitórios, estabilidade do sistema, tempo de resposta mais rápido e estruturas de rede mais simples e compactas.

Na estrutura de controle neural, foi empregada uma rede MLP, com quatro camadas de neurônios. A camada de entrada contém 12 neurônios - quatro para aos valores de posição q, quatro para os valores de velocidade nas juntas $\dot{q}$, e mais quatro neurônios para os valores de aceleração desejadas $\ddot{q}_d$.

O motivo de se empregar a aceleração da referência na rede, reside no fato da dificuldade de medir ou estimar os valores de aceleração nas juntas.

Foi escolhida como função de ativação dos neurônios a função tangente hiperbólica, com inclinação igual a um. Optou-se escolher a função tangente hiperbólica, ao invés da função logística, pelo fato da sua saída variar entre -1 e 1. Dessa forma, basta multiplicar a saída da rede pelos torques máximos na junta, para obter um espectro de variação da saída da rede compreendendo todo o conjunto possíveis de valores de torque de controle.

A topologia obtida é apresentada na tabela 5.1.

Camada	*Número de Neurônios*
Camada de Entrada	12
Camada Intermediária 1	8
Camada Intermediária 2	5
Camada de Saída	4

Tabela 5.1: Estrutura de Rede

As matrizes de pesos foram inicializadas como valores randômicos compreendidos entre $\pm$ 0.1. Os *bias* dos neurônios foram inicializadas também com valores randômicos,

mas situados na faixa de $\pm$ 0.05. Para valores iniciais de pesos da acima de 0.1, o sistema instabiliza, pois a saída da rede satura muito rapidamente, saturando também a ação de controle. Um valor baixo de inicialização para as conexões se fez necessário, pois inicialmente deseja-se que a rede tenha pouca influência sobre o manipulador. Assim, nos instantes iniciais da trajetória o controlador atua sobre o processo, enquanto a rede passa a gradualmente compensar os erros existentes.

Foram estudados o desempenho de quatro algoritmos diferentes: *backpropagation*, *ABPropagation*, *RPROP* e *quickpropagation*. Detalhes dos algoritmos testados podem ser visto no apêndice *B*. Com exceção do algoritmo *backpropagation*, todos os demais algoritmos conferem uma velocidade mais rápida de treinamento à redes do tipo MLP. Entretanto, considerar que um algoritmo é mais eficiente, ou mais rápido, depende do problema que se está sendo analisado, e também dos quesitos utilizados para avaliar o desempenho dos algoritmos, como velocidade de treinamento, número de vezes em que o algoritmo converge com sucesso, capacidade de generalização apresentada pelo algoritmo, etc.

Durante o desenvolvimento do trabalho, buscou-se utilizar os algoritmos de treinamento anteriormente citados em uma estrutura de controle do tipo *off-line*. Entretanto, observou-se que o desempenho do controlador obtido dependia muito da escolha do conjunto de exemplos, e o tempo de treinamento, para um conjunto relativamente grande de dados era lento, mesmo com o uso de algoritmos acelerados de treinamento. Além disso, em algumas situações, os algoritmos *quickpropagation* e *RPROP* apresentaram uma capacidade de generalização não muito eficiente. Por essas razões, foram abandonadas as estruturas de controle neural com treinamento do tipo *off-line*.

Os algoritmos *quickpropagation* e *RPROP* são algoritmos que apresentam uma velocidade de treinamento superior à do *backpropagation*. O algoritmo *ABPropagation*, por sua vez, constitui-se de um intermediário entre os dois primeiros algoritmos e o *backpropagation*. Depois de inúmeras simulações, estes resultados foram evidenciados para o caso do manipulador SCARA utilizado, sendo empregadas topologias de redes similares para os quatro algoritmos analisados.

Foram realizados testes empregando os algoritmos acelerados de treinamento (*RPROP* e *quickpropagation*) na estratégia de controle *on-line*. Entretanto, devido à rápida convergência que esses algoritmos apresentam, a saída da rede saturava facilmente, levando o sistema controlador neural + robô à instabilidade, tendo sido feitos testes com um conjunto relativamente grande de parâmetros. Isso se deve ao fato de que nos instantes iniciais da atuação do controlador, mesmo para pequenos valores iniciais de pesos e parâmetros que conferiam um desempenho mais lento aos algoritmos, a saída do controlador PD apresentava um valor elevado. Dessa forma, a rede inter-

preta esse sinal de controle como sendo um erro de treinamento extremamente elevado, conferindo um valor de ajuste dos pesos relativamente grande.

As figuras 5.2 e 5.3 apresentam a saída da rede, já convertida para os valores de torque nas juntas, em duas situações distintas. A primeira, para o caso da utilização do algoritmo *backpropagation*, com uma escolha adequada de taxa da aprendizagem. Observa-se que a saída da rede apresenta um comportamento mais suave.

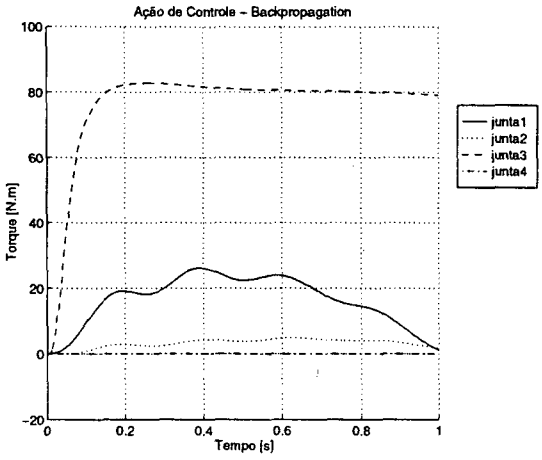

Figura 5.2: Saída da RNA com Algoritmo *Backpropagation*

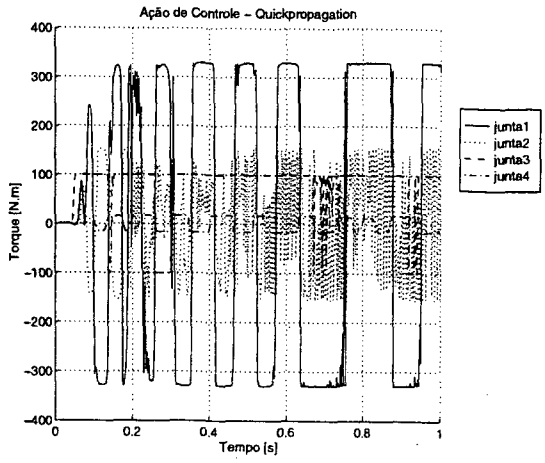

Figura 5.3: Saída da RNA com Algoritmo *Quickpropagation*

No caso da utilização do algoritmo *quickpropagation*, para todos os conjuntos de parâmetros utilizados para sintonia do algoritmo, a saída da rede apresenta um sinal que busca rapidamente compensar os erros, ocasionando um sinal de saída que satura facilmente. Dessa forma a saída da rede fica oscilando entre os extremos, prejudicando a ação de controle.

Os algoritmos *ABPropagation* e *backpropagation* apresentam uma velocidade de treinamento mais lenta. Dessa forma, o valor do passo de atualização dos pesos da rede não é muito elevado, mesmo para os instantes iniciais de atuação do sistema de controle. A rede, nesse caso, passa a atuar lentamente, não sendo vista, portanto, como uma perturbação pelo controlador digital, que passa gradualmente a ser compensado pela estrutura neural.

Optou-se por empregar o algoritmo *backpropagation*, pela sua simplicidade no ajuste de parâmetros, no caso a taxa de aprendizagem η em relação ao algoritmo *ABPropagation*, que necessita da especificação da função Γ, que é a função do erro empregada na atualização dos pesos da rede (apêndice B).

O emprego de algoritmos acelerados de treinamento é indicado para os casos de treinamento *off-line*, em que um conjunto de exemplos contendo uma quantidade relativa-

mente grande de dados a respeito da dinâmica do manipulador tem que ser aprendida por uma RNA. Como a estrutura de controle proposta está baseada no aprendizado *on-line*, as fases de aprendizagem e atuação sobre o processo ocorrem simultaneamente.

A taxa de aprendizado empregada é da ordem de 0.001. Para valores acima de 0.05 o sistema também se torna oscilatório, levando o robô a instabilidade. Isso é decorrente do mesmo problema apresentado pelos algoritmos de treinamento acelerados: a alta velocidade de reação do algoritmo de treinamento faz com que a rede busque compensar de maneira muito brusca e rápida o erro de treinamento, que nos instantes iniciais apresenta um valor elevado.

A sintonia do controlador PD digital é feita em conjunto com a rede. Através de uma boa escolha de ganhos, é possível obter um desempenho satisfatório da estrutura de controle. Como nos outros casos de controle de posição, essa sintonia foi realizada somente para o caso nominal. As situações em que são consideradas variações paramétricas, existência de perturbações e ruídos, empregam-se as estruturas de controle obtidas para o caso nominal, sem a realização de ajustes ou uma nova sintonia dos controladores. Dessa forma, é possível analisar o desempenho dos controladores para situações não previstas ou não modeladas.

O controlador PD digital utilizado em conjunto com a rede é constituído pela seguinte equação:

$$u_0(k) = \ddot{q}_d(k) + K_P e(k) + \frac{K_D}{T}\left[e(k) - e(k-1)\right] \tag{5.6}$$

com os seguintes valores de ganho:

$$K_P = \begin{bmatrix} 480 & 0 & 0 & 0 \\ 0 & 108 & 0 & 0 \\ 0 & 0 & 300 & 0 \\ 0 & 0 & 0 & 18 \end{bmatrix}$$

$$K_D = \begin{bmatrix} 210 & 0 & 0 & 0 \\ 0 & 40 & 0 & 0 \\ 0 & 0 & 250 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}. \tag{5.7}$$

É importante ressaltar que vários teste foram realizados para chegar aos valores dos ganhos dos controladores PD e PID, nas três estratégias de controle de posição. Optou-se por empregar, sempre, a sintonia que apresentasse o melhor desempenho para o caso nominal. Assim, como pode se observar nas matrizes de ganhos, as três

estratégias apresentam valores diferentes. Entretanto, como será visto nos resultados experimentais da seção 5.3, a rede neural também pode ser acoplada a um controlador previamente sintonizado atuando sobre o processo.

5.2.3 Resultados das Simulações

Condições Nominais

Considerando o manipulador livre de perturbações externas e sem imprecisões na sua modelagem, observou-se o desempenho dos controladores para o seguimento da trajetória da figura 5.1.

As figuras 5.4, 5.5 e 5.6 apresentam o desempenho dos controladores de posição.

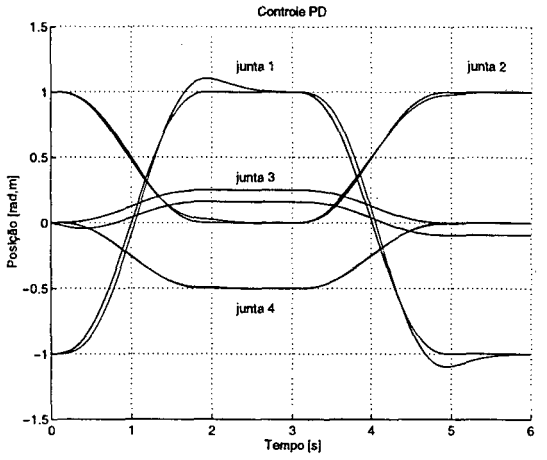

Figura 5.4: Controle PD: Caso Nominal

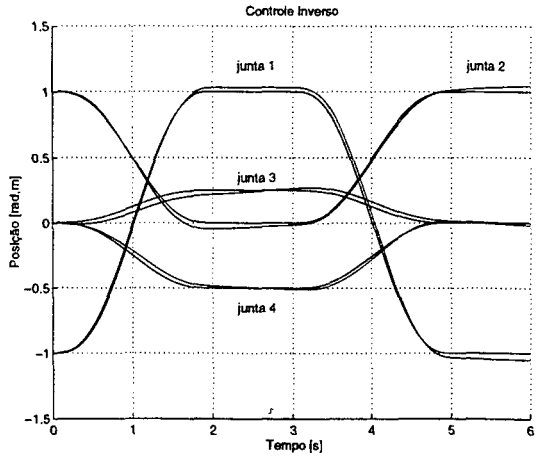

Figura 5.5: Controle Inverso: Caso Nominal

A existência de um erro em regime permanente do posicionamento da terceira junta, ocorre em função da não compensação do termo de gravidade pelo controlador PD -

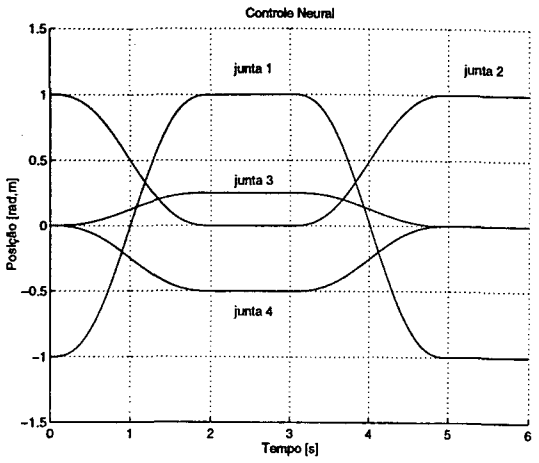

Figura 5.6: Controle Neural: Caso Nominal

figura 5.4. Para as demais juntas, o controlador PD apresenta um bom desempenho, não apresentando picos ou tempo de respostas muito longos.

A estrutura de controle inverso consegue corrigir a deficiência apresentada pelo controlador PD, no que tange à compensação da gravidade na terceira junta - figura 5.5. Isso se deve ao fato dessa estratégia empregar o vetor de torques gravitacionais para compensar o efeito da gravidade. Além disso, o desacoplamento e a linearização são garantidos, caso os valores estimados para os coeficientes do manipulador correspondam aos valores reais[2].

Por sua vez, o controlador neural apresenta um seguimento com um erro muito pequeno, em comparação as outras duas estruturas de controle - figura 5.6. Observa-se que a capacidade de compensação que a rede oferece ao sistema de controle, no caso, contrabalança o efeito gravitacional existente na terceira junta do manipulador.

Variação Paramétrica e Perturbações

Considera-se a existência de imprecisões no conhecimento dos parâmetros, decorrente de possíveis erros de estimação ou variações paramétricas devido a influências do meio. Os valores de variação considerados, podem ser vistos na tabela 5.2. Foram consideradas variações significativas, de forma a melhor evidenciar o desempenho dos controladores.

De forma a verificar o comportamento dos controladores frente a existência de perturbações, foram aplicados torques contrários ao movimento do manipulador. A

[2]Obviamente, a existência de erros devido a precisão numérica utilizada para realizar as simulações, não corresponde ao previsto pela teoria, uma vez que os valores dos parâmetros do modelo, utilizado na lei de controle, corresponde aos valores reais do manipulador

Parâmetro	Variação
m_1	+ 20%
m_3	- 20%
m_4	+ 20%
I_1	+ 10%
I_3	- 10%
I_4	+ 10%
l_1	+ 15%
l_2	- 15%

Tabela 5.2: Variação Paramétrica

tabela 5.3 apresenta as perturbações aplicadas sobre o robô.

Junta	Valor	Instante
Junta1	20% do torque máximo da junta 1	3 segundos
Junta3	20% do torque máximo da junta 4	4 segundos

Tabela 5.3: Perturbações

As figuras 5.7, 5.8 e 5.9 apresentam o desempenho dos controladores para os casos de perturbação e variação de parâmetros.

O controlador PD não sofre muita influência das variações paramétricas. Essa característica pode ser vista nos primeiros 3 segundos do seguimento de trajetória - figura 5.7. Nesse caso, os ganhos do controlador ainda são suficientes para manter um bom desempenho de controle frente as variações paramétricas, ou seja, as variações nos parâmetros ainda encontram-se dentro da faixa de estabilidade do controlador. Observa-se que a resposta, até o instante 3 segundos, é similar a resposta da figura 5.4, onde se considerava somente o caso nominal.

