

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

**Identificação e linearização por redes neurais de
um manipulador robótico com acoplamento flexível**

Autor : Renato Luiz Machado

Orientador: Eurípedes Guilherme O. Nóbrega

23/00

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA
TESE DEFENDIDA POR RENATO LUIZ
MACHADO E APROVADA PELA
COMISSÃO JULGADORA EM 28.02.2000

Rafael Lúcio
ORIENTADOR

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE



**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
DEPARTAMENTO DE MECÂNICA COMPUTACIONAL**

Identificação e linearização por redes neurais de um manipulador robótico com acoplamento flexível

Autor : **Renato Luiz Machado**

Orientador: **Eurípedes Guilherme O. Nóbrega**

Curso: Engenharia Mecânica.

Área de concentração: Mecânica dos sólidos e projeto mecânico

Dissertação de mestrado apresentada à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para obtenção do título de Mestre em Engenharia Mecânica.

Campinas, 2000

S.P. - Brasil

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

65880008

CHAMADA:
D. UNICAMP
M 18 L
V. Es.
TOMBO BC/ 41452
PROC. 278/00
C D
PREÇO R\$ 11,00
DATA 11-07-00
N.º CPD

CM-00143095-3

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

M18i

Machado, Renato Luiz

Identificação e linearização por redes neurais de um manipulador robótico com acoplamento flexível / Renato Luiz Machado.--Campinas, SP: [s.n.], 2000.

Orientador: Eurípedes Guilherme Oliveira Nóbrega.
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Redes neurais (Computação). 2. Sistemas não lineares. 3. Identificação de sistemas. 4. Robótica. I. Nóbrega, Eurípedes Guilherme Oliveira. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

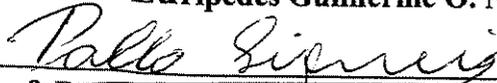
**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
DEPARTAMENTO DE MECÂNICA COMPUTACIONAL**

DISSERTAÇÃO DE MESTRADO

**Identificação e linearização por redes neurais de
um manipulador robótico com acoplamento flexível**

Autor : **Renato Luiz Machado**

Orientador: **Eurípedes Guilherme O. Nóbrega**



Prof. Dr., Pablo Siqueira Meirelles, Presidente
Universidade Estadual de Campinas



Prof. Dr. Carlos Alberto dos Santos Passos
Centro de Tecnologia para Informática



Prof. Dr. Robson Pederiva
Universidade Estadual de Campinas

Campinas, 28 de fevereiro de 2000

Agradecimentos

Este trabalho não poderia ser realizado sem a ajuda de diversas pessoas às quais presto minha homenagem:

Ao meu orientador Eurípedes G. O. Nóbrega, por ter muitas vezes me mostrado os caminhos a serem tomados.

A Pablo S. Meirelles, pela disposição de me auxiliar na etapa final deste trabalho.

A todos professores e colegas do Departamento de Mecânica Computacional, em especial Aida Fadel, André Bezerra e Belisário Huallpa pelas instruções adicionais e detalhes do trabalho.

A minha família e amigos pela ajuda, compreensão e incentivo emocional.

Ao apoio financeiro prestado pela CAPES durante todo o período de desenvolvimento deste trabalho.

“A beleza da vida está na incerteza de se descobrir o inesperado.”

Resumo

MACHADO, Renato Luiz, *Identificação e linearização por redes neurais de um manipulador robótico com acoplamento flexível.*, Campinas,: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2000. 103 p. Dissertação (Mestrado)

O objetivo deste trabalho é descrever o projeto de linearização por realimentação de manipuladores fazendo uso de redes neurais em uma configuração baseada em técnicas de controle adaptativo. O comportamento do robô projetado será estudado através de um sistema com junta flexível. O manipulador adotado é composto de três GDLs (graus de liberdade), sendo dois GDLs em corpos rígidos com um movimento de rotação e outro de translação, e uma junta flexível representada por uma correia dentada entre os dois eixos rotativos, introduzindo assim o terceiro GDL. Apresenta-se inicialmente o método de linearização dinâmica por realimentação (em inglês: *feedback linearization*). O enfoque proposto é a utilização de um mapeamento por redes neurais artificiais de múltiplas camadas para substituir os blocos de linearização e da planta. Este estudo abrange a possibilidade de se trabalhar com parâmetros desconhecidos, fazendo deste, um método que se adapta ao sistema.

Palavras Chave

Redes neurais, linearização por realimentação, identificação e robótica.

Abstract

MACHADO, Renato Luiz, *Manipulator with flexible joint identification and neural networks linearization*, Campinas,: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2000. 103 p. Dissertação (Mestrado)

The goal of this work is to present a plan of manipulators feedback linearization using neural networks based on adaptive control techniques. The behavior of designed robot will be studied through a system with flexible joint. This manipulator is composed by three DOF (degrees of freedom), with two DOF in rigid segments of rotation motion and translation motion, and one flexible joint moved for a “toothed belt” or “timing belt” between two rotary shaft, forming the third DOF. Initially it will be presented the dynamic feedback linearization method. Then artificial neural networks of multiple layers will be used in order to replace the blocks of linearization and plant. This analysis include the possibility to work with unknown parameters, that can be considered as an adaptive method to the system.

Key Words

Neural networks, feedback linearization, identification and robotic

Índice

1. Introdução.....	1
1.1. Identificação e linearização de um sistema robótico com junta flexível.....	1
1.2. Objetivo do trabalho.....	2
1.3. Controle adaptativo por modelo de referência.....	2
1.4. Organização do trabalho.....	3
2. Revisão da literatura.....	5
2.1. Introdução à robótica.....	5
2.2. O neurônio como um conceito biológico e um mecanismo computacional.....	7
2.3. Redes neurais artificiais.....	7
2.3.1. Modelo de um neurônio (<i>perceptron</i>).....	10
2.3.2. Tipos de função de ativação.....	11
2.3.3. Arquitetura da rede em múltiplas camadas.....	12
2.3.4. Retro-propagação.....	14
2.3.5. Algoritmo de retro-propagação.....	15
2.4. Sistemas dinâmicos.....	18
2.4.1. Sistema dinâmico em tempo contínuo.....	20
2.4.2. Redes neurais em sistemas dinâmicos de tempo contínuo.....	21
2.4.3. Redes neurais em sistemas dinâmicos de tempo discreto.....	
22	
2.4.4. Formulação de <i>Lagrange</i> para dinâmica de manipuladores.....	23

2.5. Identificação de sistemas.....	24
2.5.1. Identificabilidade.....	24
2.5.2. Comentários sobre modelos de identificação e redes neurais.....	25
2.5.3. Identificação da dinâmica inversa de sistemas utilizando redes neurais.....	26
2.6. Linearização por realimentação.....	27
2.6.1. Introdução.....	27
2.6.2. Sistemas de linearização de simples entrada-saída (SISO).....	28
2.6.3. Sistemas de múltiplas entradas e múltiplas saídas (MIMO).....	32
3. Modelagem teórica.....	34
3.1. Cinemática de corpos rígidos.....	34
3.1.1. Sistemas de referência.....	34
3.1.2. Representação de um vetor no sistema de referência.....	35
3.1.3. Representação do sistema.....	35
3.2. Modelagem dinâmica.....	39
3.2.1. Formulação de <i>Lagrange</i>	40
3.3. Linearização por realimentação.....	42
4. Simulação.....	48
4.1. Metodologia aplicada.....	48
4.1.1. Representação do sistema.....	48
4.2. Redes neurais em identificação.....	49
4.2.1. Identificação do sistema por redes neurais.....	50
4.2.2. Aplicação.....	51
4.3. Análise da mudança de parâmetros.....	53
4.4. Linearização por realimentação em tempo contínuo.....	56
4.4.1. Simulação do sistema com excitação por ondas quadrada e senoidal.....	56
4.4.2. Simulação do sistema com resposta ao degrau.....	63
4.4.3. Simulação do sistema em malha aberta.....	67

4.5. Comparação dos polos estimados.....	69
5. Aplicação de redes neurais em linearização e controle.....	72
5.1. Estrutura de uma rede neural.....	72
5.2. Metodologia utilizando redes neurais.....	73
5.2.1. Rede neural de linearização.....	74
5.2.2. Rede neural da planta.....	76
5.3. Substituição do linearizador por uma rede neural representada no Simulink.....	80
6. Conclusões.....	86
Referências Bibliográficas.....	89
Anexo I.....	94
Anexo II.....	95
Anexo III.....	96
Anexo IV.....	99

Lista de Figuras

1.1	Diagrama de blocos de um sistema adaptativo por modelo de referência.....	2
2.1	Organização funcional de um robô.....	6
2.2	Um neurônio: (a) conceito biológico; (b) mecanismo computacional.....	7
2.3	“Perceptron”	11
2.4	Funções de ativação: (a) linear; (b) sigmoide; (c) intervalo linear.....	12
2.5	Perceptron de múltiplas camadas.....	13
2.6	Rede completamente interconectada, com uma camada intermediária.....	15
2.7	Uma rede neural vista como um sistema.....	22
2.8	Identificação de sistemas utilizando redes neurais.....	27
2.9	Identificação de sistemas pela dinâmica inversa utilizando redes neurais.....	27
2.10	Linearização de entrada-estado.....	28
2.11	Linearização por realimentação.....	30
2.12	Sistema de linearização.....	31
3.1	Sistemas de referência.....	35
3.2	Representação esquemática do sistema de 3 GDL.....	39
4.1.	Representação do modelo não linear.....	49
4.2.	Procedimento de identificação.....	50
4.3.	Modelo para identificação com redes neurais artificiais.....	50
4.4.	Identificação de sistemas por redes neurais.....	51
4.5.	Deslocamento e velocidade do corpo 1.....	52
4.6.	Ângulo e velocidade angular do corpo2.....	52
4.7.	Ângulo e velocidade angular do corpo3.....	53