Entretanto, observa-se que com a aplicação de uma perturbação, o controlador PD não consegue responder dentro do tempo de resposta especificado pela referência. Os erros devido ao efeito de gravidade também é agravado pela perturbação existente na terceira junta.

Por estar baseado no conhecimento exato das equações dinâmicas do manipulador, a linearização e o desacoplamento feitos pela estrutura de controle inverso não ocorrem, uma vez que os parâmetros do manipulador utilizados na lei de controle são diferentes dos valores reais. Além disso, o modelo não prevê a existência de perturbações. Assim, como pode-se observar na figura 5.8, quando a perturbação é aplicada nas juntas, o erro

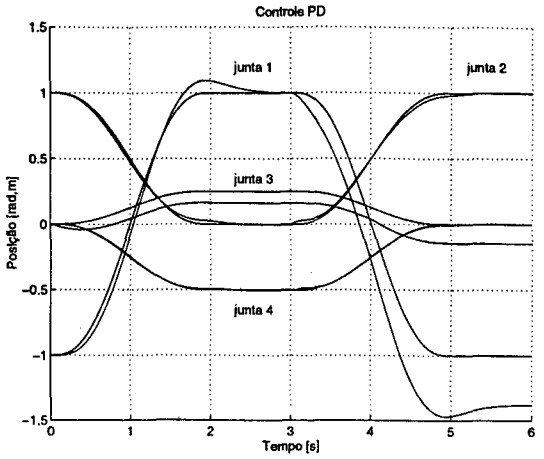

Figura 5.7: Controle PD: Variação Paramétrica e Perturbação

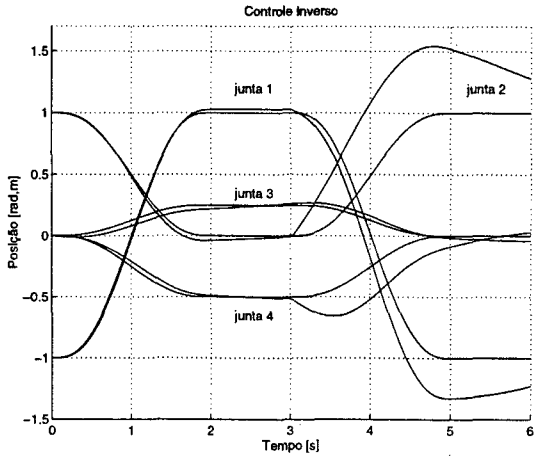

Figura 5.8: Controle Inverso: Variação Paramétrica e Perturbação

de posicionamento aumenta, gerando um *overshoot* muito elevado, e não conseguindo cumprir as especificações de controle em malha-fechada, apesar de ainda garantir a estabilidade do sistema.

Na estratégia de controle neural, a rede é capaz de compensar as variações paramétricas e as perturbações atuando sobre o manipulador. No caso simulado - figura 5.9 - a influência da perturbação não chega a ser visível, garantindo um seguimento de trajetória similar ao do caso nominal.

Variação de Carga

Procurando executar a trajetória da figura 5.1, observa-se o desempenho dos controladores para o caso em que é considerado a manipulação de uma carga desconhecida. Dessa forma, o valor desta carga não se encontra incluído no modelo dinâmico do manipulador, não sendo possível prever a variação nominal da carga acoplada ao efetuador

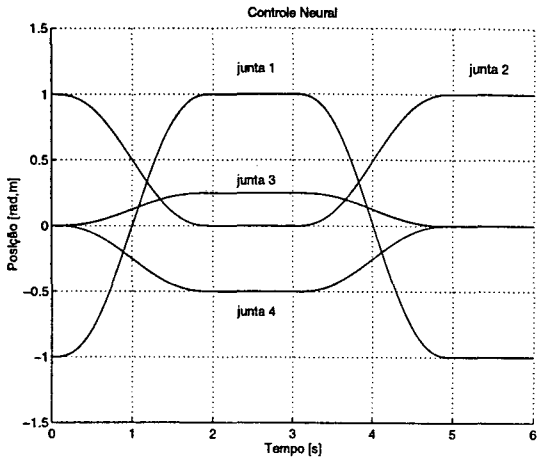

Figura 5.9: Controle Neural: Variação Paramétrica e Perturbação

final. A carga possui uma massa de 5 Kg, e encontra-se acoplada ao manipulador nos primeiros 4 segundos da trajetória.

As figuras 5.10, 5.11 e 5.12 apresentam o desempenho dos controladores para o caso de variação de carga.

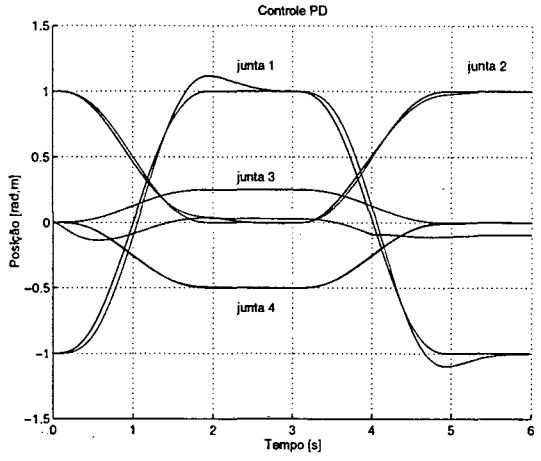

Figura 5.10: Controle PD: Variação de Carga

O acoplamento de uma carga no efetuador final, corresponde em um acréscimo de massa na terceira junta. Dessa forma, os efeitos dos torques gravitacionais são mais relevantes, como pode ser visto na figura 5.10. Conforme visto no caso nominal, o controlador PD não é capaz de compensar os efeitos dos torques gravitacionais, e um aumento de massa e inércia na terceira junta, correspondendo a uma influência maior do termo de gravidade, contribuindo para a ocorrência de um erro ainda maior no posicionamento da junta 3.

O aumento de massa na junta 3, corresponde a uma alteração nos valores de massa e do termo de gravidade da matriz de torques gravitacionais, empregadas na lei de con-

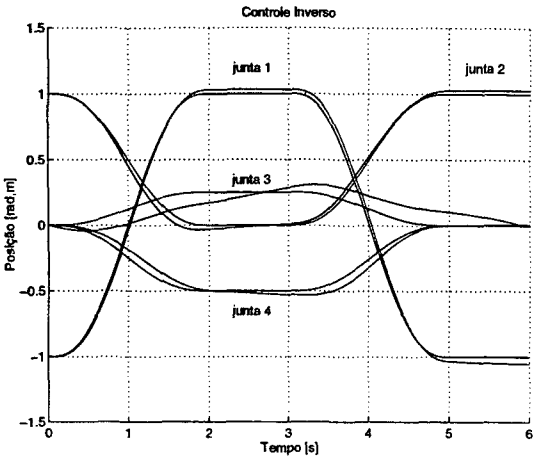

Figura 5.11: Controle Inverso: Variação de Carga

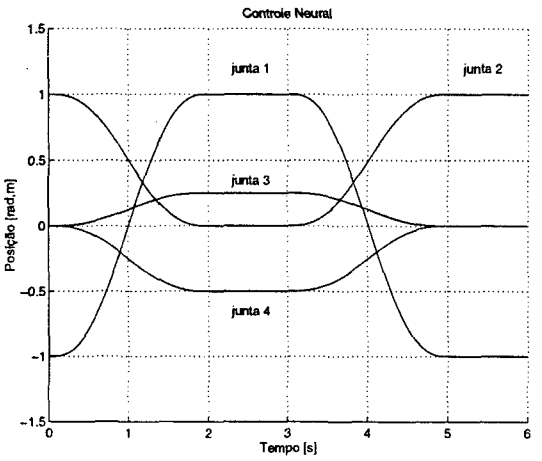

Figura 5.12: Controle Neural: Variação de Carga

trole do controlador dinâmico inverso - figura 5.11. A variação global nos parâmetros é muito menos significativa se comparado com o caso de variação paramétrica e perturbação, analisados anteriormente, uma vez que nesse caso somente um parâmetro teve seu valor alterado. Essa variação de carga pode ser vista como uma variação nas massas e inércias do terceiro ou quarto elo do manipulador. O controlador inverso, nesse caso, também apresenta seu desempenho de controle prejudicado, principalmente no controle de posição da terceira junta.

Novamente a estrutura de controle neural foi capaz de satisfazer os requisitos de controle. A rede, nesse caso, funciona como um compensador de gravidade. O torque provido pela rede é capaz de suprir a deficiência da estrutura de controle do tipo PD, que não apresenta a capacidade de compensação dos termos de gravidade - figura 5.12.

Análise do Esforço de Controle

As figuras 5.13, 5.14 e 5.15 apresentam as ações de controle para o caso onde existem variações paramétricas e perturbações atuando sobre o manipulador (segundo caso).

Observa-se que o controlador neural apresenta uma ação de controle mais oscilatória. Esse inconveniente acontece devido à dois fatores. A primeira influência é decorrente da necessidade de ajuste dos pesos da rede frente às mudanças de referência ou à existência de uma perturbação. O segundo fator provém da falta de conhecimento sobre a interação entre o controlador clássico e a rede neural, não sendo conhecidos critérios de projeto e análise de estabilidade do sistema manipulador/controlador + rede neural, de forma a incluir a otimização da lei de controle durante a fase de projeto do controlador. Além disso, o treinamento da rede neural encontra-se baseado somente no erro de posicionamento, e não leva em conta o esforço da ação de controle.

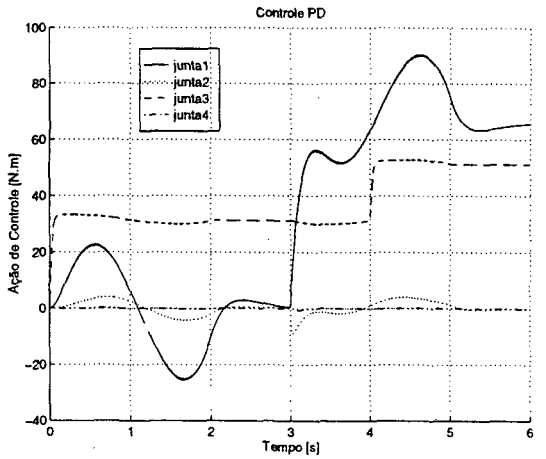

Figura 5.13: Controle PD: Ação de Controle

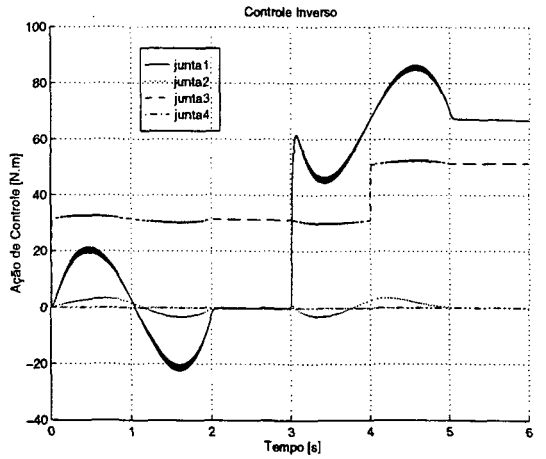

Figura 5.14: Controle Inverso: Ação de Controle

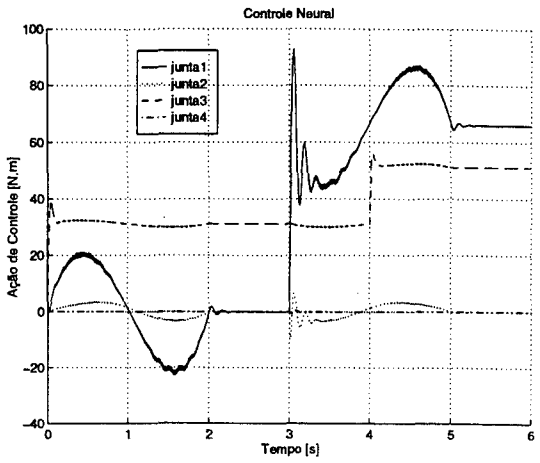

Figura 5.15: Controle Neural: Ação de Controle

Análise de Erro

Uma possibilidade de analisar o desempenho global das estruturas de controle com o objetivo de estabelecer comparações de ordem qualitativa, é estabelecer um critério de desempenho ou um índice de medida de desempenho. Buscou-se assim, verificar o erro de posicionamento nas juntas durante todo o seguimento da trajetória da figura 5.1. Dessa forma, utilizou-se um erro médio para cada junta do manipulador[3]. Esse valor é obtido através da seguinte expressão:

$$erro_i = \frac{1}{p}\sum_{k}^{p} \|(q_{id} - q_d)\| \tag{5.8}$$

onde i é a junta em questão, p o conjunto de pontos na trajetória e $q_{id} - q_d$ é o erro de posição.

Esse critério não é confiável para estabelecer uma análise precisa sobre o desempenho individual de cada estratégia, uma vez em que o cálculo do erro médio pode mascarar os resultados obtidos. Entretanto, a título de comparação entre várias estratégias de controle, atuando nas mesmas condições, tal medida de erro pode ser útil na análise qualitativa do desempenho das estruturas de controle.

As tabelas 5.4, 5.5 e 5.6 apresentam uma comparação entre os erros de posição para cada uma das estruturas de controle empregadas.

Observa-se que o desempenho do controlador inverso[4] encontra-se diretamente relacionado com a precisão do modelo do manipulador. Para os casos analisados, essa

[3] A escolha desse critério está baseada na necessidade de se utilizar um índice ou fator comum, para a análise qualitativa do erro de posição.

[4] O resultado da tabela 5.4 não corresponde, para o caso do controlador dinâmico inverso, ao previsto pela teoria, pois a parcela de erro existente é referente a falta de precisão nas simulações dos experimentos.

Nominal	PD	Inverso	Neural
Junta1 (rad)	0.0512	0.0318	0.0003
Junta2 (rad)	0.0185	0.0220	0.0005
Junta3 (m)	0.0852	0.0250	0.0005
Junta4 (rad)	0.0036	0.0144	0.0005

Tabela 5.4: Erro: Caso Nominal

Perturbação e Variação Paramétrica	PD	Inverso	Neural
Junta1 (rad)	0.1896	0.1217	0.0004
Junta2 (rad)	0.0196	0.2327	0.0005
Junta3 (m)	0.0978	0.0260	0.0009
Junta4 (rad)	0.0035	0.0743	0.0004

Tabela 5.5: Erro: Caso Variação Paramétrica e Perturbação

estrutura de controle apresentou um desempenho inferior ao do controlador PD, quando as incertezas e perturbações se faziam presentes - tabela 5.5.

Os valores de erro obtidos para o caso do controlador neural são extremamente baixos, se comparados com os outros dois métodos, indicando um bom desempenho durante a execução da trajetória. Dessa forma, pode-se concluir que o controlador neural apresenta o melhor desempenho, dentre os três controladores analisados, no que diz respeito à precisão na realização da trajetória.

Variação de Carga	PD	Inverso	Neural
Junta1 (rad)	0.0552	0.0325	0.0005
Junta2 (rad)	0.0166	0.0241	0.0007
Junta3 (m)	0.1719	0.0741	0.0032
Junta4 (rad)	0.0037	0.0280	0.0007

Tabela 5.6: Erro: Caso Variação de Carga

5.3 Resultados Experimentais

O objetivo dessa seção consiste em comprovar em uma aplicação prática os resultados obtidos com a estrutura de controle neural proposta. Dessa forma, implementou-se controlador neural de posição no manipulador robótico SCARA Inter existente no Laboratório de Robótica. O desempenho do controlador neural é comparado com o controlador PD atualmente implementado para o controle de posição e velocidade do manipulador. Os dados quanto aos valores de massa, centro de massa, inércia e comprimento de cada elo, bem como os valores máximos de posição, velocidade e aceleração encontram-se listados no apêndice A.

O sistema responsável pelo controle, intertravamento, acionamento dos motores e comunicação do robô SCARA Inter está implementado na plataforma *XOberon*. O *XOberon* consiste de um ambiente de programação orientado à objetos e concorrente, ideal para aplicações em tempo-real [Mos93] e [Rei91]. A implementação do sistema de controle neural também foi realizada nesse ambiente, e os procedimentos mais importantes encontram-se listados no apêndice C.

5.3.1 Referências Consideradas

O desempenho dos controladores foram verificados em duas situações distintas. O primeiro caso consiste do seguimento de uma trajetória em condições nominais de funcionamento do manipulador. Na segunda situação, é considerada a existência de perturbações atuando sobre a dinâmica do manipulador.

A posição inicial das juntas do manipulador é $x_0 = [0.0\ 0.0\ 0.2\ 0.0]^T$. O manipulador deve realizar um movimento empregando curvas do tipo *spline* como referência [5]. A posição final desejada para o manipulador corresponde à $x_f = [0.5\ -0.5\ 0.3\ 0.8]^T$.

A velocidade máxima permitida para as juntas corresponde à 2.0 rad/s, e a aceleração máxima à 1.5 rad/s^2.

Na situação em que se consideram as perturbações, é aplicada uma perturbação de 10% do máximo torque permitido para a junta 1.

A seguir são apresentados os projetos dos controladores e os resultados obtidos.

5.3.2 Projeto dos Controladores

Uma das vantagens do emprego da estrutura de controle neural proposta nesse trabalho, consiste do fato de poder ser utilizada com uma estrutura de controle já

[5] Atualmente, o sistema de controle e geração de trajetórias do manipulador SCARA Inter utiliza como referência funções do tipo *spline*.

implementada e atuando no sistema. Dessa forma, o projeto do controlador neural consistiu em empregar a estrutura de controle existente no manipulador, no caso o controle clássico PD, e acrescentar a malha com a rede neural artificial[6].

O controlador PD digital atualmente implementado no manipulador SCARA correspondem à:

$$u_0(k) = \ddot{q}_d(k) + K_P e(k) + \frac{K_D}{T}\left[e(k) - e(k-1)\right] \tag{5.9}$$

onde q_d corresponde aos valores desejados para as juntas, $e(k)$ corresponde ao erro de posição nas juntas. As matrizes de ganhos dos controladores correspondem à

$$K_P = \begin{bmatrix} 4.0\,10^4 & 0 & 0 & 0 \\ 0 & 2.4\,10^4 & 0 & 0 \\ 0 & 0 & 40 & 0 \\ 0 & 0 & 0 & 400 \end{bmatrix}$$

$$K_D = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 70 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.10}$$

O período de amostragem utilizado foi de 1 ms.

A rede neural empregada apresenta a mesma estrutura da rede utilizada nas simulações, ou seja, 12 neurônios na camada de entrada - um para cada valor de posição, velocidade e aceleração das juntas, 8 e 5 neurônios nas camadas intermediárias e 4 na camada de saída. A função dos neurônios são do tipo tangente hiperbólica, com inclinação igual a um.

As matrizes de pesos e *bias* dos neurônios foram inicializadas com valores randômicos da ordem de $\pm$ 0.1.

Foram empregadas duas taxas de aprendizagem. Observou-se que valores baixos da taxa de aprendizagem ofereciam bons resultados quando valores de erro de treinamento, no caso a ação de controle, eram altos. Mas quando o erro se tornava pequeno era necessário empregar uma taxa de aprendizagem maior, para compensar a lentidão no aprendizado. Assim, empregou-se uma taxa de aprendizagem da ordem de 0.03 nos instantes iniciais da trajetória, e quando a posição final do manipulador já encontra-se

[6]Os ganhos dos controladores correspondem ao do controlador PD já existente no manipulador, tendo sido definidos durante a construção do manipulador.

muito próxima do seu valor final, ou seja, quando o erro de treinamento começa a ficar pequeno, muda-se o valor da taxa de aprendizagem para 0.7.

A rede também utiliza o valor de 1 ms como período para treinamento, apresentando uma atualização constante dos valores dos pesos entre os seus neurônios.

5.3.3 Resultados dos Experimentos

Nesse item analisa-se o desempenho dos controladores PD e do controlador Neural, sem considerar as perturbações atuando sobre o manipulador.

Condições Nominais

Conforme visto em [dWSB96], o controle PD apresenta a desvantagem de não conseguir compensar os efeitos gravitacionais do manipulador, apresentando um erro em regime permanente. Essa característica foi observada nos resultados das simulações, apresentado na seção 5.2.3.

Entretanto, o manipulador SCARA Inter apresenta um sistema mecânico de compensação da gravidade atuando na junta 3 de translação, que não foi considerado no modelo utilizado para simulações. Dessa forma, é possível empregar um controlador PD, conseguindo-se obter erros bem baixos de posicionamento, mesmo sob a presença de torques gravitacionais atuando sobre o manipulador. Esse resultado pode ser observado nas figuras 5.16 e 5.17.

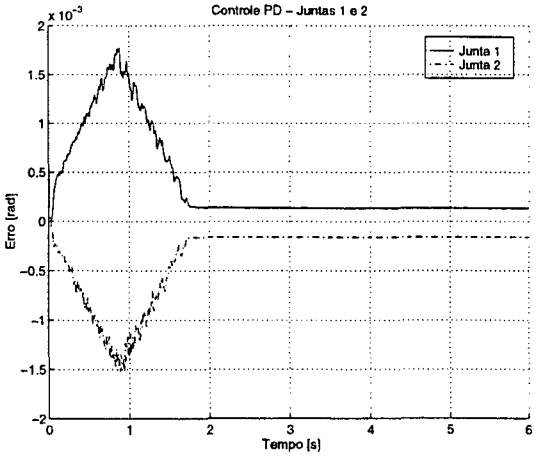

Figura 5.16: Controle PD: Condições Nominais - Junta 1 e 2

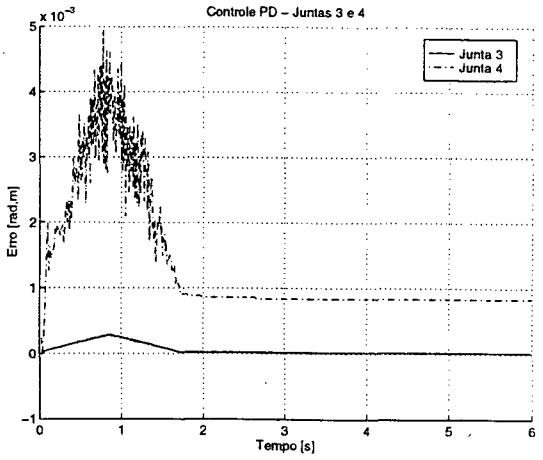

Figura 5.17: Controle PD: Condições Nominais - Junta 3 e 4

Entretanto, observa-se que não é possível atingir erro nulo de posição com o emprego de controladores PD. Em muitas aplicações, precisões da ordem de 10^{-3} podem ser

suficientes, porém em aplicações em que se deseja um posicionamento mais preciso, tal controlador não poderia ser utilizado[7].

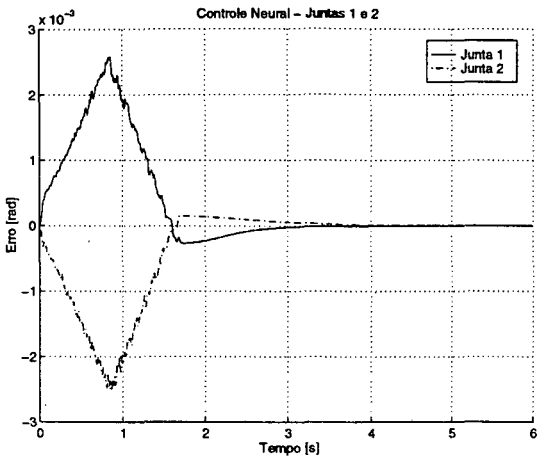

Figura 5.18: Controle Neural: Condições Nominais - Junta 1 e 2

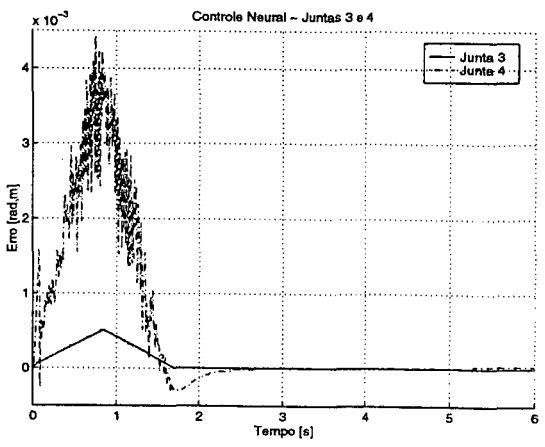

Figura 5.19: Controle Neural: Condições Nominais - Junta 3 e 4

As figuras 5.18 e 5.19 apresentam o desempenho do controlador neural. Com essa estrutura de controle é possível garantir um posicionamento mais preciso das juntas do manipulador, em regime permanente. Entretanto, durante o regime transitório, o controlador neural não é capaz de reduzir o erro de posição nas juntas do manipulador.

Observa-se que nas juntas 3 e 4 o erro de posição final encontra-se muito próximo a zero, menor que o erro de posicionamento do PD mas não nulo, pois optou-se limitar o treinamento da rede quando o valor de erro atingisse um valor muito baixo, da ordem de 10^{-5}. Esse procedimento busca evitar que a rede apresente um treinamento excessivo, em função de um valor baixo de erro mas com um número muito alto de ciclos de treinamento por segundo (1000 ciclos).

Existência de Perturbações

Para o caso em que é considerada uma perturbação atuando sobre a primeira junta do manipulador, observa-se que o controlador PD não é capaz da garantir a rejeição dessa perturbação (figuras 5.20 e 5.21).

Conforme as propriedades apresentadas do controlador PD apresentada em [dWSB96], observa-se que o controlador PD apresenta um bom desempenho de controle quando não são considerados os efeitos gravitacionais, perturbações e ruídos atuando

[7]Uma das possíveis causas do pequeno valor de erro apresentado pelo controlador PD nas figuras 5.16 e 5.17, pode estar associada às parcelas de atrito existente nas juntas

sobre o sistema. Entretanto, os erros de posição aumentam quando são considerados esses fatores, comprometendo os requisitos de controle.

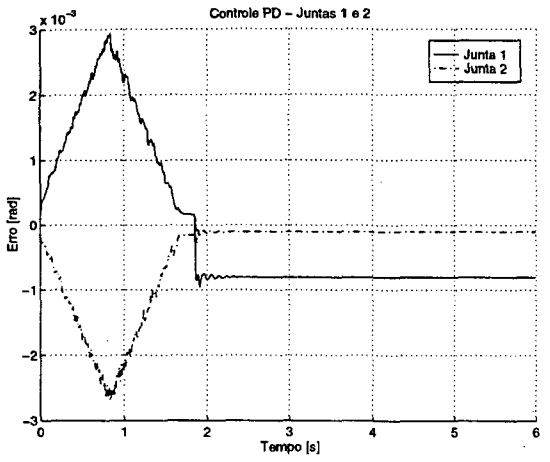

Figura 5.20: Controle PD: Perturbações - Junta 1 e 2

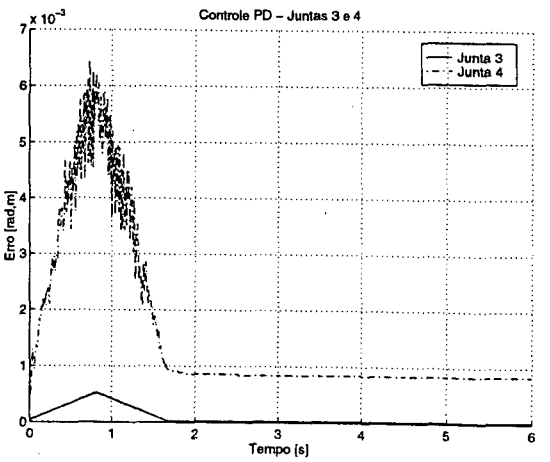

Figura 5.21: Controle PD: Perturbações - Junta 3 e 4

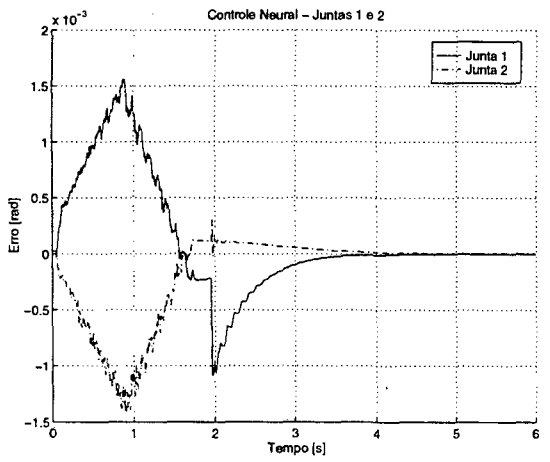

Figura 5.22: Controle Neural: Perturbações - Junta 1 e 2

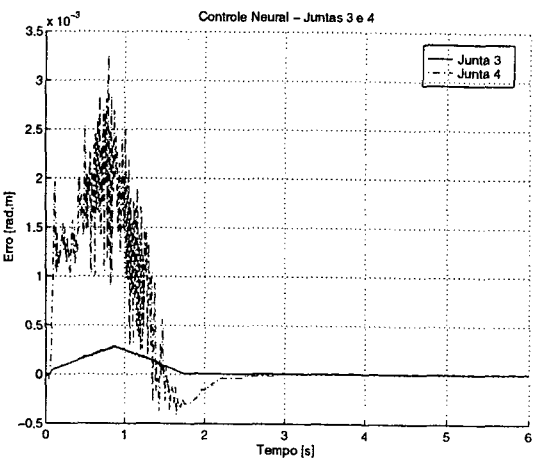

Figura 5.23: Controle Neural: Perturbações - Junta 3 e 4

Pelas figuras 5.22 e 5.23, pode-se observar que o controlador neural consegue rejeitar as perturbações atuando sobre o sistema. Uma vez que a regra de treinamento encontra-se baseada na heurística que visa minimizar a saída do controlador PD, que consiste dos erros de posição e velocidade multiplicado por ganhos, a existência de uma perturbação gera uma variação na saída do controlador. Essa variação, que corresponde ao aumento do erro de posição, gera um erro de treinamento maior, reiniciando um novo treinamento na rede, permitindo que esta seja capaz de compensar a perturbação sobre o manipulador.

Análise do Esforço de Controle

As figuras 5.24 e 5.25 apresentam os torques de controle para as juntas do manipulador, considerando o caso da existência de perturbações.

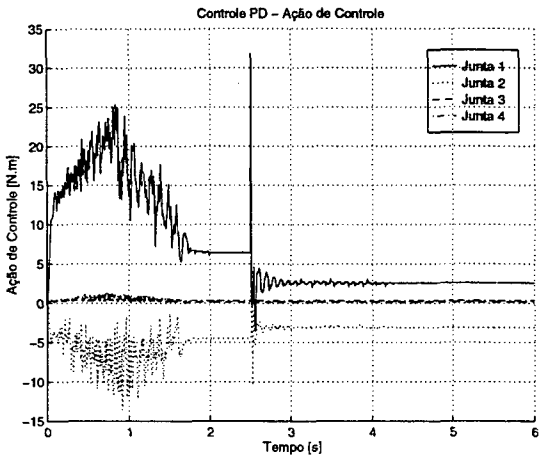

Figura 5.24: Controle PD: Ação de Controle

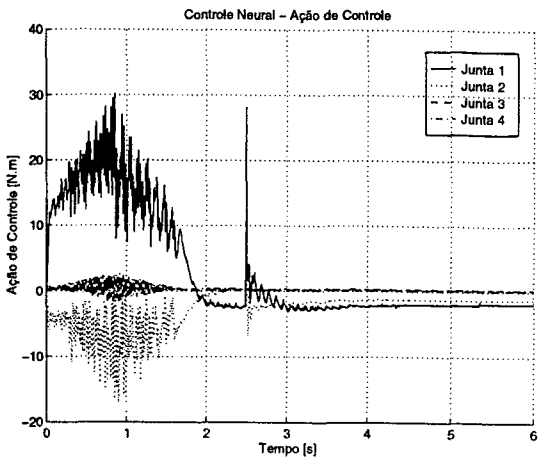

Figura 5.25: Controle Neural: Ação de Controle

Como ocorrido durante as simulações, o efeito da mudança dos pesos da rede gera uma oscilação maior na ação de controle, uma vez que a ação de controle consiste dos sinais provenientes do controlador clássico e da rede neural. Porém essa oscilação não consiste de um valor muito superior ao apresentado pela ação de controle onde somente o controlador PD é considerado.