4.8.	Deslocamento e velocidade linear do corpo1.....	54
4.9.	Ângulo e velocidade angular do corpo2.....	55
4.10.	Ângulo e velocidade angular do corpo3.....	55
4.11.	Diagrama de blocos da simulação com ondas quadrada e senoidal em malha fechada.....	56
4.12.	Diagrama de blocos que representa a planta.	58
4.13.	Diagrama de blocos que representa $f(x)$	58
4.14.	Diagrama de blocos que representa $g(x)*u$	58
4.15.	Diagrama de blocos que representa a o controle não linear.	59
4.16.	Posição do corpo1.....	59
4.17.	Velocidade linear do corpo1.....	60
4.18.	Ângulo do corpo2.....	60
4.19.	Velocidade angular do corpo2.....	61
4.20.	Ângulo do corpo3.....	62
4.21.	Velocidade angular do corpo3.....	62
4.22.	Diagrama de blocos da simulação com reposta ao degrau.	63
4.23.	Posição do corpo1.....	63
4.24.	Velocidade linear do corpo1.....	64
4.25.	Ângulo do corpo2.....	64
4.26.	Velocidade angular do corpo2.....	65
4.27.	Ângulo do corpo3.....	66
4.28.	Velocidade angular do corpo3.....	66
4.29.	Diagrama de blocos da simulação em malha aberta.....	67
4.30.	Posição corpo1 sem realimentação no controle.....	67
4.31.	Ângulo do corpo3 sem realimentação no controle.....	68
4.32.	Posição do corpo1 sem realimentação no controle.....	68
4.33.	Ângulo do corpo3 sem realimentação no controle.....	69
4.34.	Posição do corpo 1 com sinal de onda quadrada.....	70
4.35.	Posição do corpo 1 com resposta ao degrau.....	70

4.36.	Posição do corpo 1 com sinal de onda quadrada.....	71
4.37.	Posição do corpo 1 com resposta ao degrau.....	71
5.1.	Diagrama de blocos das redes neurais aplicadas ao sistema de linearização e representação da planta.....	74
5.2.	Sinal da primeira saída da rede de linearização.....	75
5.3.	Sinal da segunda saída da rede de linearização.....	75
5.4.	Sinal da terceira saída da rede de linearização.....	76
5.5.	Comparação das saídas desejada e da rede representando a posição do corpo 1.....	77
5.6.	Comparação das saídas desejada e da rede representando a posição angular do corpo 2.....	77
5.7.	Comparação das saídas desejada e da rede representando a posição angular do corpo 3.....	78
5.8.	Comparação das saídas desejada e da rede representando a velocidade angular do corpo 1.....	78
5.9.	Comparação das saídas desejada e da rede representando a velocidade angular do corpo 2.....	79
5.10.	Comparação das saídas desejada e da rede representando a velocidade angular do corpo 3.....	79
5.11.	Diagrama de blocos do sistema com uma rede neural de linearização.....	80
5.12.	Diagrama de blocos do subsistema da rede neural de linearização.....	81
5.13.	Aproximação da curva que representa o deslocamento do corpo 1.....	81
5.14.	Aproximação da curva que representa a velocidade do corpo 1.....	82
5.15.	Aproximação da curva que representa o deslocamento do corpo 3.....	82
5.16.	Aproximação da curva que representa a velocidade do corpo 3.....	83
5.17.	Rede neural em sinais de baixa frequência representando o deslocamento do corpo 1.....	84
5.18.	Representação dos sinais que representam o deslocamento do corpo 1.....	84

Nomenclatura

Letras Latinas

u_p – sinal de entrada da planta

y_p – sinal de saída da planta

p – número de entradas

x_p – vetor de entrada

w_p – pesos sinápticos

b – *bias*

k – neurônio k

u_k – combinação linear da saída do neurônio k

y_k – sinal de saída da rede neural

v_k – potencial de ativação ou nível de atividade interna

R – neurônio da camada intermediária

e – sinal de erro da saída da rede

n – número de iterações

y_d – sinal de saída desejado

N – número de amostras

m – massa

q – variável de estado

t – tempo

L – Lagrangeano

K – energia cinética

V – energia potencial

$\hat{x}(k)$ - estimador de $x(k)$

$\hat{y}(k)$ - estimador de $y(k)$,

u – controle não linear (lei de controle)

v – nova lei de controle
z – novas coordenadas
T – matriz de transformação de coordenadas
I – sistema de referência inercial
A – sistema de referência móvel
J – momento de inércia do motor
 k_m - rigidez da correia
g - aceleração da gravidade
 q_i - deslocamento do corpo 1
 \dot{q}_i - velocidade linear
l - posição
 \bar{v} - velocidade linear
h - altura de elevação
 x_m - deformação da mola
 I_j – momento de inércia dos corpos
T – período

Letras Gregas

φ – função de ativação
 θ - limiar de ativação
 Σ - Junção somatória
 ε - soma instantânea dos erros ao quadrado
 ε_m - média dos erros ao quadrado
 η - taxa de aprendizagem
 Δ - regra delta
 δ - gradiente
 α e β - posição angular
 $\bar{\omega}$ - velocidade angular
 τ - torque do motor;
 ξ - modelo de estado em forma matricial
 ψ - transformação das coordenadas não lineares
 ω - frequência

Abreviações

tanh – tangente hiperbólica

INITFF – valores iniciais aos pesos e bias de uma rede *feed-forward*

TRAINLM – treinamento da rede *feed-forward* utilizando o método Levenberg-Marquardt

SIMUFF – simulação da rede *feed-forward*

TANSIG – função de ativação não linear tangente hiperbólica

PURELIN – função de ativação linear

Siglas

a – ângulo

d – deslocamento

DMC – Departamento de Mecânica Computacional

LM – Levenberg-Marquardt

SQE – Soma Quadrática do Erro

vl – velocidade linear

Capítulo 1

Introdução

1.1 Identificação e linearização de um sistema robótico com junta flexível

As técnicas de tratamento de sinais em sistemas dinâmicos não lineares têm progredido consideravelmente nos últimos anos, onde as linhas de pesquisa na área de inteligência artificial têm apresentado resultados animadores para estes complexos problemas. Pode-se destacar alguns trabalhos desenvolvidos nesta área como: Narendra (1990), Irwin (1992), Noriega (1998), entre outros. Estes trabalhos demonstram a atual empregabilidade de redes neurais como diferentes soluções a problemas de sistemas não lineares. Um outro método de bastante sucesso, utilizado para problemas não lineares, é a linearização por realimentação. Aplicações em equações que descrevem a dinâmica de robôs compostos por juntas de rotação e translação, podem ser observadas nos exemplos encontrados em Spong(1989), Isidori(1989) e Slotine(1986).

Neste trabalho, o mecanismo estudado como um sistema não linear tem seu comportamento dinâmico reproduzido através de uma rede neural com múltiplas camadas, treinada com os dados obtidos através de simulação. A partir da rede assim obtida, referida como uma rede de identificação da planta a ser controlada, a etapa seguinte é a implementação da linearização por realimentação através de uma segunda rede neural, visando reduzir a complexidade das equações do sistema a ser controlado.

1.2 Objetivo do trabalho

Uma das principais propostas deste trabalho é analisar e projetar a lei de controle por realimentação para sistemas não lineares. Propõe-se mostrar a aplicação de redes neurais a um sistema robótico, através da compensação de suas não linearidades. Serão adotadas técnicas de controle adaptativo por modelo de referência fazendo uso de redes neurais, através de uma configuração de dupla malha em cascata. A rede neural de linearização deve atuar para compensar as não linearidades da planta, de modo a apresentar uma relação entrada/saída que, além de linear, deve ser compatível com o modelo de referência adotado para o sistema de malha fechada da primeira malha, possibilitando assim um controlador simples e linear na malha externa.

1.3 Controle adaptativo por modelo de referência

A utilização de redes neurais para linearização do sistema é baseada na técnica de Controle Adaptativo por Modelo de Referência, um método eficiente e bem conhecido para aplicação em plantas com parâmetros desconhecidos ou variáveis ao longo do tempo. Este método tem sido utilizado em problemas de alto nível de complexidade, cobrindo uma ampla faixa de aplicação prática como em: controle de aeronaves, mísseis, satélites, processos químicos e metalúrgicos, entre outros (Chalam, 1987). O objetivo do controle é ajustar os parâmetros do sistema para minimizar o erro entre os sinais de saída da planta e do modelo de referência.

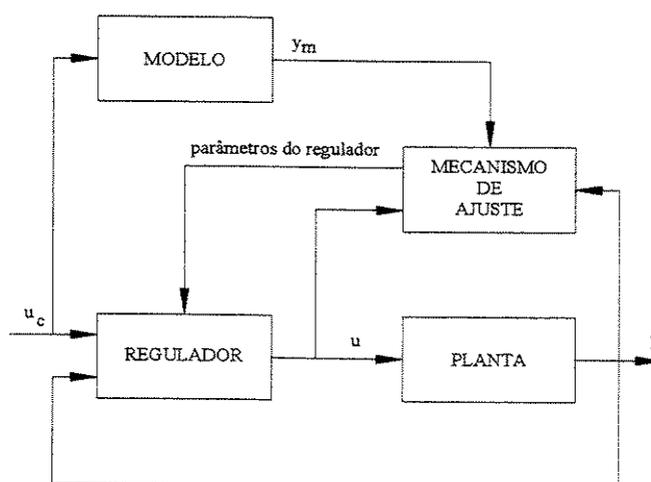


Figura 1.1 – Diagrama de blocos de um sistema adaptativo por modelo de referência

Na Figura 1.1 está um diagrama de blocos de um sistema de controle adaptativo por modelo de referência, onde a “Planta” do sistema tem como excitação um sinal adequado (“Regulador”) para se obter uma saída desejada. O “Modelo” referencia os parâmetros do sistema, e o “Mecanismo de Ajuste” direciona o sinal como uma realimentação de informações.

1.4 Organização do trabalho

Capítulo 1

Neste capítulo, apresentou-se sucintamente as técnicas utilizadas no trabalho e introduziu-se as idéias gerais de linearização, identificação neural e a utilização dos métodos baseados em controle adaptativo. Procurou-se enfatizar a aplicação dos métodos citados com uma apresentação genérica do sistema envolvido, além de mostrar a forma organizacional de apresentação da dissertação.

Capítulo 2

O principal objetivo deste capítulo é sintetizar o estudo teórico que abrange parte da linha de pesquisa na área de robótica, redes neurais, sistemas dinâmicos, identificação de sistemas e linearização por realimentação. Determina assim o direcionamento para toda a abordagem do trabalho, através da análise baseada em estudos de diversas obras e autores especialistas em cada área. Apresentam-se também neste capítulo a formulação e os algoritmos gerais para o tratamento do problema proposto.