5.4 Conclusões

Os resultados apresentados nesse capítulo comparam o desempenho de três estratégias de controle de posição para robôs manipuladores. Duas delas empregam técnicas bem conhecidas e estudadas dentro da área de robótica: controladores do tipo PD e o controle dinâmico inverso. A terceira estratégia consistiu do emprego de uma rede neural artificial como um sistema de compensação de um controlador do tipo PD.

A estratégia de controle neural procura corrigir ou compensar a ação de controle do controlador PD, nas situações em que o mesmo não é capaz de atuar, tornando o sistema robusto às incertezas paramétricas, sendo capaz de rejeitar perturbações com bastante eficiência. Além disso, a estrutura das redes empregada nesse trabalho apresentam uma topologia simples, compreendida por uma quantidade de camadas e neurônios relativamente pequenas.

Em todas as situações analisadas (simulações e implementação experimental) a estrutura de controle neural foi capaz de seguir as trajetórias com um erro mínimo de posicionamento, apresentando um desempenho superior ao dos demais controladores e a vantagem de não necessitar do conhecimento da dinâmica do manipulador para permitir a compensação de não-linearidades e perturbações.

Além disso, o fato de não ser necessária uma etapa prévia de treinamento *off-line* no esquema de controle neural proposto, constitui-se de uma vantagem do emprego de RNA's em sistemas de controle para robôs manipuladores, facilitando o emprego das redes neurais e o projeto do sistema de controle como um todo.

A implementação prática no robô SCARA do Laboratório de Robótica permitiu comprovar os resultados obtidos nas simulações. A capacidade de garantir erro nulo e compensar as não-linearidades e perturbações apresentada pela estrutura de controle neural também se fizeram presentes nos resultados experimentais obtidos.

De modo geral, o desempenho dos algoritmos de controle utilizando redes neurais foi superior ao controlador PD clássico, quando o sistema está sujeito à perturbações externas ou variações paramétricas. É importante ressaltar que o robô SCARA Inter, por ser um robô de pesquisa, possui pequenos efeitos como folgas, zonas mortas, etc. Entretanto, esses efeitos são mais significativos em robôs industriais e, neste caso, o emprego de algoritmos robustos devem apresentar resultados ainda melhores, quando comparados aos controladores clássicos.

Capítulo 6

Resultados: Controle Híbrido de Força/Posição

6.1 Introdução

Baseado nas equações dinâmicas apresentado na seção 2.3.3, serão apresentados os resultados das simulações para o controle híbrido de força/posição do robô SCARA, empregando os três tipos diferentes de estrutura de controle, vistos no capítulo 4:

- estrutura de controle híbrido dinâmico, descrito em [Yos90];
- estratégia de controle híbrida neural utilizando o modelo dinâmico do manipulador, derivada da estratégia acima, e
- estratégia de controle híbrida neural sem considerar o modelo dinâmico do manipulador.

Foram utilizadas somente as três primeiras juntas do manipulador, desconsiderando a quarta junta, responsável pela definição da orientação do efetuador final.

Os controladores PD e PID utilizados correspondem a implementações discretas dos controladores contínuos apresentados no capítulo 4.

Análise do desempenho das estruturas de controle são feitas, considerando os casos nominais, existência de perturbações, incertezas e variações paramétricas nos coeficientes da equação do manipulador.

Novamente foram empregados os softwares MATLAB 4.0 e SIMULINK 1.3c para realização das simulações.

6.2 Especificação da Tarefa

O projeto do controlador híbrido dinâmico envolve um estudo da geometria do ambiente onde o manipulador vai executar a tarefa. A necessidade do conhecimento das direções de movimento livre e restrito é uma imposição na abordagem híbrida de controle de força. Como pode ser visto nas equações que descrevem a lei de controle - seção 4.4, o conhecimento explícito da geometria da superfície em contato está incluído diretamente nas equações, através das matrizes de seleção E_P e E_F.

Após a definição e especificação da tarefa a ser desenvolvida pelo manipulador, segue-se a fase de ajuste dos controladores.

O período de amostragem utilizado foi o mesmo empregado no caso do controle de posição, ou seja, 2.5 ms.

A tarefa a ser executada pelo manipulador consiste na livre movimentação nas direções $x-y$, exercendo uma força perpendicular ao plano xy. Considera-se a existência de um objeto, considerado infinitamente rígido, a uma distância z_e da origem do sistema de coordenadas do espaço operacional (colocado no elo da base) - figura 6.1.

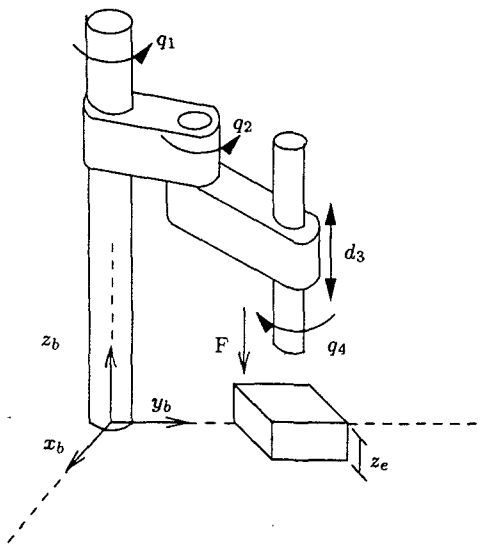

Figura 6.1: Especificação da Tarefa

O sistema de coordenadas r empregado para descrever as restrições do manipulador, é dado pelo sistema de coordenadas do espaço operacional, ou seja, $r = [x\ y\ z]$.

A restrição imposta pelo meio ao movimento do manipulador é dada pela seguinte relação:

$$p_1(r) = z - z_e \tag{6.1}$$

que representa a existência da superfície de um objeto na posição $z = z_e$.

De acordo com a equação 4.11, a matriz de seleção E_F será dada por

$$E_F = [0\,0\,1]\,. \tag{6.2}$$

Como os subespaços de controle de força e posição são complementares, ou seja, $[0_{n-m}\ E_F] + [E_P\ 0_m] = I_n$, onde n é o número de graus de liberdade do manipulador e m é o número de restrições, obtém-se como matriz de seleção para o subespaço de posição

$$E_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{6.3}$$

As posições y_P no subespaço de posição são dados por

$$y_P = E_P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{6.4}$$

e a força no subespaço de força é obtida através do emprego do mapeamento

$$f_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} E^{-T} Q_r^T \tag{6.5}$$

onde

$$E = \begin{bmatrix} E_P \\ E_F \end{bmatrix}. \tag{6.6}$$

Uma vez que o sistema de coordenadas empregado para descrever as restrições corresponde ao sistema de coordenadas do espaço operacional, a matriz de transformação Q_r é igual a identidade.

O sensor de força foi modelado como sendo uma mola de rigidez K, capaz de medir as forças nos três eixos cartesianos. Os momentos existentes em cada um desses eixos coordenados, decorrentes dos torques aplicados, foram desprezados. O comportamento do sensor é dado por

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \end{bmatrix}. \tag{6.7}$$

onde $K_{(.)}$ corresponde à rigidez do sensor nas direções x, y e z, e $\delta_{(.)}$ representa os deslocamentos nos elementos de medição do sensor, quando em contato com uma superfície, também em cada uma das direções dos eixos coordenados.

Foi empregado um valor de 10^4 N/m como valor da rigidez K_z do sensor de força. Na estratégia de controle híbrido de força/posição, as medidas de força no subespaço de posição não são consideradas.

Como pode-se observar, a mudança da geometria do meio exige a reformulação da estrutura de controle, uma vez que suas equações encontram-se baseadas diretamente nas matrizes de seleção E_P e E_F.

6.3 Trajetórias de Referência

As trajetórias de referência de posição são definidas por funções suaves e duplamente diferenciáveis, enquanto que a referência de força é dada por uma sequência de degraus.

Para realizar o mapeamento das variáveis do espaço cartesiano para o espaço de juntas, faz-se necessário o emprego da inversa do Jacobiano. Nesses casos, existem pontos no espaço de trabalho do manipulador em que a inversa do Jacobiano não existe, correspondendo às singularidades. Quando o manipulador se aproxima ou passa por uma dessas singularidades, existe uma perda de graus de liberdade, ocasionando a perda de estabilidade do controlador, em função da não existência de solução para a cinemática inversa do manipulador. Dessa forma, a especificação das posições iniciais e das trajetórias a serem seguidas pelo manipulador deve ser feita de forma a evitar a proximidade dos pontos singulares.

As trajetórias de referência para posição são dadas por:

$$x_d = x_0 - 0.2 + 0.05 \sin(4t) \tag{6.8}$$

$$y_d = y_0 - 0.2 - 0.05 \sin(3t) \tag{6.9}$$

onde x_0 e y_0 são as posições iniciais do efetuador final, dadas no sistema de coordenadas do espaço operacional. A referência de força consiste em

$$f_{Fd} = \begin{cases} 5N & \text{para } t \leq 1.5 \text{ s} \\ 20N & \text{para } 1.5 < t \leq 3 \text{ s} \\ 13N & \text{para } t > 3 \text{ s.} \end{cases} \tag{6.10}$$

De forma a evitar o problema de tratamento de colisões e impactos, devido à aproximação do manipulador à uma superfície de contato, procura-se posicionar o efetuador final do manipulador suficientemente próximo a essa superfície.

A posição inicial do manipulador no espaço cartesiano é dada por $r_0 = [0.413\ 0.253\ 0.42]^T$ em m. Para a direção do movimento restrito, eixo z, considera-se a existência de um objeto na posição $z_e = 0.4$ m.

O período de amostragem empregado foi de 2.5 ms.

A análise de desempenho dos controladores é feita em duas situações distintas:

Valores Nominais + Perturbação: Assume-se que os valores estimados dos parâmetros do manipulador são exatos, estando sujeito a perturbações externas proveniente do ambiente.

Variação Paramétrica: Considera-se que os parâmetros estimados possuem incertezas, não correspondendo aos valores reais do manipulador.

6.4 Projeto dos Controladores

A escolha dos ganhos dos controladores foi feita baseada nos mesmos critérios utilizado no caso de controle de posição. Ou seja, considerando somente as condições nominais de operação do manipulador, os ganhos dos controladores são escolhidos de forma que cada uma das estratégias de controle, tomadas individualmente, apresentem os melhores desempenhos de resposta. O objetivo, portanto, não consiste da análise direta dos controladores entre si, o que exigiria um conjunto de parâmetros iguais para as três estratégias (ganho dos controladores, taxas de aprendizado e topologia de rede), mas sim do desempenho de cada estrutura em separado, frente às variações e imprecisões paramétricas e perturbações que influenciam o comportamento dinâmico do manipulador.

6.4.1 Controle Híbrido Dinâmico

A lei de controle híbrido para um manipulador sem redundância é dada por [Yos90]:

$$\tau = \tau_P + \tau_F \tag{6.11}$$

$$\tau_P = \hat{H}(q)\ddot{q}_d + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) \tag{6.12}$$

$$\ddot{q}_d = J^{-1}\left\{E^{-1}\left(\begin{bmatrix} u_1 \\ 0 \end{bmatrix} - \dot{E}J\dot{q}\right) - \dot{J}\dot{q}\right\} \tag{6.13}$$

$$\tau_F = J^T E_F^T u_2 \tag{6.14}$$

onde $E = E_F {E_P}^T$, e u_1 é um vetor de dimensão $(6-m)$ e u_2 possui dimensão m e

correspondem à sinais de controle provenientes de controladores PD e PID, respectivamente. As matrizes $\hat{(\cdot)}$ são as matrizes estimadas da equação dinâmica do manipulador, contendo os valores nominais dos parâmetros do manipulador.

A equação do controlador PD na malha de posição é dada por:

$$u_1(k) = K_{Ppi}e_{pi}(k) + \frac{K_{Dpi}}{T}\left[e_{pi}(k) - e_{pi}(k-1)\right] \tag{6.15}$$

onde $e_{pi} = y_{Pdi} - y_{Pi}$, com y_{Pdi} correspondendo às referências de posição, para $i = 1, 2$.

A equação do controlador PID é dada por

$$u_0(k) = u_0(k-1) + p_0e_f(k) + p_1e_f(k-1) + p_2e_f(k-2) \tag{6.16}$$

com

$$\begin{aligned} p_0 &= K_{Pf}\left(1 + \frac{K_{Df}}{T}\right) \\ p_1 &= -K_{Pf}\left(1 + 2\frac{K_{Df}}{T} - K_{If}T\right) \\ p_2 &= \frac{K_{Pf}K_{Df}}{T} \end{aligned} \tag{6.17}$$

onde T é o valor do período de amostragem e e_f é o erro de força calculado no subespaço de força.

Os ganhos obtidos para os controladores PD na malha de posição correspondem à:

$$\begin{aligned} K_{Pp} &= \begin{bmatrix} 80 & 0 \\ 0 & 70 \end{bmatrix} \\ K_{Dp} &= \begin{bmatrix} 40 & 0 \\ 0 & 30 \end{bmatrix} \end{aligned} \tag{6.18}$$

e para o controlador PID na malha de força

$$\begin{aligned} K_{Pf} &= 5 \\ K_{Df} &= 0.3 \\ K_{If} &= 0.2 \end{aligned} \tag{6.19}$$

Os ganhos dos controladores foram ajustados de forma a obter-se o melhor desempenho para o caso nominal, negligenciando as perturbações e incertezas nos parâmetros. Empregou-se a aproximação retangular para aproximar o termo integral no controlador PID.

6.4.2 Controle Híbrido Neural e Controle Híbrido Neural sem Modelo Dinâmico

O controlador híbrido neural possui a seguinte lei de controle - conforme seção 4.4.2:

$$\begin{aligned} \tau &= \hat{M}(q)\ddot{q}_d + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) + \tau_F \\ \ddot{q}_d &= J^{-1}\left\{E^{-1}\left(\begin{bmatrix} u_1 + \phi_P \\ 0 \end{bmatrix} - \dot{E}J\dot{q}\right) - \dot{J}\dot{q}\right\} \\ \tau_F &= J^T E_F^T(u_2 + \phi_F). \end{aligned} \tag{6.20}$$

O controlador híbrido neural sem modelo dinâmico apresenta como lei de controle, conforme visto em 4.4.2, a seguinte expressão:

$$\begin{aligned} \tau &= \ddot{q}_d + \tau_F + \hat{G}(q) \\ \ddot{q}_d &= J^{-1}\left\{E^{-1}\left(\begin{bmatrix} u_1 + \phi_P \\ 0 \end{bmatrix} - \dot{E}J\dot{q}\right) - \dot{J}\dot{q}\right\} \\ \tau_F &= J^T E_F^T(u_2 + \phi_F) \end{aligned} \tag{6.21}$$

Os sinais de controle u_1 e u_2 correspondem as implementações digitais dos controladores PD, do tipo

$$\begin{aligned} u_1(k) &= K_{Pp}e_y(k) + K_{Dp}\left(\frac{e_y(k) - e_y(k-1)}{T}\right) \\ u_2(k) &= K_{Pf}e_f(k) + K_{Df}\left(\frac{e_f(k) - e_f(k-1)}{T}\right) \end{aligned} \tag{6.22}$$

onde e_y e e_f são os erros de posição e força, respectivamente.