Capítulo 3

A modelagem teórica é composta pelo estudo do comportamento da trajetória do manipulador através das variáveis cinemáticas de posição e velocidade pelo método *Newton-Euler*, e as equações de movimento para os três corpos, utilizando o método de *Lagrange*. Através das equações de movimento, parte-se para a aplicação da linearização por realimentação,

obtendo a forma matricial do modelo de estado e as equações não lineares que representam o sistema robótico estudado.

Capítulo 4

São feitas as análises referentes ao desempenho do manipulador representando nos gráficos correspondentes a identificação do sistema por redes neurais, a linearização por realimentação, e a estimação dos polos. Para isso são executadas as simulações, através da ferramenta interativa Simulink, componente do ambiente MATLAB®, comparando-se o comportamento do robô submetido à variação de alguns parâmetros e da excitação, buscando-se assim a observação e análise dos resultados obtidos.

Capítulo 5

Um dos pontos principais do trabalho é a caracterização da capacidade das redes neurais multicamadas de representar plantas com parâmetros variantes no tempo. Desse modo, pode ser feita a substituição do linearizador e da planta por redes neurais. A substituição faz uso da capacidade das rede neurais serem treinadas apenas com os sinais de entrada e saída de um dado módulo, não sendo necessário ter conhecimento dos seus parâmetros internos.

Capítulo 6

São apresentadas as conclusões da análise da utilização nos resultados obtidos através das simulações, evidenciando a possibilidade de aplicação efetiva em manipuladores experimentais. São também apresentadas as propostas de trabalhos futuros que poderiam ser estudados e complementados às análises realizadas nesta dissertação.

Capítulo 2

Revisão da literatura

2.1 Robótica

Desde suas primeiras utilizações nos anos 20 até hoje, o robô tem sido constantemente alvo de pesquisadores para auxílio em certas tarefas que necessitam da substituição do homem, seja em força, velocidade, precisão ou resistência ao ambiente hostil. A robótica compreende uma área de aplicações muito ampla, contudo o seu traço dominante continua sendo o tratamento dos problemas colocados pelo desenvolvimento estrutural e funcional de máquinas e processos. Como exemplo, o uso de manipuladores em operações repetitivas ou de precisão, ou ainda poupando risco ao homem.

A robótica é multidisciplinar por envolver conhecimentos das áreas de sistemas mecânicos, eletrônicos e informáticos, entre outros. Além do aspecto fundamental científico e tecnológico, a gama de outros interesses que a robótica desperta é muito diversificada. Apesar de todo o esforço de pesquisa, os robôs atuais ainda estão muito aquém do que seria desejado, sendo a robótica um domínio científico onde a sua difusão e expectativa junto ao leigo é muito maior do que o estado real do conhecimento na área. Isto constitui na verdade uma das motivações para o seu estudo e avanço tecnológico.

2.2 O neurônio como um conceito biológico e um mecanismo computacional

A rede neural esforça-se a imitar uma forma de processamento de informações do cérebro humano. O componente básico do cérebro humano é o neurônio. O neurônio é constituído de uma célula com extensões ramificadas chamadas “dendritos”, por receber entradas, e o “axônio”, que transporta a saída do neurônio para os dendritos dos outros neurônios (Figura 2.2). A junção entre um axônio e um dendrito é chamado “sinapse”. O neurônio é responsável por executar o cálculo do limiar, coletando sinais de suas sinapses e somando-os. Se a força de um sinal combinado excede um certo limiar, o neurônio envia seu próprio sinal, o qual é uma transformação do sinal de entrada original.

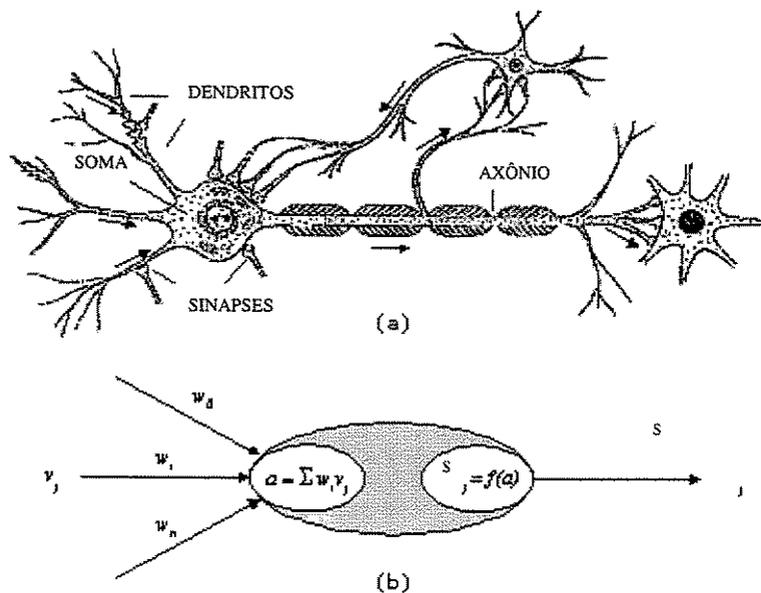


Figura 2.2 – Um neurônio: (a) conceito biológico; (b) mecanismo computacional

2.3 Redes neurais artificiais

Redes neurais artificiais podem ser definidas como uma ferramenta computacional emergente utilizada para obter resultados aproximados de problemas não lineares através de um mapeamento de entradas e saídas, com habilidade de aprendizagem e armazenamento de dados adaptados ao ambiente computacional. As redes neurais artificiais foram baseadas no cérebro humano, podendo-se fazer algumas comparações entre eles.

Redes neurais artificiais assemelham-se ao cérebro em alguns aspectos, podendo-se destacar dois deles:

- O conhecimento é adquirido pela rede através de um processo de aprendizagem;
- As conexões entre os neurônios, conhecidas como pesos sinápticos, são usadas para armazenar os dados.

Embora bastante discutidas, está claro que há algumas similaridades entre os atuais algoritmos de neurônios artificiais e os sistemas biológicos. O fato dos sistemas biológicos mostrarem capacidade de reconhecimento bem superior aos nossos algoritmos, mantém-se um incentivo de continuidade com a conexão neurobiológica.

Redes neurais com arquitetura em camadas e fluxo de informações, com ou sem realimentação, possuem grande poder de adaptação e representação não linear. A presença de realimentação introduz memória no processamento, incluindo assim a capacidade de representação de sistemas dinâmicos.

As redes neurais despontam como uma ferramenta alternativa bastante competitiva quando comparados aos métodos tradicionais, tais como aproximação de funções multivariáveis, identificação e controle de sistemas dinâmicos. Sua versatilidade é devido principalmente à sua capacidade de representação de comportamentos não lineares, sendo potencialmente capaz de produzir um desempenho equivalente ou superior ao de qualquer outro método.

Para explorar esta capacidade de representação, é necessário desenvolver algoritmos de treinamento adequados que permitam tratar eficientemente o poder dessas estruturas. Deve-se tomar proveito da disponibilidade atual de recursos computacionais de alto desempenho, constituindo um tema de grande interesse no estudo sobre abordagens conexionistas, o desenvolvimento de novas técnicas de treinamento.

Embora a representação de sistemas por redes neurais exista em diferentes formas, há certos parâmetros comuns entre todos os paradigmas, por exemplo:

- A maioria das redes neurais utilizam o parâmetro peso a ser ajustado para gerar a aprendizagem;
- A maioria das redes neurais tem diversas unidades interconectadas;
- A maioria das redes neurais usa uma regra de aprendizagem, a qual dita o fluxo de informações na rede;
- As redes neurais são baseadas na minimização de uma função de custo, função de energia ou funções de combinação complexa.

As redes neurais proporcionam uma solução aproximada para uma grande variedade de problemas científicos e de engenharia, considerados tradicionalmente difíceis. Embora não exista uma definição exata, os autores em geral enfatizam um ou outro aspecto das redes neurais, sendo possível assim listar algumas palavras chave a elas associadas, e que podem resumir suas características:

- Adaptabilidade;
- Inspiradas no neurônio biológico;
- Solução para problemas não lineares.

As soluções adaptativas são desejáveis em várias situações apresentando a habilidade de se adaptarem a extensas quantidades de dados, principalmente quando os parâmetros dos sistemas são variáveis ao longo do tempo. Tal característica implica na possibilidade de aplicação de redes neurais a soluções computadas “on-line”.

A capacidade de lidar com não-linearidade é um requisito fundamental à ciência e à engenharia atual. Em vários campos, os métodos não lineares são desenvolvidos caso a caso. As redes neurais podem gerar soluções mais gerais para estes problemas, fazendo desta técnica um dos mais atrativos instrumentos de resolução de problemas (Haykin, 1999 – Arbib, 1995 – Hecht, 1990).

Num contexto histórico, pode-se destacar alguns pesquisadores envolvidos no estudo e aplicação de redes neurais desde McCulloch e Pitts em 1943 até Chua e Yang em 1988 que encontram-se resumidamente nos anexos I e II deste trabalho, onde são ressaltados o período cronológico e atividade científica de cada pesquisador na área de redes neurais, verificando suas influências nos trabalhos mais recentes.