O projeto das redes neurais ϕ_P e ϕ_F para as duas propostas de controle neural são semelhantes. As considerações feitas a respeito do projeto da estrutura das redes neurais são válidas para os dois métodos de controle neural. Além disso, as topologias obtidas para as redes são similares para as duas propostas.

O projeto do controlador digital neural consiste de duas fases distintas: o ajuste dos controladores PD e a definição da topologia das duas redes neurais.

Os problemas e dificuldades enfrentados no caso do emprego de RNA´s no controle de juntas, também se repetiram para o caso do controle de força/posição.

A definição de uma quantidade suficiente de camadas e neurônios de forma a manter um compromisso entre capacidade de aprendizagem e desempenho de treinamento também foram levadas em consideração na etapa de ajuste das redes.

Como ocorrido no caso do controle de posição, os algoritmos de aprendizagem acelerados, *quickpropagation* e o *RPROP*, também apresentaram um comportamento indesejável no desempenho de controle, saturando a saída da rede em função do elevado valor de atualização dos pesos, ocasionando a instabilidade do sistema de controle. O algoritmo *backpropagation* foi empregado, uma vez em que a necessidade de um aprendizado mais lento, gradual e que não gerasse uma saturação na saída da rede nos instantes iniciais de controle se fazia necessário.

As redes neurais são implementadas conforme a equação 4.25. O controlador PID na malha de força é substituído por um controlador do tipo PD, conforme visto na seção 4.4.2.

As redes na malha de posição, ϕ_P, e na malha de força, ϕ_F, empregadas nas estruturas de controle são do tipo MLP.

Tanto a rede ϕ_P, quanto a rede ϕ_F, possuem o mesmo número de neurônios na camada de entrada. Ela contém 9 neurônios - três para aos valores de posição q, três para os valores de velocidade nas juntas $\dot{q}$, e mais três neurônios para os valores de aceleração $\ddot{q}$.

O mapeamento na malha de força apresenta uma complexidade menor, se comparado com a malha de posição. Dessa forma, a estrutura de rede é mais simples que a obtida para a rede da malha de posição. A estrutura das duas redes pode ser observada nas tabelas 6.1 e 6.2.

Camada	*Número de Neurônios*
Camada de Entrada	9
Camada Intermediária 1	7
Camada Intermediária 2	5
Camada de Saída	2

Tabela 6.1: Estrutura de Rede da Malha de Posição

Camada	*Número de Neurônios*
Camada de Entrada	9
Camada Intermediária	6
Camada de Saída	1

Tabela 6.2: Estrutura de Rede da Malha de Força

Em ambas as redes, as funções de ativação dos neurônios são do tipo tangente hiperbólica, com inclinação igual a um. Os motivos da escolha da função tangente

hiperbólica foram os mesmos apresentados na escolha dessa função para as redes neurais empregadas no controle de posição, apresentado no capítulo anterior.

Como ocorrido com o caso de controle neural de posição, os problemas decorrentes da escolha de valores não suficientemente pequenos para inicialização dos pesos das redes, também foram observados na estratégia de controle neural de força/posição. Dessa forma, as matrizes de pesos e bias foram inicializadas com valores aleatórios compreendidos entre $\pm$ 0.05. A taxa de aprendizagem do algoritmo *backpropagation* foi da ordem de 0.0005 para a rede na malha de posição e 0.001 para a rede na malha de força.

Os controladores foram ajustados de forma a apresentar a melhor resposta para o caso nominal. Da mesma forma que procedido no projeto do controlador híbrido dinâmico, foram desprezadas as perturbações, incertezas e variações paramétricas.

Os ganhos obtidos para os controladores PD na estratégia de controle híbrido neural foram:

$$\begin{aligned} K_{Pp} &= \begin{bmatrix} 80 & 0 \\ 0 & 70 \end{bmatrix} \\ K_{Dp} &= \begin{bmatrix} 40 & 0 \\ 0 & 30 \end{bmatrix} \end{aligned} \tag{6.23}$$

que correspondem aos mesmos ganhos empregados no controle híbrido dinâmico.

Para os ganhos do controlador PD da malha de força, os ganhos obtidos corresponderam à:

$$\begin{aligned} K_{Pf} &= 5 \\ K_{Df} &= 0.1 \end{aligned} \tag{6.24}$$

Os ganhos obtidos para os controladores PD na estratégia de controle híbrido neural sem modelo dinâmico foram:

$$\begin{aligned} K_{Pp} &= \begin{bmatrix} 40 & 0 \\ 0 & 35 \end{bmatrix} \\ K_{Dp} &= \begin{bmatrix} 30 & 0 \\ 0 & 25 \end{bmatrix} \end{aligned} \tag{6.25}$$

para o controlador PD na malha de posição, e

$$\begin{aligned} K_{Pf} &= 5 \\ K_{Df} &= 0.1 \end{aligned} \tag{6.26}$$

para o controlador PD na malha de força.

6.5 Resultado das Simulações

6.5.1 Condições Nominais + Perturbações

Considerando que as estimações dos parâmetros do manipulador correspondem aos valores reais exatos, analisa-se o comportamento dos controladores frente a existência de perturbações atuando sobre a dinâmica do robô.

As perturbações consistem de torques contrários ao movimento do manipulador, de valor constante, aplicado em instantes de tempo específicos. O objetivo consiste em verificar a capacidade de rejeição à perturbações dos controladores empregados. Dessa forma, aplicou-se um sinal de perturbação na junta 1 do manipulador, da ordem de 45% da torque máximo suportado por essa junta, no instante 2.5 s. Uma segunda perturbação é introduzida à junta 3, através de outro sinal constante de torque, correspondendo a 5% do valor de torque máximo suportado pela mesma, no instante 3 s.

As figuras 6.2, 6.3 e 6.4 apresentam os resultados obtidos no controle de posição.

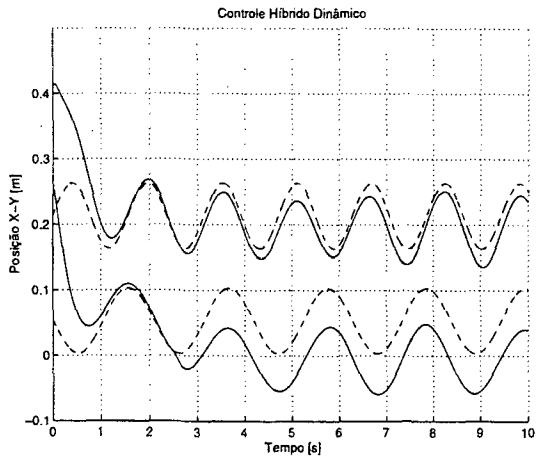

Figura 6.2: Controle Híbrido Dinâmico: Posição - Caso Nominal com Perturbação

A estrutura de controle híbrida dinâmica apresenta um bom desempenho nos instantes iniciais da trajetória. Entretanto, quando a perturbação é aplicada ao sistema,

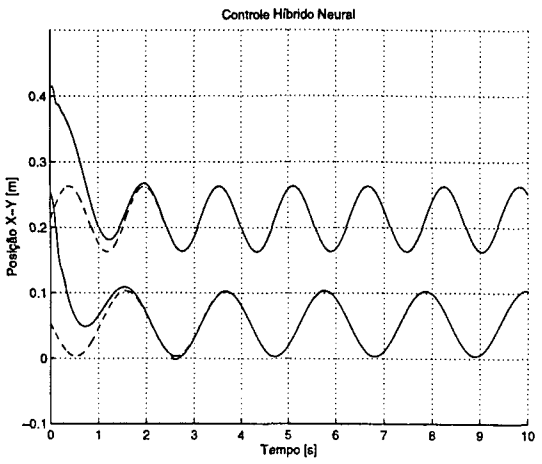

Figura 6.3: Controle Híbrido Neural: Posição - Caso Nominal com Perturbação

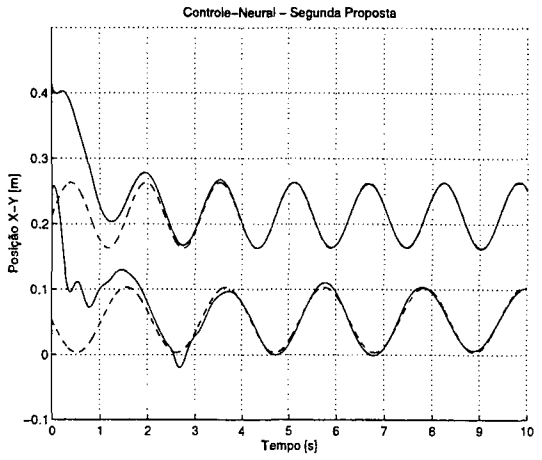

Figura 6.4: Controle Híbrido Neural sem Modelo Dinâmico - Caso Nominal com Perturbação

a estrutura de controle não é capaz de corrigir o erro - figura 6.2. Isso se deve a dois fatores. O primeiro está baseado no fato de que a estratégia de controle inverso busca linearizar e desacoplar o modelo do manipulador, transformando-o em um conjunto independente de n duplo integradores. Entretanto, quando são consideradas as incertezas paramétricas, tal linearização não é assegurada. Dessa forma, o desempenho de controle é prejudicada. O segundo fator consiste do emprego de uma estrutura de controle do tipo PD que não é capaz de rejeitar perturbação. Esse problema poderia ser minimizado com uma estrutura do tipo PID.

Como pode ser visto na figura 6.3, a estrutura de controle híbrido neural apresenta um bom seguimento de trajetória, tornando imperceptível a existência da perturbação sobre a dinâmica do robô.

Na estrutura controle híbrido neural sem modelo dinâmico, existe uma fase inicial

em que as posições $x - y$ apresentam um comportamento mais oscilatório, em comparação aos demais controladores - figura 6.4. Isso se deve ao fato de que nessa proposta, o modelo dinâmico do manipulador não se encontra incluído na lei de controle. Dessa forma, a rede neural necessita de um tempo maior de ajuste dos seus parâmetros. Entretanto, tal estrutura é capaz de garantir o seguimento de trajetória com um erro menor que a estrutura de controle híbrido dinâmico.

As figuras 6.5, 6.6 e 6.7 apresentam o desempenho dos controladores durante a garantia da referência de força.

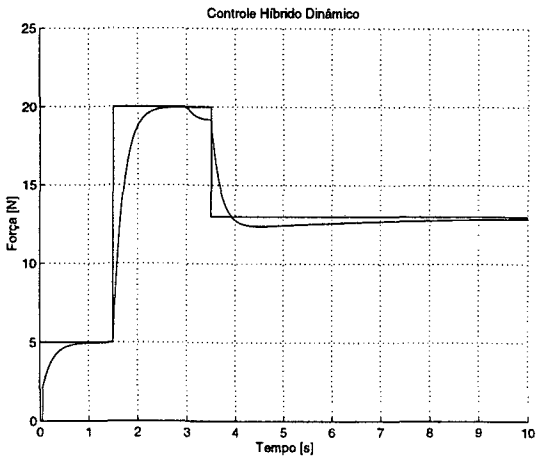

Figura 6.5: Controle Híbrido Dinâmico: Força - Caso Nominal com Perturbação

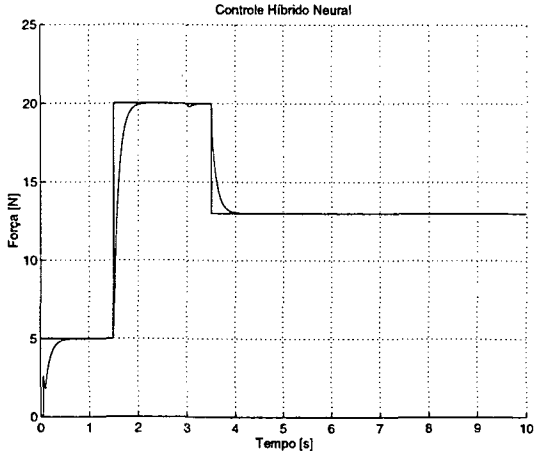

Figura 6.6: Controle Híbrido Neural: Força - Caso Nominal com Perturbação

Ao contrário da malha de posição, na malha de força do controlador híbrido dinâmico a perturbação é rejeitada devido ao emprego de um controlador do tipo PID. Entretanto, o tempo de resposta para corrigir a trajetória é muito lento - figura 6.5.

Conforme as figuras 6.6 e 6.7, observa-se que as redes neurais apresentam uma razoável capacidade de rejeição às perturbações. O fato das redes utilizarem a saída dos

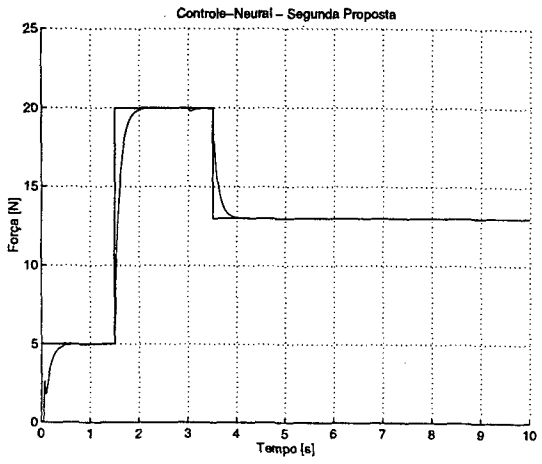

Figura 6.7: Controle Híbrido Neural sem Modelo Dinâmico: Força - Caso Nominal com Perturbação

controladores de posição e força como sinal de treinamento, implica em que qualquer variação brusca na saída desses controladores, como ocorre com a aplicação de uma perturbação, a rede entenda essa variação como um sinal de erro e a faça rapidamente se ajustar às novas condições de operação. Dessa forma, em situações em que o controlador PID era lento, a rede o substitui e garante uma convergência mais rápida para o valor de referência. Observa-se também que os controladores neurais apresentam um tempo de resposta ao controle de força relativamente menor que a estrutura de controle híbrido dinâmico.

6.5.2 Variação Paramétrica

A maior desvantagem do controlador híbrido dinâmico consiste na necessidade do conhecimento exato do modelo do manipulador. Imprecisões no conhecimento dos parâmetros podem prejudicar o desempenho do controlador.

Dessa forma, verificou-se o desempenho dos controladores de força/posição frente as variações paramétricas nos coeficientes do robô manipulador - tabela 6.3.

As figuras 6.8, 6.9 e 6.10 apresenta o resultado das simulações para o caso de controle de posição.

Nos três casos a variação paramétrica não afeta consideravelmente o desempenho dos controladores. Entretanto, observa-se que o controlador híbrido dinâmico apresenta um erro pequeno em regime permanente, fato que não ocorre nos controladores neurais, que apresentam um erro em regime nulo. Essa fato é decorrente da capacidade em que a estruturas de controle neurais empregadas possuem de compensar a atuação do controlador principal. Desse modo, caso o modelo da equação 4.19 apresente impre-

Parâmetro	Variação
m_1	+ 20%
m_3	- 20%
m_4	+ 30%
I_1	+ 15%
I_3	- 15%
I_4	+ 10%
l_1	+ 30%
l_2	- 30%

Tabela 6.3: Variações Paramétricas

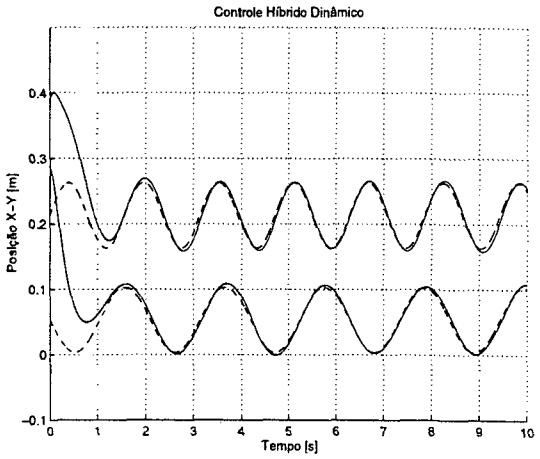

Figura 6.8: Controle Híbrido Dinâmico: Posição - Caso Variação Paramétrica

cisões paramétricas ou simplificações do modelo real do manipulador, as redes neurais são capazes de anular os efeitos decorrentes da submodelagem e falta de conhecimento do manipulador, atuando como um sistema de compensação.