2.3.1 Modelo de um neurônio (*perceptron*)

Em termos matemáticos (Haykin, 1999), pode-se descrever o modelo de um neurônio k (Figura 2.3), composto de p entradas de valores x , onde w são os pesos sinápticos, u é a combinação linear da saída, b é o *bias*, podendo este ser ou não considerado na expressão. $\varphi(.)$ é a função de ativação, e y é a saída, com as seguintes equações:

$$u_k = \sum_{j=0}^p w_{kj} x_j$$

$$y_k = \varphi(u_k - \theta_k)$$

Quando for utilizado o limiar de ativação θ , este aplica-se ao potencial de ativação ou nível de atividade v_k , tornando-se uma transformação relativa:

$$v_k = u_k - \theta_k$$

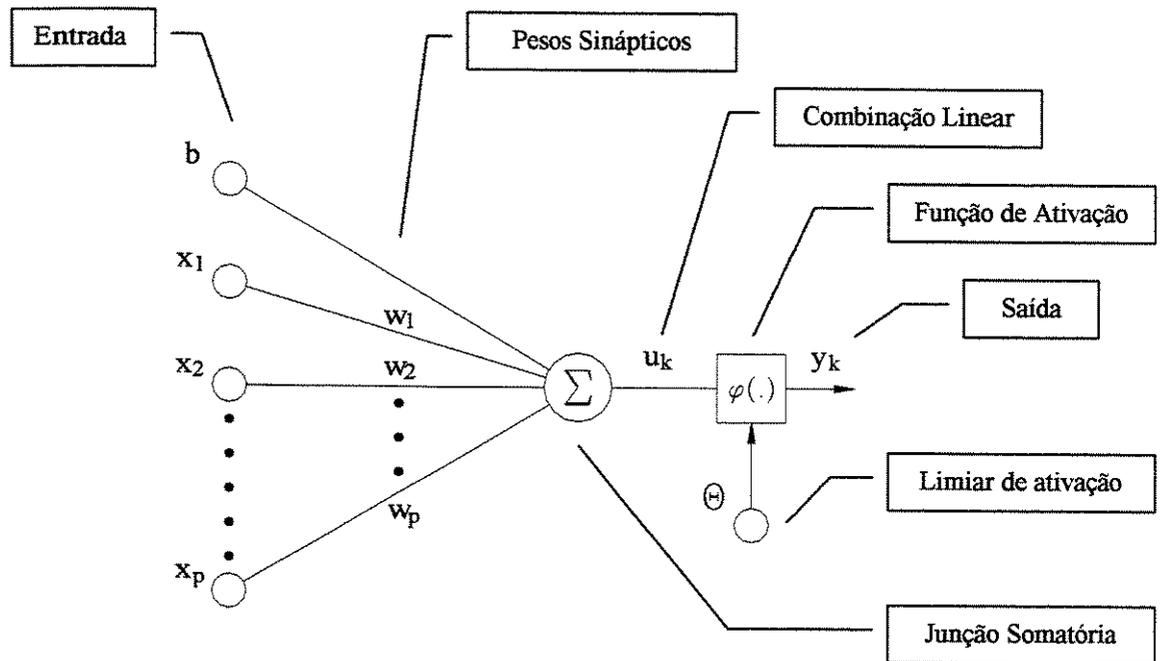


Figura 2.3 – Perceptron

2.3.2 Tipos de função de ativação

Diversas funções de ativação são normalmente usadas. As mais comuns são apresentadas a seguir.

- Função Linear

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases}$$

$$y_k = \begin{cases} 1 & \text{se } v_k \geq 0 \\ 0 & \text{se } v_k < 0 \end{cases}$$

Onde o nível de atividade interno v_k do neurônio é:

$$v_k = \sum_{j=1}^p w_{kj} x_j - \theta_k$$

- Função de Intervalo Linear

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq +1/2 \\ v & \text{se } +1/2 > v > -1/2 \\ 0 & \text{se } v \leq -1/2 \end{cases}$$

- Função Sigmóide

Função logística, onde a é o parâmetro de inclinação:

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$

As vezes deseja-se obter a função de ativação variando de -1 a $+1$, a qual assume uma forma anti-simétrica, definindo a função limiar:

$$\varphi(v) = \begin{cases} 1 & \text{se } v > 0 \\ 0 & \text{se } v = 0 \\ -1 & \text{se } v < 0 \end{cases}$$

- Função tangente hiperbólica:

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1 - \exp(-v)}{1 + \exp(-v)}$$

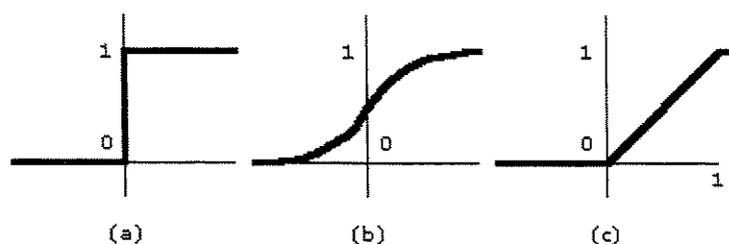


Figura 2.4: Funções de ativação: (a) linear; (b) sigmoide; (c) intervalo linear

2.3.3 Arquitetura da rede em múltiplas camadas

A forma estrutural dos neurônios de uma rede neural está diretamente ligada ao algoritmo de aprendizagem usado para treinar a rede. Existem diversos algoritmos de aprendizagem e

abordagens para cada tipo de aplicação (anexo III), optando-se pela utilização de redes com realimentação de múltiplas camadas para este trabalho. Esta classe de redes é distinta por apresentar uma ou mais camadas intermediárias. Os neurônios intermediários interferem na saída da rede quando excitada por entradas externas. A utilização de camadas intermediárias amplia o campo de atuação de uma rede, fazendo com que suas interligações sinápticas interajam sobre toda a estrutura.

Os sinais adquiridos na camada de entrada fornecem elementos com característica de ativação, os quais constituem os sinais de entrada ao neurônio da segunda camada. Os sinais de saída desta última tornam-se as entradas da terceira camada e assim sucessivamente para toda a rede. Os sinais da saída da última camada da rede constituem a resposta geral conforme as características de ativação dos sinais de entrada da primeira camada.

As redes de múltiplas camadas são constituídas de unidades de sensoriamento que são as entradas da primeira camada, uma ou mais camadas intermediárias que propagam os sinais camada a camada através de suas bases, e a última camada de saída que proporciona a resposta, sendo referenciadas como *perceptrons* de múltiplas camadas (Figura 2.5).

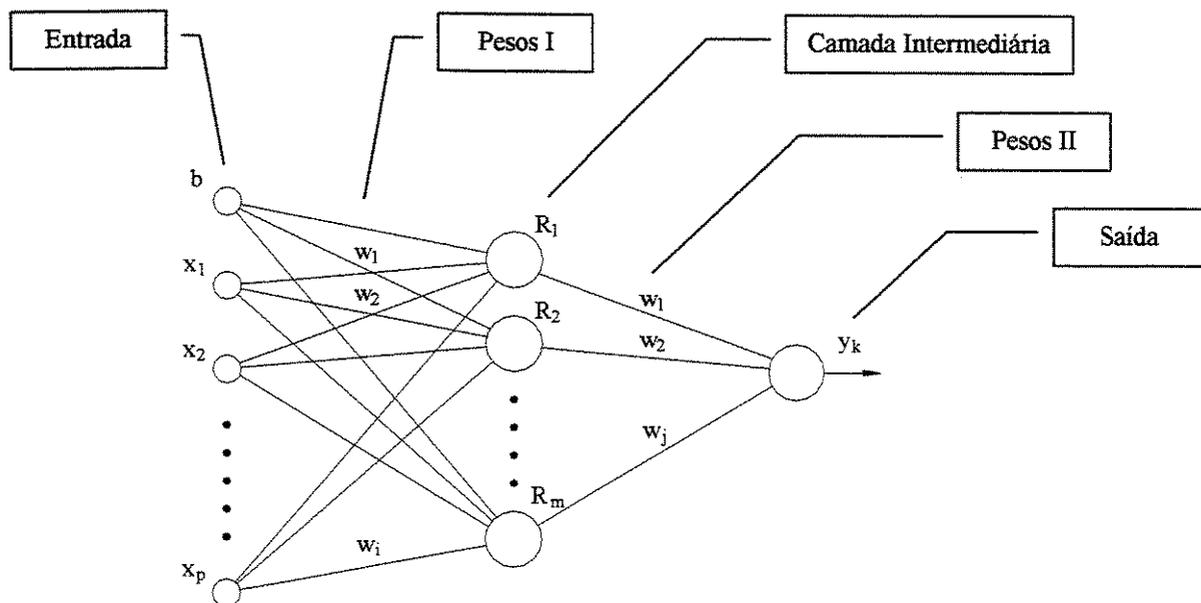


Figura 2.5 – *Perceptron* de múltiplas camadas

Os *perceptrons* de múltiplas camadas têm sido aplicados com êxito na resolução de problemas complexos, por seus treinamentos de maneira supervisionada através do conhecido algoritmo de retro-propagação, baseado na regra de aprendizagem de erro-correção.

2.3.4 Retro-propagação

A retro-propagação é uma técnica específica de utilização do gradiente no espaço-peso para uma rede alimentada em múltiplas camadas. A idéia básica desta técnica é calcular as derivadas parciais de uma função de aproximação realizada pela rede em relação a todos elementos do vetor peso ajustável para um dado valor do vetor de entrada.

Os elementos do vetor peso são ordenados por camadas, depois pelos neurônios de cada camada, e então pelo número de sinapses do neurônio.

O algoritmo de retro-propagação tem sido o mais utilizado para treinamento supervisionado de *perceptrons* de múltiplas camadas, sendo basicamente uma técnica de gradiente e não uma técnica de otimização. Este algoritmo realiza o cálculo do gradiente no sentido inverso da direção da rede, baseando-se nos erros calculados em sua saída, como uma regra de cadeia, retornando o valor desejado em sua camada de entrada, e assim, recalculando a rede. Pode ser usado para a atualização dos pesos processando seus dados após cada iteração para se obter a convergência de um ponto estacionário desejado, onde a taxa de aprendizagem utilizada deve ser lentamente reduzida. Esta propriedade de convergência comporta-se como uma aproximação estocástica. Uma alta taxa de aprendizagem faz com que haja uma divergência na função que atualiza os pesos e da função que calcula o erro.

2.3.5 Treinamento supervisionado

O processo de treinamento supervisionado, isto é, quando é utilizado um agente externo que indica à rede a resposta desejada para o padrão de entrada, opera em uma rede de alimentação direta (*feedforward*) regular, como é representado na Figura 2.6. Geralmente, as redes são

constituídas de vários neurônios agrupados em camadas. Cada unidade contém um estado ou nível de ativação que é determinado pela entrada recebida de outras unidades da rede.