As figuras 6.11, 6.12 e 6.13 apresentam o resultado das simulações para o caso de controle de força.

As diferenças existentes entre os parâmetros estimados para o manipulador e os seus valores reais foram suficientes para prejudicar o desempenho do controlador de força - figura 6.11.

Novamente as estruturas de controle neural apresentaram um bom desempenho, garantindo erro nulo em regime, dentro dos requisitos de tempo especificado - figuras 6.12 e 6.13.

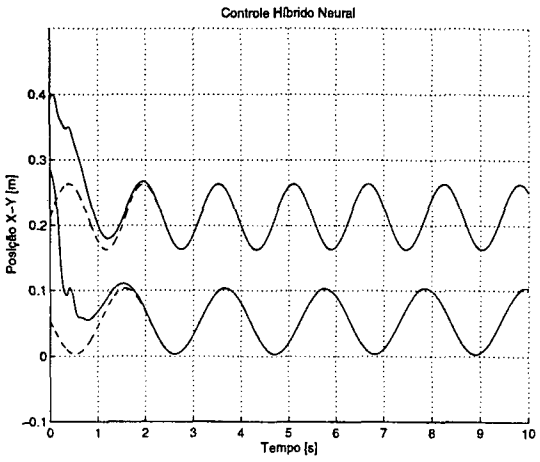

Figura 6.9: Controle Híbrido Neural: Posição - Caso Variação Paramétrica

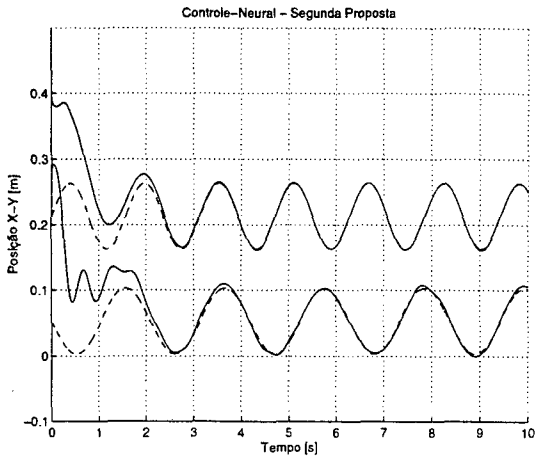

Figura 6.10: Controle Híbrido Neural sem Modelo Dinâmico: Posição - Caso Variação Paramétrica

6.5.3 Análise do Esforço de Controle

As figuras 6.14, 6.15 e 6.16 apresentam as ações de controle para o caso onde existem perturbações atuando sobre o manipulador. Observa-se que os controladores neurais apresentam oscilações maiores nos transitórios. Isso se deve ao fato da necessidade da rede ajustar seus pesos para uma nova situação, quando da ocorrência de mudanças bruscas de referência ou da existência de perturbações.

Quando a perturbação é aplicada ao manipulador, os controladores PD necessitam gerar uma ação de controle capaz de garantir o seguimento da trajetória - figura 6.14. Como a perturbação gera um erro de posicionamento, geralmente com uma diminuição brusca da sua movimentação, o controlador PD apresenta um comportamento com um sinal de amplitude elevada no momento da aplicação da perturbação, devido ao termo derivativo. Esse sinal é interpretado pela rede como um erro de amplitude elevado,

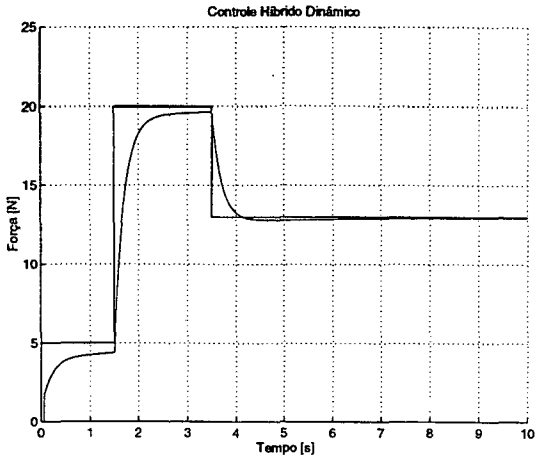

Figura 6.11: Controle Híbrido Dinâmico: Força - Caso Variação Paramétrica

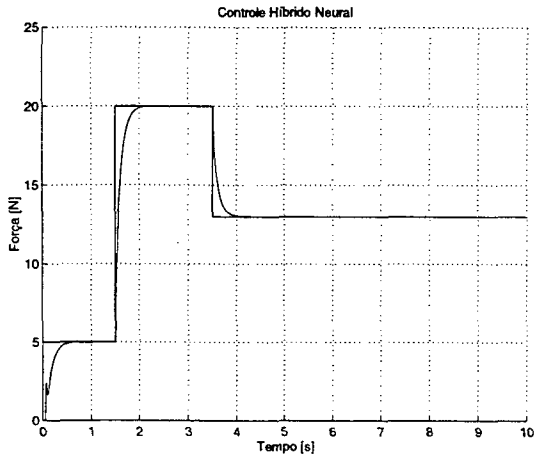

Figura 6.12: Controle Híbrido Neural: Força - Caso Variação Paramétrica

gerando um valor de atualização dos pesos também elevado, ocasionando uma alteração mais significativa na saída da rede, e podendo gerar um comportamento oscilatório.

Além disso, a estrutura neural necessita de um tempo para reajustar os seus pesos quando novas condições ocorrem. Dessa forma, a ação de controle se torna mais oscilatória, em comparação com a do controlador híbrido dinâmico implementado - figuras 6.15 e 6.16.

Na proposta de controle híbrido neural sem modelo dinâmico, ainda observa-se um comportamento oscilatório da ação de controle, durante os momentos iniciais do seguimento da trajetória - figura 6.16. Isso se deve ao fato da rede ainda não ser capaz de compensar o modelo dinâmico do manipulador. Nas propostas de controle híbrido dinâmico e controle híbrido neural tal característica não se faz presente, uma vez que a utilização do modelo dinâmico do manipulador na lei de controle garante uma ação de controle mais suave, devido aos desacoplamentos e linearizações que a lei de controle

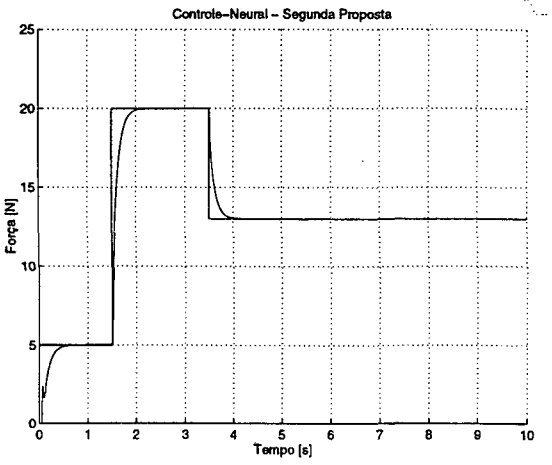

Figura 6.13: Controle Híbrido Neural sem Modelo Dinâmico: Força - Caso Variação Paramétrica

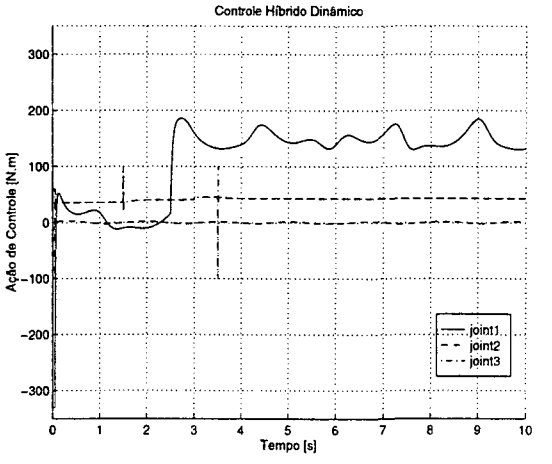

Figura 6.14: Controle Híbrido Dinâmico: Ação de Controle

proporciona. Entretanto, estas estratégias apresentam a desvantagem da necessidade da dedução analítica completa da equação dinâmica do manipulador.

6.5.4 Análise de Erro

As tabelas 6.4 e 6.5 apresentam uma análise qualitativa do desempenho dos controladores utilizados. O critério de erro empregado é o da equação 5.8, mas utilizando-se as variáveis de posição X-Y e o valor da força, ao invés da posição nas juntas.

Observa-se que para o critério empregado, as duas estruturas de controle neural utilizadas, apresentaram um valor de erro menor que o controlador híbrido dinâmico (cerca de 50% menor). Dessa forma, fica evidenciado a capacidade das estruturas neurais de compensar as perturbações e variações paramétricas, e, para o controlador híbrido neural sem modelo dinâmico, a capacidade de compensar o modelo completo

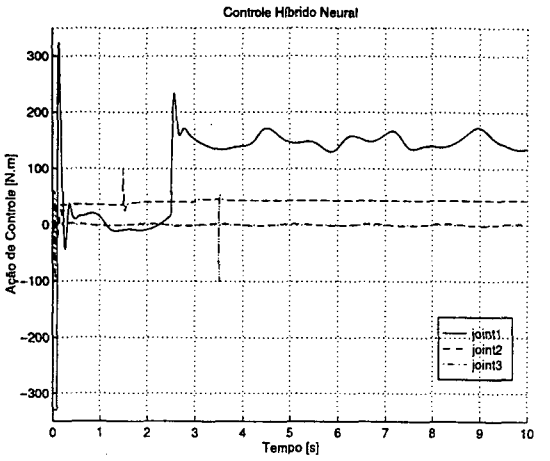

Figura 6.15: Controle Híbrido Neural: Ação de Controle

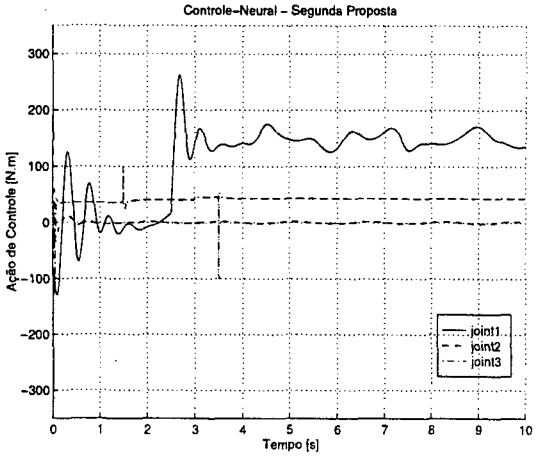

Figura 6.16: Controle Híbrido Neural sem Modelo Dinâmico: Ação de Controle

do manipulador.

6.6 Conclusões

A maior desvantagem apresentada pela estrutura de controle apresentada por [Yos90], consiste da necessidade do conhecimento das matrizes que compõem o modelo dinâmico do manipulador. Na primeira proposta de controle neural de força/posição apresentada nesse trabalho, também se faz necessário o conhecimento da estrutura e dos valores dos parâmetros das equações do manipulador. Entretanto, o desempenho do controlador neural não é afetada quando são considerados erros de modelagem ou incertezas paramétricas, e também a existência de perturbações atuando sobre o manipulador.

Porém essas duas estratégias de controle requerem a dedução analítica completa

Nominal com Perturbação	Híbrido	Neural	Neural-II
Posição X (m)	0.0239	0.0102	0.0158
Posição Y (m)	0.0515	0.0091	0.0173
Força (N)	0.7108	0.2863	0.2863

Tabela 6.4: Erro: Caso Nominal e Perturbação

Variação Paramétrica	Híbrido	Neural	Neural-II
Posição X (m)	0.0133	0.0094	0.0139
Posição Y (m)	0.0131	0.0104	0.0187
Força (N)	0.7542	0.2898	0.2903

Tabela 6.5: Erro: Caso Variação Paramétrica

da equação dinâmica do manipulador. Esse problema pode ser contornado com o emprego de uma estrutura de controle que seja capaz de compensar de forma *on-line* (sem treinamento prévio) o modelo dinâmico do manipulador, ou seja, a estrutura de controle não necessita da inclusão das matrizes dinâmicas do manipulador. Dessa forma o controlador passa a atuar como um compensador mais geral, encarregado de aprender toda a dinâmica do manipulador, ao invés de somente ajustar valores nominais já conhecidos pela lei de controle, como é o caso da primeira estrutura de controle neural de força/posição. O controlador híbrido neural sem modelo dinâmico apresenta essa característica.

Além disso, essa estratégia de controle neural somente necessita do conhecimento da geometria do meio (direções de movimento livre e restrito), do Jacobiano analítico e do termo de gravidade para realizar o controle de força/posição de robôs manipuladores, e com um desempenho superior ao controlador híbrido dinâmico proposto em [dWSB96].

Observa-se, entretanto, que o desempenho do controlador híbrido neural sem modelo dinâmico é inferior ao do controlador híbrido neural. Tal fato acontece, uma vez que a segunda estrutura de controle apresenta um papel de maior responsabilidade dentro do sistema de controle. Enquanto o primeiro controlador neural era responsável em compensar os termos não previstos por um modelo paramétrico estimado, na segunda proposta a rede além de compensar as não-linearidades, perturbações atuantes sobre o manipulador, ela tem que ser capaz de compensar os termos de Coriolís e torques centrífugos e a própria matriz de inércia.

O inconveniente apresentado pelas duas estruturas de controle neural consiste na ação de controle mais oscilatória, decorrente da necessidade de constante ajuste dos pesos de suas conexões, e na falta de critérios formais para análise e projeto dos controladores neurais.

Capítulo 7

Conclusões e Perspectivas

7.1 Conclusões

O emprego de estruturas adaptativas em controle de robôs é uma alternativa às técnicas de controle convencionais, que necessitam do conhecimento analítico do modelo do manipulador. A vantagem do emprego de redes neurais em controle de robôs manipuladores, reside justamente na não necessidade do conhecimento da planta, sendo capaz de aprender o modelo dinâmico do manipulador, incluindo as não-linearidades, e também as incertezas estruturais e não-estruturais.

Nas tarefas de controle de robôs manipuladores, os controladores clássicos do tipo PD/PID não são suficientes, pois não apresentam boas propriedades frente à existências de perturbações e variações paramétricas. Estratégias de controle que incluem um modelo paramétrico do robô em sua estrutura, como o controle inverso de juntas ou o controle híbrido de força/posição, procuram aumentar o desempenho da sua estrutura de controle. Porém, essas estratégias necessitam de um conhecimento mais preciso sobre os parâmetros do modelo do manipulador. A falta de precisão no modelo estimado empregado na lei de controle, pode levar o sistema a apresentar um desempenho fora das especificações. Dessa forma, uma rede neural que complemente a ação de um controlador principal, assumindo o papel de compensar não-linearidades e perturbações não-modeladas atuantes sobre a dinâmica do manipulador, apresentou-se como uma solução adequada à falta de conhecimento sobre essa dinâmica.

As estruturas de controle neural propostas nesse trabalho não necessitam da etapa prévia de treinamento *off-line*, dispensando o levantamento de um conjunto de exemplos, que para o caso do manipulador robótico, constituiria-se de um número elevado de dados, exigindo o emprego de algoritmos de treinamento mais rápidos e eficientes e estruturas de redes maiores e mais complexas. Assim, foi possível empregar redes com

número relativamente baixo de camadas e neurônios, para o controle de uma planta com a complexidade apresentada pelos robôs manipuladores: não-linearidades, alto grau de acoplamento entre suas variáveis, perturbações externas, dificuldade de levantamento dos parâmetros e coeficientes do seu modelo e estrutura multivariável.