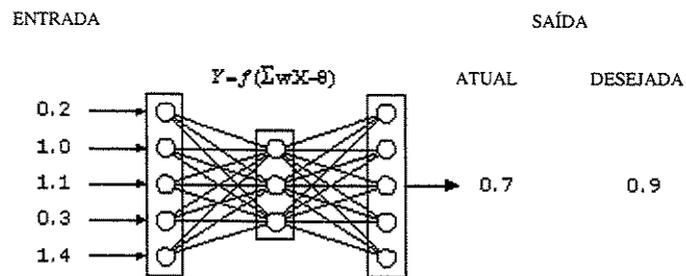


Figura 2.6 – Rede completamente interconectada, com uma camada intermediária

2.3.6 Algoritmo de retro-propagação

O sinal erro e na saída do neurônio j com n iterações é definido com a diferença da saída desejada d pela saída real y do sinal:

$$e_j(n) = d_j(n) - y_j(n)$$

Derivando ambos os lados da equação anterior em relação a $y_j(n)$, obtém-se:

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1$$

A soma instantânea dos erros ao quadrado ε é dada por:

$$\varepsilon(n) = \frac{1}{2} \sum_j e_j^2(n)$$

Derivando ambos os lados da equação anterior em relação a $e_j(n)$, obtém-se:

$$\frac{\partial \varepsilon(n)}{\partial e_j(n)} = e_j(n)$$

A média dos erros quadráticos depende do número de amostras N :

$$\varepsilon_m = \frac{1}{N} \sum_{n=1}^N \varepsilon(n)$$

O nível de atividade interna da rede v na entrada da não linearidade, associada ao neurônio j é portanto:

$$v_j(n) = \sum_{i=0}^P w_{ji}(n) y_i(n)$$

Derivando a equação anterior em relação a $w_{ji}(n)$:

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

Daí a função sinal y aparece na saída do neurônio j , na iteração n que é dada por:

$$y_j(n) = \varphi(v_j(n))$$

E depois, derivando a equação anterior em relação a $v_j(n)$, obtém-se:

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi'_j(v_j(n))$$

O algoritmo de retro-propagação aplica uma correção ao peso sináptico (atualização dos pesos) proporcionando um gradiente $\partial \mathcal{E}(n) / \partial w_{ji}(n)$, representando assim, um fator de sensibilidade de acordo com a regra da cadeia, onde teremos:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

Substituindo as frações da equação anterior:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -e_j(n) \varphi'_j(v_j(n)) y_i(n)$$

A regra delta Δw_{ji} aplicada a w_{ji} , considerando a taxa de aprendizagem η do algoritmo de retro-propagação, onde o sinal *negativo* indica um gradiente decrescente em espaço peso, define-se:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$$

Substituindo a equação anterior, tem-se:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$$

Define-se o gradiente $\delta_j(n)$ da saída, que indica a mudança do peso sináptico, como:

$$\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n) \varphi_j'(v_j(n))$$

Para o neurônio que encontra-se na camada intermediária é determinado o sinal erro de todos os neurônios conectados a este, sendo assim definida a retro-propagação, onde para o gradiente local δ do neurônio intermediário i :

$$\delta_i(n) = -\frac{\partial \mathcal{E}(n)}{\partial y_i(n)} \frac{\partial y_i(n)}{\partial v_i(n)} = -\frac{\partial \mathcal{E}(n)}{\partial y_i(n)} \varphi_i'(v_i(n))$$

Com o neurônio de saída k , tem-se:

$$\mathcal{E}(n) = \frac{1}{2} \sum_k e_k^2(n)$$

Derivando a equação anterior em relação a $y_i(n)$, obtém-se:

$$\frac{\partial \mathcal{E}(n)}{\partial y_i(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_i(n)}$$

Utilizando a regra da cadeia para a derivada parcial $\partial e_k(n)/\partial y_i(n)$ à equação anterior:

$$\frac{\partial \mathcal{E}(n)}{\partial y_i(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_i(n)}$$

Fazendo o erro como a diferença entre a saída desejada d_k e a real y_k :

$$e_k(n) = d_k(n) - y_k(n) = d_k(n) - \varphi_k(v_k(n))$$

Derivando a equação anterior em relação a $v_k(n)$:

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi'_k(v_k(n))$$

O nível de atividade interna da rede, com p entradas é então:

$$v_k(n) = \sum_{i=0}^p w_{ki}(n) y_i(n)$$

Derivando a equação anterior em relação a $y_i(n)$:

$$\frac{\partial v_k(n)}{\partial y_i(n)} = w_{ki}(n)$$

Substituindo as equações, tem-se a derivada parcial desejada:

$$\frac{\partial \mathcal{E}(n)}{\partial y_i(n)} = -\sum_k e_k(n) \varphi'_k(v_k(n)) w_{ki}(n) = -\sum_k \delta_k(n) w_{ki}(n)$$

Finalmente, substituindo a equação anterior, obtém-se o gradiente local δ do neurônio intermediário i :

$$\delta_i(n) = \varphi'_i(v_i(n)) \sum_k \delta_k(n) w_{ki}(n)$$

Resumindo as relações do algoritmo de retro-propagação, a correção $\Delta w_{ji}(n)$ aplicada ao peso sináptico que está conectando o neurônio intermediário i ao neurônio j é definido pela regra delta:

$$\Delta w_{ij}(n) = \eta \delta_i(n) y_j(n)$$

2.4 Sistemas dinâmicos

Pode-se resumir sistemas dinâmicos pela interação do sistema ou mecanismo com o meio ambiente. O sistema será influenciado pelo aspecto do ambiente local – entradas do sistema –

enquanto a atividade do ambiente será responsável pelos aspectos da atividade corrente do sistema – saídas do sistema. As entradas e saídas que na verdade entram na teoria do sistema são uma pequena amostra do fluxo de interações com o resto do universo. Um sistema real contém cinco elementos:

- Conjunto de entradas: variáveis do ambiente que afetará diretamente o comportamento do sistema.
- Conjunto de saídas: variáveis do sistema que serão observadas e afetadas significativamente pelo ambiente.
- Conjunto de estados: variáveis internas do sistema (que podem ou não serem variáveis de saídas) que determinam a relação entre a entrada e a saída. Essencialmente, o estado do sistema é o “resíduo interno do passado” ("internal residue of the past") do sistema, ou seja, quando conhecemos esse estado, temos informação sobre o seu comportamento, facilitando prever a forma com que as entradas e saídas futuras estão sendo relacionadas.
- Função de transição de estado: é a função que determina como o estado mudará quando o sistema obtiver entradas variadas.
- Função de saída: é a função que determina o que a saída do sistema produzirá com uma dada entrada e um dado estado.

Pode-se dizer que os sistemas dinâmicos consideram as variações do sistema com o tempo, e podem ser derivados do fluxo de informação, fluxo de energia ou mesmo de sistemas puramente matemáticos. O tratamento matemático de sistemas dinâmicos utiliza o conceito de equações diferenciais, fazendo com que os fundamentos físicos da análise de sistemas dinâmicos estejam relacionados com a geração e análise de equações diferenciais.

Uma equação diferencial pode ser linear ou não linear distinguindo-se quando é linear se, e somente se, cada termo da equação é de primeiro grau com relação a suas variáveis dependentes ou com relação a qualquer derivada de suas derivadas dependentes. Por outro lado, uma equação diferencial não linear é qualquer equação diferencial que não seja linear em relação a alguma de

suas variáveis dependentes. Sistemas físicos são geralmente não lineares, proporcionando assim uma motivação para seu estudo.

2.4.1 Sistema dinâmico em tempo contínuo

Em mecanismos, o estado do sistema compreende as posições de seus componentes que são diretamente observáveis e suas velocidades que podem ser estimadas em uma trajetória observada sobre um período de tempo. O tempo é contínuo, isto é, caracterizado por um conjunto de números reais, em que as mudanças de estado são descritas por uma equação diferencial. O tipo básico de sistemas em tempo contínuo produz um mecanismo clássico em que o estado presente e a saída determinam a taxa de mudança de estado. Isto exige que a saída, a entrada, e espaços de estado sejam espaços contínuos em que mudanças contínuas possam ocorrer. Considere o simples exemplo de um ponto de massa em movimento retilíneo; em determinado tempo, sua posição $y(t)$ é a saída observável do sistema, e a força $u(t)$ agindo sobre ele é a força aplicada ao sistema; a segunda lei de Newton diz que a força aplicada ao sistema iguala-se ao produto de sua massa e sua aceleração: $\ddot{y}(t) = m^{-1}u(t)$, onde a aceleração $\ddot{y}(t)$ é a segunda derivada de $y(t)$. De acordo com a Lei de Newton, o estado do sistema é dado pela posição e velocidade da partícula. Chamamos o par posição-velocidade no tempo, de estado instantâneo $q(t)$ do sistema. Uma equação simples nos fornece informação suficiente para deduzir a taxa de mudança $dq(t)/dt$ deste estado. Usando a matriz, temos:

$$\frac{d}{dt} \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \dot{y}(t) \\ \ddot{y}(t) \end{bmatrix} = \begin{bmatrix} \dot{y}(t) \\ m^{-1}u(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t)$$

enquanto

$$y(t) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T q(t)$$

Este é um exemplo de sistema linear em que a taxa de mudança de estado depende linearmente do estado e da entrada presentes, e a saída presente depende linearmente do estado presente, isto é, há matrizes A, B, e C tal que:

$$\frac{dq(t)}{dt} = Aq(t) + Bu(t)$$

$$y(t) = Cq(t)$$

Geralmente, um sistema físico pode ser expresso genericamente pelo par de equações:

$$\frac{dq(t)}{dt} = f(q(t), u(t))$$

$$y(t) = g(q(t))$$

A primeira expressa a taxa de mudança $dq(t)/dt$ do estado como uma função do estado $q(t)$ e a entrada ou o vetor de controle $u(t)$ expresso no tempo t .

2.4.2 Redes neurais em sistemas dinâmicos de tempo contínuo

Agora apresenta-se a definição de um neurônio integrador de escoamento (*leaky integrator neuron*) como um sistema em tempo contínuo. O estado interno do neurônio é seu potencial de membrana $m(t)$, e sua saída é a taxa de ativação (*firing rate*) $M(t)$. A função de transição de estado da célula é expressa como:

$$\tau \frac{dm(t)}{dt} = -m(t) + \sum_j w_j X_j(t) + h_j$$

enquanto a função de saída da célula é dada pela equação:

$$M(t) = \sigma(m(t))$$

Assim, havendo m entradas $X_i(t)$, $i=1, \dots, m$, o espaço de entrada do neurônio é \mathbf{R}^m , com valor atual $(X_1(t), \dots, X_m(t))$, enquanto os espaços de estado e a saída do neurônio são iguais a \mathbf{R} , com valores atuais $m(t)$ e $M(t)$, respectivamente.