O emprego de uma estrutura de controle híbrida de força/posição que necessite somente do conhecimento do Jacobiano do manipulador, e do termo de compensação de gravidade, dispensando a complexa dedução analítica do modelo dinâmico completo do manipulador, e que apresente um desempenho de controle superior ao controle híbrido dinâmico tradicional, representa uma alternativa viável na área de controle de força/posição de robôs manipuladores.

Com relação à outras estruturas de controle neural de juntas e de controle híbrido de força/posição em robôs manipuladores, de maneira geral, as estratégias propostas apresentam as seguintes vantagens:

- Não necessidade do treinamento *off-line*, não sendo necessário, portanto, o levantamento de um conjunto de exemplos.
- Não necessidade do conhecimento do modelo da planta (controle neural de posição e controle híbrido neural sem modelo dinâmico).
- A planta é tratada com toda sua complexidade, não sendo feito considerações a respeito das variações paramétricas ou da necessidade de linearizações e pontos de operação.
- Não apresenta problemas de instabilidade nas fases iniciais de operação, fase em que a rede ainda não se encontra com os pesos bem ajustados.
- O controlador neural apresenta uma boa resposta em todas as situações: condições nominais, variação paramétrica e perturbações.
- Carga computacional relativamente simples: não precisa do modelo da planta ou da inversa do Jacobiano da planta.
- Existe a possibilidade de acoplar a rede a uma estrutura de controle já atuando sobre o processo (controle PD).
- Emprego de algoritmos simples de treinamento.
- Obtenção de estruturas de rede mais compactas e simples.

Um aspecto muito interessante encontrado nas estruturas de controle neural implementadas nesse trabalho, consiste no fato de que a atuação da rede não se encontra

baseada no conhecimento da dinâmica do manipulador, mas sim no desempenho dos controladores convencionais. O fato de se empregar os sinais de controle para o treinamento da rede, implica em que o seu funcionamento vai depender do desempenho dos controladores convencionais. Ou seja, em situações em que o controlador convencional não consegue controlar, a rede neural supre as deficiência da estrutura convencional, como são os casos de variação paramétrica, perturbação e ruídos. Nesses casos, a rede não tem a informação de que a planta está sujeita a condições não modeladas e não previstas, mas sim de que a estrutura de controle convencional não está sendo capaz de atender as especificações, sendo necessário um esforço de controle adicional, por parte da rede neural. O controlador neural apresenta, portanto, uma estrutura robusta a dinâmicas não modeladas e não previstas em condições normais de operação do manipulador.

Entretanto, as estruturas de controle neural apresentaram ações de controle mais oscilatórias, em comparação com as estratégias convencionais de controle de posição e controle híbrido força/posição. Outra desvantagem, consiste na necessidade de existir um controlador principal (controlador PD). Nas estratégias de controle neural analisadas, a rede não substitui o controlador como em outras propostas, mas somente compensa a sua atuação em situações em que o mesmo não consegue agir adequadamente. Além disso, ainda não se encontram estabelecidas provas e critérios formais de estabilidade para o sistema robô/controlador + rede neural, que permitissem uma análise mais consistente da estrutura de controle durante a fase de projeto do sistema de controle.

7.2 Perspectivas

O projeto do controlador neural envolve a escolha de um conjunto de parâmetros, como o tipo de topologia da rede neural, a arquitetura de controle a ser empregada, o número de camadas e neurônios da rede, etc., estando baseado, muitas vezes, em critérios heurísticos de ajustes e especificação de parâmetros. Dessa forma, não é possível obter critérios formais de análise de estabilidade e projeto dos controladores. Um primeiro avanço no presente trabalho, seria procurar estabelecer, se possível, uma análise mais formal do conjunto controlador clássico + rede neural + robô manipulador, objetivando dessa maneira, propor critérios que assegurem a estabilidade do sistema, auxiliando a fase de projeto do controlador[1].

[1]Na seção 4.3.2, são indicados dois caminhos, ainda a serem pesquisados mais profundamente, mas que apontam para a solução do problema da análise de estabilidade de redes neurais em controle:

O critério de treinamento da rede encontra-se unicamente baseado nos erros de posicionamento (considerando que a saída dos controladores PD constituem dos erros de posição e velocidade multiplicados por ganhos). Com esse tipo de sinal de treinamento, a rede procura ajustar sua estrutura de forma a minimizar esses erros. Porém, observou-se que o sinal de controle do sistema controlador PD + RNA apresenta-se mais oscilatório que as estruturas convencionais. Isso se deve ao fato de não estar incluído no projeto ou na estrutura da rede, um mecanismo que procure minimizar um critério baseado que leve em consideração a ação de controle. Sendo assim, procurar treinar a rede tendo como função a ser minimizada não somente o erro de posição, mas também a ação de controle, e por ventura outro critério aqui não mencionado, constitui-se de um aprimoramento substancial dos resultados obtidos nesse trabalho.

A implementação de controladores neurais para robôs baseados em redes do tipo CMAC ou RBF começa a despontar dentro da área da robótica, com resultados promissores [KV96]. Procurar comparar o desempenho dessas estruturas de rede com as tradicionais redes MLP, com o objetivo de comparar as capacidades de aprendizagem local (CMAC, RBF) e global (MLP) em sistemas rápidos e complexos como os manipuladores robóticos, também constitui de um tópico a ser desenvolvido.

Outro aspecto a ser explorado na área de controle de força/posição corresponde ao emprego de redes neurais na modelagem e identificação da dinâmica do meio em contato com o manipulador. Alguns trabalhos recentes encontram-se desenvolvidos, como [FSTM92], [VGBT93], [MN94], [GHW97] e [JH98]. Entretanto, existe ainda muitas considerações e trabalhos a serem feitos, procurando estabelecer resultados mais definitivos na área de controle de força.

E por último, a implementação dos algoritmos de controle neural de força/posição propostos no robô SCARA do Laboratório de Robótica, constitui uma aplicação direta e imediata dos resultados obtidos nessa dissertação.

análise da estrutura de controle considerando a rede neural como uma perturbação, baseada nas propriedades dos controladores clássicos em robótica, ou como uma matriz de realimentação de estados com ganhos não lineares, com as entradas da rede definidas de forma a permitir alcançar todo o espaço de estados do manipulador.

Apêndice A

Dados do Manipulador SCARA

Neste apêndice apresentam-se os valores dos parâmetros do manipulador SCARA do Laboratório de Robótica da Universidade Federal de Santa Catarina, que atualmente se encontra nas dependências do Laboratório de Automação Industrial do Departamento de Automação e Sistemas - figura A.1.

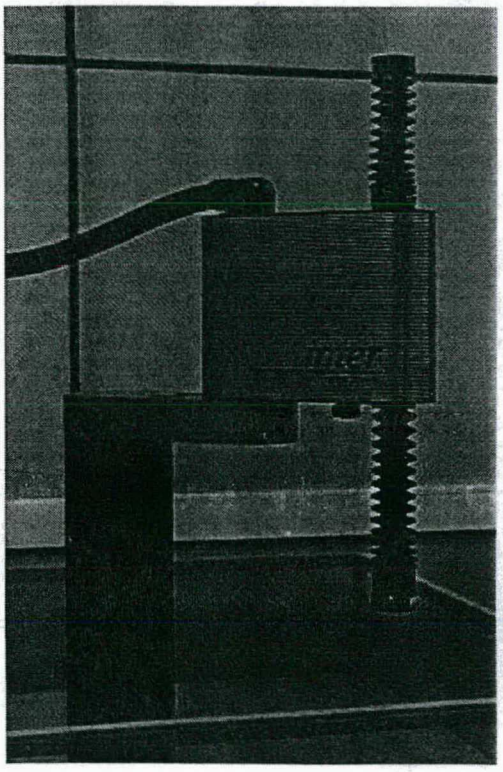

Figura A.1: Manipulador SCARA do Laboratório de Robótica

Os valores dos parâmetros do manipulador e outros detalhes técnicos podem ser encontrados em [GWG98]. Os dados obtidos, estão baseados nas informações provenientes de catálogos, ou através de medição ou estimação. Esse manipulador utilizada

motores elétricos para o acionamento das juntas.

Os valores máximos e mínimos de posição, velocidade, aceleração e torque nas juntas são apresentados na tabela A.1.

Parâmetro	Junta 1	Junta2	Junta 3	Junta 4
Posição Mínima (rad)	- 2.25	-1.9	0.14 (m)	-2.5
Posição Máxima (rad)	2.25	1.9	0.4 (m)	2.5
Velocidade (rad/s)	± 3	± 3	± 0.88 (m/s)	± 20
Aceleração (rad/s^2)	± 80	± 100	± 3 (m/s^2)	± 500
Torque Mínimo (Nm)	± 333	± 157	± 6.975	± 16.74

Tabela A.1: Valores Limites

Os valores dos parâmetros e coeficientes do manipulador encontram-se listados na tabela A.2.

Parâmetro	Valor
Massa do elo 1 (m_1)	11.4 Kg
Massa do elo 2 (m_2)	19.5 Kg
Massa do elo 3 (m_3)	2 Kg
Massa do elo 4 (m_4)	1.5 Kg
Inércia do elo 1 (I_1)	0.23 Kg m^2
Inércia do elo 2 (I_2)	0.16 Kg m^2
Inércia do elo 3 (I_3)	0.1 Kg m^2
Inércia do elo 4 (I_4)	0.1 Kg m^2
Comprimento do elo 1 (l_1)	0.25 m
Comprimento do elo 2 (l_2)	0.25 m
Centro de massa do elo 1 (l_{c1})	0.118 m
Centro de massa do elo 2 (l_{c2})	0.116 m
Origem do sistema de coordenadas da terceira junta d_0	0.678 m

Tabela A.2: Parâmetros do Manipulador

Apêndice B

Algoritmos de Treinamento para Redes MLP

A seguir estão descritos os algoritmos de treinamento para redes do tipo MLP - *Multilayer Perceptron.* Maiores detalhes podem ser vistos em [KV93], [FS91], [CBL98], [Rie94], [PFA+94] ou [Fah88].

B.1 Algoritmo Backpropagation

O algoritmo *backpropagation* certamente é o algoritmo de treinamento mais conhecido dentro da comunidade de redes neurais [RM86]. O algoritmo *backpropagation* também é conhecido como *regra delta generalizada*, por derivar de um método de treinamento para redes diretas, sem ciclos e de uma camada, proposto por Widrow e Hoff [CBL98], conhecida como *regra delta* [FS91]. Ele pode ser visto com um algoritmo de gradiente descendente, aplicado ao problema de otimização não-linear [HSZG92].

Duas fases compõe esse algoritmo. Durante a primeira fase, denominada de fase *forward*, são apresentados os conjuntos de vetores de entrada à rede, e calculado o vetor de saída. Na fase seguinte, também chamada de fase *backward*, o erro entre a saída obtida na fase anterior e o valor desejado para a saída da rede é retropropagado em direção à camada de entrada, para o cálculo da atualização dos pesos.

O algoritmo *backpropagation* procura modificar os pesos da rede de forma a minimizar uma função de erro global, dada pela equação B.1.

$$E = \frac{1}{2}\sum_{p}\sum_{i=1}^{n}(d_i^T - y_i^T)^2 \tag{B.1}$$

onde E é a medida do erro total da rede, p é o número de padrões apresentados à rede, n é o número de neurônios de saída, d_i é a i-ésima saída desejada e y_i é a i-ésima saída

gerada pela rede. Sem perda de generalidade, pode-se afirmar que a minimização da função de erro para cada padrão individual de treinamento, levará a minimização da equação B.1. Dessa forma, o erro pode ser definido como sendo

$$E_p = \frac{1}{2}\sum_{k=1}^{n}(d_k^T - y_k^T)^2 \tag{B.2}$$

A função B.2 , define uma superfície de erro em função dos valores dos pesos da rede. O objetivo do algoritmo, torna-se, portanto, obter os valores de peso que apresentam o menor valor para a função B.1 ou B.2, ou seja, que o algoritmo atinja o mínimo global na superfície de erro.

Dessa forma, e de acordo com a regra delta, a variação do peso deve ser feita proporcionalmente ao negativo do gradiente do erro em relação aos pesos da rede, ou seja,

$$\Delta w_{ij} = -\eta\frac{\partial E}{\partial w_{ij}} \tag{B.3}$$

onde w_{ij} é o valor da conexão, ou peso, entre a entrada o neurônio j e a entrada i, e o termo η é a taxa de aprendizagem. O valor da taxa de aprendizagem apresenta um compromisso entre a velocidade com que o erro diminui na superfície de erro, e a capacidade do algoritmo permanecer nos mínimos do gradiente do erro.

Para o desenvolvimento das equações que descrevem o algoritmo, considera-se que um neurônio tem como função de ativação a equação B.4

$$y_j^p = \Phi_i(x_j^p) \tag{B.4}$$

onde

$$x_j^p = \sum_{i=1}^{n} u_i^p w_{ij} \tag{B.5}$$

é a função da entrada do neurônio, n representa o número de conexões de entrada no neurônio j, e u_i^p é o valor da entrada i do neurônio j em relação ao padrão p do conjunto de treinamento, composto por $u_i = \{u_i^1, u_i^2, \ldots, u_i^p\}$. Como será visto adiante, observa-se que a função de ativação do neurônio Φ deve ser diferenciável.

O próximo passo é calcular o gradiente do erro. Utilizando-se a regra da cadeia, tem-se que

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial x_j}\frac{\partial x_j}{\partial w_{ij}} \tag{B.6}$$

com $x_j = \sum_{i=1}^{n} u_i w_{ji}$. A segunda derivada da expressão B.6 pode ser calculada através da relação

$$\frac{\partial x_j}{\partial w_{ji}} = \frac{\partial \sum_{l=1}^{n} u_l w_{jl}}{\partial w_{ji}} = u_i \tag{B.7}$$

O primeiro termo depois da igualdade na equação B.6, pode ser expandida para

$$\delta_i = \frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial x_j} \tag{B.8}$$

onde δ_i é introduzido com a finalidade de simplificar a notação. A derivada da função de ativação y_i do neurônio em relação à função de entrada x_i do mesmo pode ser calculada por

$$\frac{\partial y_j}{\partial x_j} = \frac{\partial \Phi(x_j)}{\partial x_j} = \Phi^{'}(x_j) \tag{B.9}$$

Por sua vez, o cálculo da primeira derivada da equação B.8 vai depender da localização do neurônio j na estrutura da rede. Se o mesmo estiver na última camada, o valor da derivada pode ser determinada por

$$\frac{\partial E}{\partial y_j} = \frac{\partial (\frac{1}{2}\sum_{i=1}^{k}(d_i - y_j)^2}{dy_i} = (d_j - y_j) \tag{B.10}$$

e para o caso em que o neurônio pertença a camada intermediária ou escondida, o valor da derivada é obtida através da seguinte expressão:

$$\begin{aligned} \frac{\partial E}{\partial y_j} &= \sum_{l=1}^{M} \frac{\partial E}{\partial x_l}\frac{\partial x_l}{\partial y_j} \\ &= \sum_{l=1}^{M} \frac{\partial E}{\partial x_l}\frac{\partial (\sum_{i=1}^{n} w_{il} y_i)}{dy_j} \\ &= \sum_{l=1}^{M} \frac{\partial E}{\partial x_l} w_{jl} \end{aligned} \tag{B.11}$$

com

$$\sum_{l=1}^{M} \frac{\partial E}{\partial x_l} w_{jl} = \sum_{l=1}^{M} \delta_l w_{jl} \tag{B.12}$$

Substituindo-se os dois termos na equação B.8, pode-se chegar a

$$\delta_j = (d_j - y_j)\Phi^{'}(x_j) \tag{B.13}$$

para a camada de saída, e

$$\delta_j = \Phi'(x_j) \sum_l \delta_l w_{lj} \tag{B.14}$$

para as camadas intermediárias. Dessa forma, é possível generalizar a equação de ajustes de pesos como sendo

$$w_{ji}(t+1) = w_{ji}(t) + \eta \delta_j(t) u_i(t). \tag{B.15}$$

O algoritmo *backpropagation* apresenta duas características que prejudicam o treinamento das redes MLP. A primeira delas, consiste na baixa velocidade de convergência quando próximo de um mínimo da função de erro, devido ao fato da atualização dos pesos ser proporcional ao valor do gradiente de erro. A outra característica, ocorre quando o algoritmo atinge um mínimo local e não mais consegue sair da região de atração desse mínimo, não sendo atingido, portanto, o mínimo global da função de erro. Esse problema é conhecido como *problema dos mínimos locais* [FS91], [KV93] e [CBL98].