Será visto brevemente como uma rede neural compreendida de neurônios integradores de escoamento também podem ser vistos como sistemas em tempo contínuo. Como mostrado na

Figura 2.7, caracteriza-se uma rede neural pela escolha de N neurônios, que podem ser separados em várias camadas carregando seus sinais, cada conexão e cada linha para uma única entrada de um outro neurônio ou alimentação externa à rede para melhorar uma das K linhas de saída. Assim cada entrada de um dado neurônio deve ser conectada a cada saída de um outro neurônio ou a uma das L linhas de entrada da rede. Então temos o conjunto de entradas $X \in \mathbf{R}^L$, o conjunto de estado $Q \in \mathbf{R}^N$, e o conjunto de saída $Y \in \mathbf{R}^K$. Se a i -ésima linha de saída vem do j -ésimo neurônio, então a função de saída é determinada pelo fato de que o i -ésimo componente da saída no tempo t seja a taxa de ativação $M_j(t) = \sigma_j(M_j(t))$ do j -ésimo neurônio no tempo t . A função de transição de estado para a rede neural segue das funções de transição de estado de cada neurônio N :

$$\tau \frac{dm_i(t)}{dt} = -m_i(t) + \sum_j w_{ij} X_{ij}(t) + h_i$$

tem-se que $X_{ij}(t)$ é a saída $M_k(t)$ do k -ésimo neurônio ou o valor $x_i(t)$ atualmente sendo aplicado na l -ésima linha de entrada da rede total (Arbib, 1995).

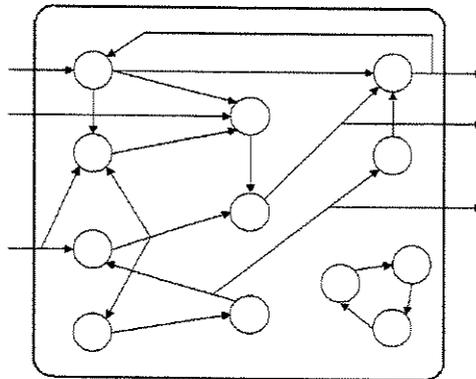


Figure 2.7 – Uma rede neural vista como um sistema. A entrada no tempo t é a característica de ativação nas linhas de entrada; a saída é característica de ativação nas linhas de saída; e o estado interno é o vetor de taxas de ativação de todos os neurônios da rede.

2.4.3 Redes neurais em sistemas dinâmicos de tempo discreto

Em contraste com o sistema em tempo contínuo, que deve ter espaços de estado contínuo, no qual as equações diferenciais para a função de transição de estado podem ser definidas, os sistemas em tempo discreto devem ter qualquer espaço de estado contínuo ou discreto. Um

espaço de estado discreto é um conjunto sem estrutura métrica específica ou topológica. Por exemplo, um neurônio de McCulloch-Pitts é considerado para operar numa escala em tempo discreto, com $t = 0, 1, 2, 3, \dots, m$ linhas de entrada binária e somente uma linha de saída binária. Tal neurônio tem um conjunto de entrada $\{0, 1\}^m$, estado e saída $\{0, 1\}$. O neurônio é caracterizado pelos pesos de conexão e limiar θ (*threshold*). Se num tempo t o valor da i -ésima entrada é $x_i(t)$, então o estado e a saída do próximo instante $y(t+1)$, é igual a 1 se e somente se $\sum_i w_i x_i(t) \geq \theta$ (Arbib, 1995).

O esquema de aprendizagem conhecido como retro-propagação é baseado em neurônios que estão em tempo discreto, mas com entradas, estado e saída tomados com valores contínuos em algumas variações.

Na ciência da computação, uma automação é um sistema com entrada, espaços de estado e saída em tempo discreto. Descrevemos uma automação pelo conjunto de X entradas, Y saídas e Q estados, juntos com a função do próximo estado $\delta: Q \times X \rightarrow Q$, e a função de saída $\beta: Q \rightarrow Y$. Se o autômato (manipulador) encontra-se no estado q e recebe entrada x no tempo t , então o próximo estado será $\delta(q, x)$ e sua próxima saída será $\beta(q)$. Uma rede neural de McCulloch-Pitts é uma rede semelhante à representada na Figura 2.4, no entanto uma rede em tempo discreto tem cada neurônio como um neurônio de McCulloch-Pitts. Maiores detalhes sobre autômatos finitos podem ser encontrados em (McCulloch and Pitts, 1943).

2.4.4 Formulação de *Lagrange* para dinâmica de manipuladores

A formulação de Lagrange para dinâmica de manipuladores descreve o comportamento de um sistema dinâmico em termos de trabalho e energia armazenada no sistema, ao contrário das forças e momentos dos corpos individuais. As equações dinâmicas podem ser derivadas sistematicamente em qualquer sistema de coordenadas.

Considerando q_1, \dots, q_n as coordenadas generalizadas que posicionam um sistema dinâmico, a energia cinética K e a energia potencial V armazenadas no sistema, o *Lagrangeano* é definido como:

$$L(q_i, \dot{q}_i) = K - V \quad i=1, \dots, n$$

Usando o *Lagrangeano*, a equação de movimento do sistema dinâmico é dado, conforme a equação de *Lagrange*, por:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i \quad i=1, \dots, n$$

onde Q_i é a força generalizada correspondente à coordenada generalizada q_i .

Entende-se por força generalizada, uma força hipotética que é aplicada na coordenada generalizada e produz um trabalho, em relação ao seu deslocamento, equivalente ao realizado pelo conjunto de todas as forças externas.

Para melhor entendimento destas equações, segue sua aplicação no estudo do manipulador proposto neste trabalho, no Capítulo 3, como também pode-se observar exemplos citados em Asada, 1986.

2.5 Identificação de sistemas

2.5.1 Identificabilidade

Enfatiza-se brevemente a seguir o termo *identificabilidade* entre os vários outros conceitos em sistemas de controle como observabilidade e controlabilidade discutidos por diversos autores, Elgerd (1967); Lee (1964); Kalman (1960). O processo em discussão é tratado pelo uso das equações:

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k)$$

onde x é um vetor de dimensão n .

O processo é chamado *identificável* se for possível determinar a matriz A do sistema pela medição das variáveis de estado.

$$\begin{aligned}x(1) &= Ax(0) \\x(2) &= A^2x(0) \\&\vdots \\x(n) &= A^n x(0)\end{aligned}$$

ou

$$[x(1):x(2):\dots:x(n)] = A[x(0):Ax(0):\dots:A^{n-1}x(0)]$$

Os vetores x são conhecidos como a única solução para que A possa ser encontrado somente se a seguinte matrix tem ordem (*rank*) n :

$$[x(0):Ax(0):\dots:A^{n-1}x(0)]$$

O conceito de identificabilidade encontra-se de forma mais detalhada em diversas obras, por exemplo em Eykhoff (1974).

2.5.2 Comentários sobre modelos de identificação e redes neurais

A identificação de sistemas por redes neurais até o momento é alvo de pesquisa em andamento e ainda não muito difundida, como discutido em Arbib(1995). O número de camadas a ser usado na rede bem como o número de neurônios contidos em cada camada são escolhidos por tentativa e erro através de testes e experimentação. Há também diversos métodos, geralmente baseados no gradiente, para o ajuste dos parâmetros, e o critério de performance também varia conforme o método e estrutura da rede.

$$\sum_k [e_i^2(k) + (u(k) - u(k-1))^2]$$

onde, no sistema em tempo discreto, k representa o tempo, e o erro na saída e u é a entrada do sistema.

Quando as variáveis de estado da planta são conhecidas, o modelo da planta pode ser representado por (modelo paralelo):

$$\hat{x}(k+1) = N_f[\hat{x}(k), u(k)]$$

$$\hat{y}(k) = N_h[\hat{x}(k)]$$

onde $\hat{x}(k)$ e $\hat{y}(k)$ são os estimadores de $x(k)$ e $y(k)$, e N_f e N_h são as redes neurais usadas na aproximação dos mapeamentos de f e h respectivamente. As redes N_f e N_h são treinadas usando os erros $x(k) - \hat{x}(k)$ para N_f e $y(k) - \hat{y}(k)$ para N_h . Um modelo usado como alternativa tem a forma (modelo série-paralelo):

$$\hat{x}(k+1) = N_f[x(k), u(k)]$$

$$\hat{y}(k) = N_h[x(k)]$$

sendo que este modelo usa o vetor de estado conhecido $x(k)$ no lugar de $\hat{x}(k)$. Do ponto de vista prático, a identificação de sistemas pelo modelo em série-paralelo é substancialmente melhor, por sua superioridade no tratamento com o método usado ao ajuste de parâmetros de sua rede com a retro-propagação estática, ao invés da determinação das derivadas parciais do erro (método do gradiente) em relação a seus parâmetros equivalendo-se à retro-propagação dinâmica ou redes recorrentes, onde a sua aplicação torna-se mais complexa.

2.5.3 Identificação da dinâmica inversa de sistemas utilizando redes neurais

A utilização de redes neurais na identificação de um sistema não linear pela dinâmica inversa é mostrada na Figura 2.9. Em contraste à situação direta apresentada na Figura 2.8, a entrada da rede é a saída do processo, e os valores desejados na saída da rede corresponde a uma aproximação do sinal de entrada do processo.

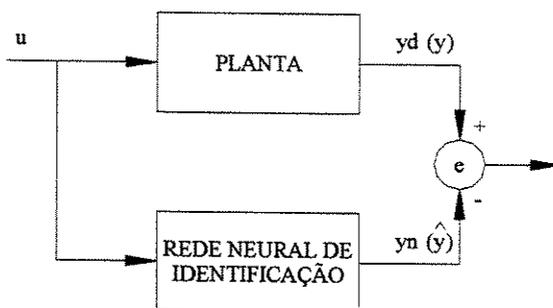


Figura 2.8 – Identificação de sistemas utilizando redes neurais

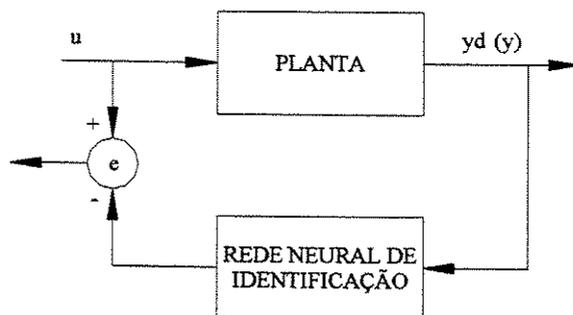


Figura 2.9 – Identificação de sistemas pela dinâmica inversa utilizando redes neurais

2.6 Linearização por realimentação

2.6.1 Introdução

A linearização por realimentação é uma abordagem muito utilizada em controle de sistemas não lineares encontrada em estudos publicados nos últimos anos. Este método generaliza o conceito de dinâmica inversa de manipuladores rígidos, isto é, podem ser entendidos como a mesma metodologia. A idéia básica da linearização por realimentação é construir uma lei de controle que linearize o sistema não linear após uma mudança de coordenadas do espaço de estado. Este método diferencia-se dos métodos tradicionais (exemplo: Linearização Jacobiana conforme Slotine-1991) quando a linearização por realimentação é obtida pela exata transformação de estado e sua realimentação, ao contrário da aproximação linear da dinâmica.

Com este método, busca-se simplificar a forma da dinâmica do sistema pela escolha de uma representação diferente de estado. A técnica de linearização por realimentação pode ser vista

como uma forma de “transformação dos modelos de sistemas originais em modelos equivalentes de uma forma mais simples”. A linearização por realimentação tem sido aplicada a alguns problemas práticos como: controle de helicópteros; alta performance em aviões, robôs industriais e mecanismos biomecânicos. Apesar das inúmeras vantagens da linearização por realimentação, ainda estão sendo estudadas algumas limitações e abordagens da técnica, tomando-se isto como um estímulo para o seu crescente desenvolvimento nesta área.

Como idéia geral, é dado abaixo um sistema em malha fechada apresentando duas realimentações, sendo a interna de linearização da entrada-estado, e a externa a estabilização da dinâmica em malha fechada, onde a entrada de controle u pode ser vista como uma parte da transformação da não linearidade e uma parte de compensação linear (Figura 2.10).

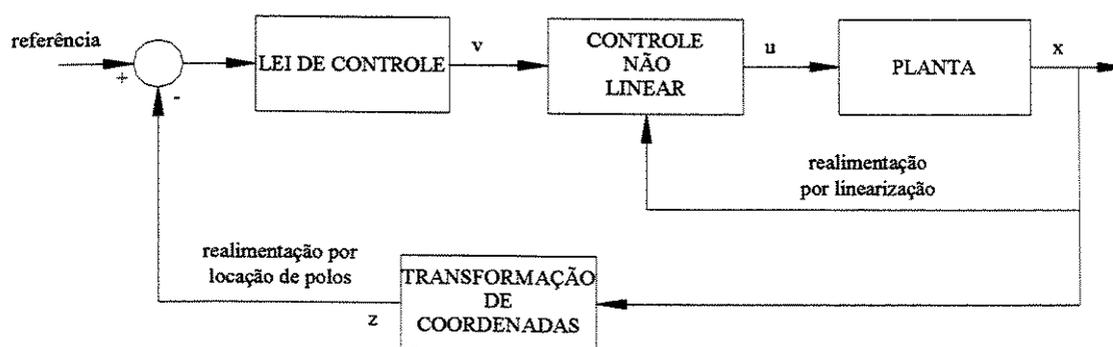


Figura 2.10 – Linearização de entrada-estado

Segundo Spong (1989), no caso de manipuladores rígidos o controle dinâmico inverso e o controle linearizado por realimentação são o mesmo. Somente para facilitar o entendimento da linearização por realimentação, será brevemente apresentado o sistema de simples entrada-saída mas o sistema abordado neste trabalho será tratado como sistema de múltiplas entradas e saídas.

2.6.2 Sistemas de linearização de simples entrada-saída (SISO)

Uma das principais propostas deste trabalho é analisar e projetar a lei de controle por realimentação para sistemas não lineares. Em quase todas as situações, assumimos que o estado x do sistema esteja disponível para medições, e tomamos a entrada do sistema dependendo deste estado e, possivelmente, de sinais externos de referência. Se o valor do controle no tempo t

depende somente de valores no mesmo instante de tempo do estado x e da entrada externa de referência, o controle é dito como “Modo de Controle por Realimentação em Estado Estático”. Por outro lado, se o controle depende também de um conjunto adicional de variáveis de estado, isto é, se este controle recebe a própria saída do sistema dinâmico, tendo seu próprio estado interno, pode-se dizer que este é um “Modo de Controle por Realimentação em Estado Dinâmico”(Isidori, 1989).

Neste trabalho apresenta-se apenas a aplicação da parte teórica direcionada ao problema de linearização do manipulador. Uma completa descrição do método de linearização por realimentação pode ser encontrada em Spong e Vidyasagar (1989), Slotine e Li (1991), entre outros.

O controle por realimentação de estado estático (lei de controle não linear) é:

$$u = \alpha(x) + \beta(x)v$$

onde v é a “entrada equivalente” de referência da linearização utilizada para determinar u .

O sistema não linear é representado pela equação de estado:

$$\dot{x} = f(x) + g(x)u$$

e a transformação não linear de estado é:

$$y = h(x)$$

onde y é a saída do sistema. Pela linearização do sistema, generalizamos uma relação diferencial linear entre a saída y e uma nova entrada v .

O controle com o sistema produz uma malha fechada com estrutura:

$$\dot{x} = f(x) + g(x)\alpha(x) + g(x)\beta(x)v$$

$$y = h(x)$$

As funções $\alpha(x)$ e $\beta(x)$ caracterizam o controle que pode ser visto na Figura 2.11:

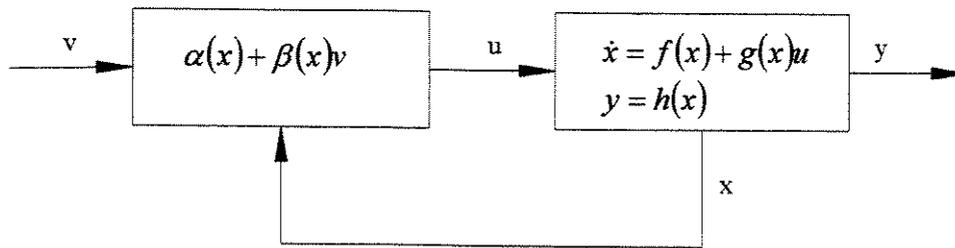


Figura 2.11– Linearização por realimentação

Considerando um sistema não linear, a mudança de coordenadas necessita construir a forma normal que é dada por:

$$\Phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_n(x) \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix}$$

isto é, pela função $h(x)$ e sua primeira derivada por $f(x)$. O $L_f h$ é a nova função escalar, ou seja, a derivada de h em relação a f (“Lie Bracket” em Slotine, 1991). A seguir, é necessário completar em ordem, as transformações em novas coordenadas:

$$z_i = \phi_i(x) = L_f^{i-1}(x) \quad 1 \leq i \leq n$$

o sistema apresenta-se descrito pela equação na forma:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_{n-1} &= z_n \\ \dot{z}_n &= b(z) + a(z)u \end{aligned}$$

onde $z = (z_1, \dots, z_n)$, $\dot{z} = \phi(\dot{x})$, tomando-se a função $a(z)$ não nula.

Agora, segue a lei de controle com realimentação do estado:

$$u = \frac{1}{a(z)}(-b(z) + v)$$

O resultado do sistema em malha fechada segue as equações e o diagrama correspondente na Figura 2.12:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_{n-1} &= z_n \\ \dot{z}_n &= v \end{aligned}$$

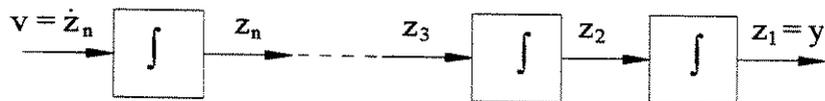


Figura 2.12 – Sistema de linearização

estando agora linear e controlável. Assim, conclui-se que certos sistemas não lineares podem ser transformados em um sistema linear e controlável, lembrando que esta transformação depende de dois passos básicos:

- Mudança de coordenadas, definida em função de \dot{x} ;
- Realimentação do estado, também definida em função de \dot{x} .

O objetivo da linearização por realimentação do sistema *SISO* é fazer com que a saída $y(t)$ acompanhe a saída desejada $y_d(t)$. Uma dificuldade deste modelo é que a saída y está indiretamente relacionada com a entrada u , através da variável de estado x e das equações não lineares de estado. Não é fácil visualizar como a entrada u pode ser projetada para controlar o comportamento de aproximação da saída y . Esta dificuldade pode ser reduzida se conseguirmos encontrar uma relação simples e direta entre a saída do sistema y e a entrada do controle u .

Maiores detalhes, podem ser obtidos na literatura sobre o método (Spong, 1989 – Slotine, 1991 – Isidori, 1989).

2.6.3 Sistemas de múltiplas entradas e múltiplas saídas (MIMO)

Esta seção ilustra como um sistema não linear com m entradas e n saídas pode ser transformado em um sistema linear e controlável pelo método de realimentação e mudança de coordenadas do espaço de estado estendendo-se ao que já foi exposto na seção anterior (SISO).

A versão de multivariável adequada da realimentação do estado considerada no problema de simples entrada-saída correspondente é um caso em que cada entrada u_i depende do estado x do sistema e das novas entradas de referência v_1, \dots, v_m como:

$$u_i = \alpha_i(x) + \sum_{j=1}^m \beta_{ij}(x)v_j$$

onde $\alpha_i(x)$ e $\beta_{ij}(x)$, para $1 \leq i, j \leq m$, são as funções de plano definidas em um subconjunto de \mathbf{R}^n . Note que o número de componentes da nova entrada de referência:

$$v = \text{col}(v_1, \dots, v_m)$$

tem sido escolhida exatamente igual ao número de componentes da entrada original u .