Uma variante do algoritmo *backpropagation* consiste em incluir um segundo termo na equação B.3, chamado de *momentum*. Esse termo busca suprir alguma das deficiências do algoritmo *backpropagation*. A equação B.16 apresenta a inclusão do termo de momentum na equação padrão do *backpropagation*

$$\Delta w_{ij}(t) = -\eta \frac{\partial E}{\partial w_{ij}}(t) + \mu \Delta w_{ij}(t-1) \tag{B.16}$$

onde μ é o fator de *momentum*.

De forma a buscar resolver algumas das deficiências do *backpropagation*, diversos algoritmos foram propostos, baseados ou não no algoritmo nos métodos do *gradiente descendente*: algoritmos baseados na regra *Delta Bar*; método de *Newton*; *Quickpropagation* [Fah88]; *Resilient Backpropagation* ou *RPROP* [Rie94]; *SuperSAB*; *ABPropagation* [PFA+94]; *Fast Elleane 5,6,7* [KV93], dentre outros.

B.2 Algoritmo ABPropagation

A regra de atualização dos pesos empregada pelo algoritmo *ABPropagation* também está baseada no algoritmo de gradiente descendente, ou seja, assim como no algoritmo *backpropogation*, os pesos são atualizados na direção negativa do gradiente de erro. Entretanto, a diferença entre esses dois algoritmos, reside no fato que para o algoritmo

ABPropagation, a taxa de aprendizagem é uma função específica do erro e do gradiente de erro, escolhidos de forma a acelerar a convergência do treinamento. O termo *ABPropagation* vem de *Adaptive Backpropagation*.

O algoritmo trabalha no modo *batch*, ou seja, os pesos da rede são atualizados após serem contabilizados as parcelas de erro em relação a cada um dos exemplos do conjunto de treinamento.

Os pesos da rede são atualizados de acordo com a seguinte equação:

$$w_{i+1} = w_i - \Gamma(E)\frac{\frac{\partial E}{\partial w}}{\left|\left|\frac{\partial E}{\partial w}\right|\right|^2} \tag{B.17}$$

onde $\Gamma(E)$ corresponde a uma função que depende do erro. Existem várias possibilidades para a definição da função $\Gamma(E)$, dentre elas, podem ser utilizadas:

$$\Gamma(E) = \begin{cases} \eta \\ \eta E \\ \eta \tanh\left(\frac{E}{E_0}\right) \end{cases} \tag{B.18}$$

onde η e E_0 são constantes positivas, representando o passa de atualização dos pesos e o fator de normalização, respectivamente.

Uma das vantagens apresentadas pelo algoritmo *ABPropagation*, consiste na dependência da taxa de aprendizagem em relação ao valor instantâneo da função de erro E. Essa característica confere ao algoritmo uma aceleração da sua convergência. Além disso, o algoritmo não introduz parâmetros extras na regra de aprendizagem, como os algoritmos *quickpropagation* e *RPROP*

B.3 Algoritmo Quickpropagation

Proposto por S. Fahlman pela primeira vez em [Fah88], o algoritmo *quickpropagation* supõe que a curva de erro em relação aos pesos da rede, para cada um dos pesos, pode ser aproximada por uma parábola, com concavidade voltada para cima. Também assume-se que a alteração na inclinação dessa curva de erro depende somente do peso em questão, não sendo influenciada pela alteração dos demais pesos da rede.

A equação de modificação dos pesos é dada por

$$\Delta w(t) = \frac{S(t)}{S(t-1) - S(t)}\Delta w(t-1) \tag{B.19}$$

onde $S(t)$ e $S(t-1)$ são os valores atual e anterior de $\partial E/\partial w$. Entretanto, Fahlman propôs um algoritmo mais completo, para evitar problemas de oscilação e para limitar o crescimento nos valores dos pesos da rede. O algoritmo encontra-se descrito na seqüência.:

FOR cada peso w_i THEN
IF $\Delta w_{i-1} > 0$ THEN
IF $\frac{\partial E}{\partial w_i} > 0$ THEN
$\Delta w_i := \eta \frac{\partial E}{\partial w_i}$
IF $\frac{\partial E}{\partial w_i} > \frac{\mu}{1+\mu} \frac{\partial E}{\partial w_{i-1}}$ THEN
$\Delta w_i := \Delta w_i + \mu \Delta w_{i-1}$
ELSE
$\Delta w_i := \Delta w_i + \frac{\frac{\partial E}{\partial w_i} \Delta w_{i-1}}{\frac{\partial E}{\partial w_{i-1}} - \frac{\partial E}{\partial w_i}}$
ELSEIF $\Delta w_{i-1} < 0$ THEN
IF $\frac{\partial E}{\partial w_i} < 0$ THEN
$\Delta w_i := \eta \frac{\partial E}{\partial w_i}$
IF $\frac{\partial E}{\partial w_i} < \frac{\mu}{1+\mu} \frac{\partial E}{\partial w_{i-1}}$ THEN
$\Delta w_i := \Delta w_i + \mu \Delta w_{i-1}$
ELSE
$\Delta w_i := \Delta w_i + \frac{\frac{\partial E}{\partial w_i} \Delta w_{i-1}}{\frac{\partial E}{\partial w_{i-1}} - \frac{\partial E}{\partial w_i}}$
ELSE
$\Delta w_i := \eta \frac{\partial E}{\partial w_i}$
$w_i := w_i + \Delta w_i$
IF $w_i > máximo\_peso$ THEN
emphreiniciar o treinamento
END

O termo μ é fator que limite o crescimento dos pesos. Pode-se demonstrar que a regra de adaptação proposta por Falhman é equivalente a aplicação local do método de Newton, o qual pode ser derivado do termo de primeira ordem da expansão em série de Taylor da função de erro [Rie94].

B.4 Algoritmo RPROP

O termo RPROP provém do inglês *Resilent backpropagation* e é um esquema de treinamento adaptativo local, empregado para treinamento de redes do tipo MLP. O princípio básico do algoritmo é eliminar a influência da amplitude do sinal da derivada na adaptação dos pesos. Nesse caso somente o valor do sinal da derivada é considerada. O treinamento é feito no modo *batch*.

$\forall ij : \Delta_{ij}(t) = \Delta_0$

$\forall ij : \frac{\partial E}{\partial w_{ij}}(t-1) = 0$
Repita
Calcular o Gradiente $\frac{\partial E}{\partial w}(t)$
Para todos os pesos e *bias*
IF $\frac{\partial E}{\partial w_{ij}}(t-1)\ \frac{\partial E}{\partial w}(t)\ > 0$ THEN
$\Delta_{ij}(t) = \text{mínimo}(\Delta_{ij}(t-1)\eta^{+}, \Delta_{max})$
$\Delta w_{ij}(t) = -\text{sinal}(\frac{\partial E}{\partial w_{ij}}\ \Delta_{ij}(t)$
$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$
$\frac{\partial E}{\partial w_{ij}}(t-1) = \frac{\partial E}{\partial w_{ij}}(t1)$
ELSEIF $\frac{\partial E}{\partial w_{ij}}(t-1)\ \frac{\partial E}{\partial w}(t)\ < 0$ THEN
$\Delta_{ij}(t) = \text{máximo}(\Delta_{ij}(t-1)\eta^{-}, \Delta_{min})$
$w_{ij}(t+1) = w_{ij}(t) - \Delta w_{ij}(t-1)$
$\frac{\partial E}{\partial w_{ij}}(t-1) = 0$
ELSE
$\Delta w_{ij} = -\text{sinal}(\frac{\partial E}{\partial w_{ij}}\ \Delta_{ij}(t)$
$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$
$\frac{\partial E}{\partial w_{ij}}(t-1) = \frac{\partial E}{\partial w_{ij}}(t)$
Até convergir

onde Δ_{ij} corresponde ao passo dado pelo algoritmo em direção ao mínimo da função, geralmente tendo valor inicial igual a 0.1 (Δ_0). As taxas de aprendizado são dadas por η^{+}, geralmente com valor na ordem de 1, e η^{-}, com valor entre 0 e 1. A grande vantagem do *RPROP* reside no fato de estar baseado somente no sinal do gradiente da função de erro para atualização dos pesos, evitando o aprendizado muito lento quando o erro se torna pequeno.

Apêndice C

Módulos do *XOberon*

O sistema de controle do manipulador SCARA Inter do Laboratório de Robótica foi implementado utilizado o ambiente de programação XOberon. Os módulos do programa utilizam a linguagem de programação *oberon* [Mos93] e [Rei91]. Ela consiste de uma linguagem concorrente e orientada à objetos.

Os módulos principais utilizados para implementar o controlador neural correspondem aos módulos *Main3.mod* e *StateCtrl.Mod*. Maiores detalhes quanto ao sistema de controle e supervisão do robô SCARA Inter podem ser encontrados em [GWG98]. Os dois módulos já se encontravam implementados, sofrendo alterações de forma a possibilitar a inclusão das rotinas para o controlador neural.

C.1 Módulo *StateCtrl.mod*

Esse módulo implementa a ação de controle PD. Ele foi alterado de forma a incluir novos procedimentos, onde encontram-se definidos os algoritmos de treinamento e a nova ação de controle neural. Os procedimentos mais relevantes são apresentados na seqüência.

```
StateCtrl.Mod
(** mecos Robotics AG / Robot Install GmbH *)
(** Autor: R.Hueppi, E.Nielsen; Date: 960914 *)
(** Version 960914 *)

(** Changed by Sandro Battistella/DAS**)
(** Neuro-Controler **)
(** 09.03.99 **)
```

```
IMPORT
  ⋮
CONST
Version = 990309;
(* número de entradas e neurônios em cada camada *)
ni = 12; nh1 = 8; nh2 = 5; no = 4; maxneuron = 12;

TYPE
(* Tipo VectorL é uma camada da rede *)
VectorL = ARRAY maxneuron OF LONGREAL;
Layer = RECORD
ninlayer, noutlayer : INTEGER;
weight : ARRAY maxneuron,maxneuron OF LONGREAL;
bias: VectorL;
out: VectorL;
norm: LONGREAL;
END;
  ⋮
VAR
version- : LONGINT;
V1 : Layer; V2 : Layer;  W : Layer;
seedp, seedm: LONGINT;
norm* : LONGREAL;
learningrate*, controlepd*, lr*: LONGREAL;
perturbacao*,erro0,erro1,erro2,erro3: LONGREAL;
  ⋮
PROCEDURE (c : StateCtrl) Algo*(istpos, sollpos, istspeed,
   sollspeed : REAL) : REAL;
(* Algoritmo que implementa o controlador PID *)
VAR
posdiff, poskorr,deltaspeed, forceout : REAL; p : StateCtrlPara;
BEGIN
p:= c.stateCtrlPara;
```

```
sollspeed:= RF.RampFilter(sollspeed, c.rs);
sollpos:= RF.RampFilter(sollpos, c.rp);
poskorr:=(sollspeed + istspeed)*p.systemdelayDiv2;
posdiff:= sollpos - istpos - poskorr;
deltaspeed:= sollspeed-istspeed;

(* I-Part *)
p.integ := p.integ+sollpos-istpos;
IF ((p.integ>0)&(p.integ>p.iLimit)) THEN
p.integ:= p.iLimit
ELSIF ((p.integ<0)&(p.integ<-p.iLimit)) THEN
p.integ:= -p.iLimit
END;

forceout:= (posdiff*p.kPos+deltaspeed+p.i*p.integ)*p.ineDivclock;
IF c.bsFilter#NIL THEN forceout:= c.bsFilter.Algo(forceout) END;
RETURN forceout
END Algo;

:

PROCEDURE InitLayer (seedu: LONGINT; VAR V: Layer);
(* Inicialização randômica dos pesos da rede *)
VAR
k,p : INTEGER;
u, uant : LONGINT;
ro : LONGREAL;
BEGIN
ro := 0.01; u := 0;
uant := seedu; norm := 0.0;
FOR k := 0 TO V.ninlayer-1 DO
FOR p := 0 TO V.noutlayer-1 DO
u := (seedm*uant) MOD seedp;
V.weight[k,p] := 2*ro*(u-1)/(seedp-2)-ro;
uant := u; norm := norm + V.weight[k,p];
END;
u := (seedm*uant) MOD seedp;
V.bias[k] := 2*ro*(u-1)/(seedp-2)-ro;
```

```
uant := u;
END;
norm := norm/(V.ninlayer*V.noutlayer);
END InitLayer;

PROCEDURE OutNet(VAR V: Layer; antoutnet: VectorL);
(* Saída da Rede *)
VAR
nin,nout,k,p : INTEGER;
newout : VectorL;
aux : LONGREAL;
aux2: REAL;
BEGIN
aux := 0.0;
FOR k := 0 TO V.noutlayer-1 DO
aux := 0.0;
FOR p := 0 TO V.ninlayer-1 DO
aux := aux + V.weight[k,p]*antoutnet[p];
END;
aux2 := 2*SHORT(aux + V.bias[k]);
V.out[k] := LONG((M.exp(aux2)-1.0)/(M.exp(aux2)+1.0));
END;
END OutNet;

PROCEDURE DeltatanOut (VAR delta: VectorL; nout: INTEGER; yh: VectorL;
erro: GD.Vector4);
(* Calcula o delta do algoritmo backpropagation para a camada de saida. *)
VAR
k :  INTEGER;
BEGIN
FOR k := 0 TO nout-1 DO
delta[k] := (1.0-yh[k]*yh[k])*erro[k];
END;
END DeltatanOut;

PROCEDURE Deltatan (VAR delta: VectorL; Vatual, Vpost: Layer;
deltapost: VectorL);
```

```
(* Calcula o delta do algoritmo backpropagation para as demais camadas
   da rede. *)
VAR
aux : LONGREAL;
k,p : INTEGER;
BEGIN
FOR k := 0 TO Vpost.ninlayer-1 DO
aux := 0.0;
FOR p := 0 TO Vpost.noutlayer-1 DO
aux := aux + Vpost.weight[p,k]*deltapost[p];
END;
delta[k] := aux*(1.0-Vatual.out[k]*Vatual.out[k]);
END;
END Deltatan;

PROCEDURE Learnbp(VAR V: Layer; in: VectorL; delta: VectorL; lr: LONGREAL);
(* Atualiza os pesos da rede *)
VAR
k,p : INTEGER;
BEGIN
FOR k := 0 TO V.noutlayer-1 DO
FOR p := 0 TO V.ninlayer-1 DO
V.weight[k,p] := V.weight[k,p] + learningrate*delta[k]*in[p];
IF V.weight[k,p] > maximopeso THEN
NewWeight(sementepeso,V.weight[k,p]);
INC(cont1);
END;
END;
V.bias[k] := V.bias[k] + learningrate*delta[k];
IF V.bias[k] > maximopeso THEN
NewWeight(sementepeso,V.bias[k]);
INC(cont1);
END;
END;
END Learnbp;

PROCEDURE NeuroAlgo*(acaopd,pos, vel, acc: GD.Vector4; VAR torqnet: GD.Vector4);
```