A composição das equações das novas entradas do sistema e a equação não linear produz um sistema de malha fechada com a mesma estrutura descrita pelas equações:

$$\dot{x} = f(x) + \sum_{i=1}^m g_i(x)\alpha_i(x) + \sum_{i=1}^m \left(\sum_{j=1}^m g_j(x)\beta_{ji}(x) \right) v_i$$

$$y_i = h_i(x)$$

$$\vdots$$

$$y_m = h_m(x)$$

A equação não linear de forma compacta:

$$u = \alpha(x) + \beta(x)v$$

em que

$$\alpha(x) = \begin{bmatrix} \alpha_1(x) \\ \vdots \\ \alpha_m(x) \end{bmatrix} \quad \beta(x) = \begin{bmatrix} \beta_{11}(x) & \cdots & \beta_{1m}(x) \\ \vdots & \ddots & \vdots \\ \beta_{m1}(x) & \cdots & \beta_{mm}(x) \end{bmatrix}$$

são vetor m e matriz $m \times m$ respectivamente. A malha fechada pode ser rescrita de melhor forma:

$$\dot{x} = f(x) + g(x)\alpha(x) + g(x)\beta(x)v$$

$$y = h(x)$$

Também foi assumido sistematicamente que a matriz $\beta(x)$ é não singular para todo x .

Geralmente, em caso de manipuladores com vários corpos, as equações representam um sistema não linear de múltiplas entradas, onde as condições de linearização por realimentação são mais difíceis, mesmo não mudando muito da aplicação nos sistemas de simples entrada. Pode-se também desacoplar o sistema, isto é, linearizar o sistema de tal forma que o resultado do sistema linear esteja composto de vários subsistemas, onde cada qual é afetado por um único sinal de entrada da malha externa (Spong, 1989).

Este sistema é melhor entendido quando usado exemplos para demonstrá-lo (conforme Spong, 1989 – Slotine, 1991 – Isidori, 1989). No capítulo posterior é feito o tratamento de linearização por realimentação para o manipulador estudado neste trabalho, tomando-se esse como exemplo para compreensão do método aplicado.

Capítulo 3

Modelagem teórica

3.1 Cinemática de corpos rígidos

O estudo da cinemática dos corpos rígidos deve-se à necessidade de conhecer, gerar, transmitir ou controlar o movimento do corpo, resultante das forças que nele são aplicadas. Um corpo rígido pode ser descrito como sendo um sistema de partículas para o qual as distâncias entre elas permanecem inalteradas. O movimento geral de um corpo rígido é composto por translação, que é definida como qualquer movimento no qual qualquer linha no corpo permaneça paralela a sua posição original, e rotação, que é o movimento angular onde as partículas e linhas do corpo movem-se em trajetória circular ao redor do eixo de rotação no mesmo ângulo ao mesmo tempo (Meriam, 1994).

3.1.1 Sistemas de referência

Nesta etapa, são aplicados os conceitos de cinemática relacionados aos sistemas de referência inercial I e móveis A e B , matrizes de transformação de coordenadas do sistema móvel para o inercial T^T ou o inverso T . Para isto são utilizados o vetor de posição q e sua velocidade \dot{v} ou \dot{w} , expressos tanto no sistema de referência inercial como nos móveis.

Para se descrever o movimento de um corpo tem-se primeiramente que definir o sistema de referência para que os vetores de posição e velocidade possam ser representados em relação a este

referencial. Pode-se também representar a aceleração deste movimento derivando-se a velocidade em função do tempo, mas nessa seção será tratada apenas a cinemática do manipulador. Este sistema de referência ou base, que pode ser inercial ou móvel, é definido como uma base vetorial representada por vetores unitários ou versores i, j e k representando os eixos de coordenadas X, Y e Z respectivamente.

3.1.2 Representação de um vetor no sistema de referência

Vetor posição: o vetor posição q é definido pelos versores i, j e k em grandezas escalares que indicam a amplitude deste vetor nas respectivas direções, devendo este ser representado no sistema de referência inercial após representado no sistema referencial móvel.

Vetor de velocidade: o vetor de velocidade absoluta \dot{q} é definido como a derivada do vetor posição, realizada com o vetor posição representado no sistema referencial inercial.

3.1.3 Representação do sistema

O esquema apresentado na Figura 3.1 representa um sistema robótico com 3 corpos e 3 graus de liberdade, tendo-se duas bases de rotação (A e B) e um movimento de translação (corpo 1), onde são definidos os sistemas de referência.

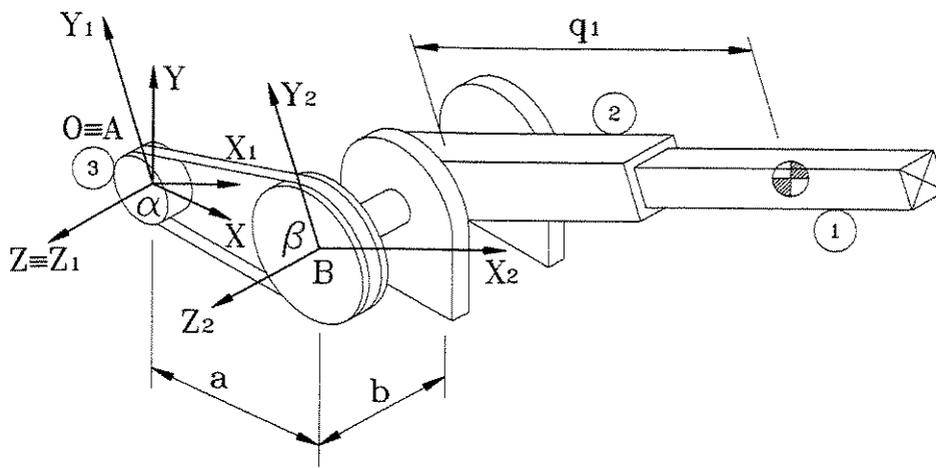
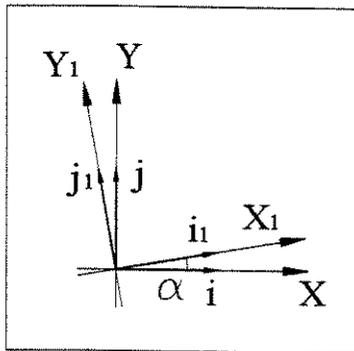


Figura 3.1 – Sistemas de referência

Definição dos sistemas de referência:

- Sistema inercial I (X, Y e Z), representado pelos versores i j k , respectivamente;
- Sistema móvel A (X_1 , Y_1 e Z_1), representado pelos versores i_1 j_1 k_1 , respectivamente;
- Sistema móvel B (X_2 , Y_2 e Z_2), representado pelos versores i_2 j_2 k_2 , respectivamente;

Os versores i_1 , j_1 e k_1 representam uma rotação positiva da base móvel A em torno do eixo Z (inercial) $\equiv Z_1$:



Equações do sistema de coordenadas:

$$i_1 = \cos \alpha * i + \text{sen } \alpha * j + 0 * k$$

$$j_1 = -\text{sen } \alpha * i + \cos \alpha * j + 0 * k \quad (\text{I})$$

$$k_1 = 0 * i + 0 * j + 1 * k$$

Matriz de transformação de coordenadas:

$$T_\alpha = \begin{Bmatrix} i_1 \\ j_1 \\ k_1 \end{Bmatrix} = \begin{bmatrix} \cos \alpha & \text{sen } \alpha & 0 \\ -\text{sen } \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} i \\ j \\ k \end{Bmatrix} \quad (\text{II})$$

Mudança dos sistemas de coordenadas:

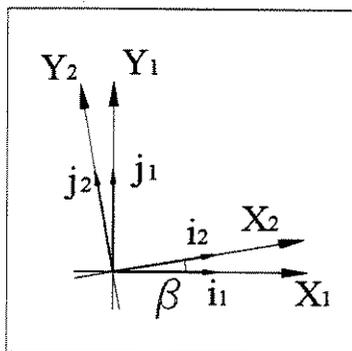
$${}_A S = T_\alpha * {}_I S$$

$${}_I S = T_\alpha^T * {}_A S \quad (\text{III})$$

Velocidade angular ω_1 do sistema móvel A (em rad/s) expressa na base inercial I :

$${}_1\omega_1 = \begin{Bmatrix} 0 \\ 0 \\ \dot{\alpha} \end{Bmatrix} \quad (\text{IV})$$

Os versores i_2 , j_2 e k_2 representam uma rotação positiva da base móvel B em torno do eixo Z_1 , // Z_2 :



Equações do sistema de coordenadas:

$$\begin{aligned} i_2 &= \cos \alpha * i_1 + \sin \alpha * j_1 + 0 * k_1 \\ j_2 &= -\sin \alpha * i_1 + \cos \alpha * j_1 + 0 * k_1 \\ k_2 &= 0 * i_1 + 0 * j_1 + 1 * k_1 \end{aligned} \quad (\text{V})$$

Matriz de transformação de coordenadas:

$$T_\beta = \begin{Bmatrix} i_2 \\ j_2 \\ k_2 \end{Bmatrix} = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} i_1 \\ j_1 \\ k_1 \end{Bmatrix} \quad (\text{VI})$$

Mudança dos sistemas de coordenadas:

$$\begin{aligned} {}_B S &= T_\beta * {}_A S \\ {}_A S &= T_\beta^T * {}_B S \end{aligned} \quad (\text{VII})$$

Velocidade angular ω_2 do sistema móvel B expressa na base móvel A :

$${}_A \omega_2 = \begin{Bmatrix} 0 \\ 0 \\ \dot{\beta} \end{Bmatrix} \quad (\text{VIII})$$

Velocidade angular ω_2 do sistema móvel B expressa na base inercial I :

$${}_I \omega_2 = {}_I \omega_1 + T_\alpha^T * {}_A \omega_2 = \begin{Bmatrix} 0 \\ 0 \\ \dot{\alpha} \end{Bmatrix} + \begin{bmatrix} \cos \alpha & -\text{sen } \alpha & 0 \\ \text{sen } \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ \dot{\beta} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \dot{\alpha} + \dot{\beta} \end{Bmatrix} \quad (\text{IX})$$

Vetor posição q_1 expresso na base móvel B :

$${}_B q_1 = \begin{Bmatrix} q_1 \\ 0 \\ -b \end{Bmatrix} \quad (\text{X})$$

Vetor posição q_1 expresso na base móvel A :

$$q_1 = T_\beta * {}_B q_1 + a = \begin{bmatrix} \cos \beta & -\text{sen } \beta & 0 \\ \text{sen } \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} q_1 \\ 0 \\ -b \end{Bmatrix} + \begin{Bmatrix} a \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} a + q_1 \cos \beta \\ q_1 \text{sen } \beta \\ -b \end{Bmatrix} \quad (\text{XI})$$

Vetor posição q_1 expresso na base inercial I :

$${}_I q_1 = T_\alpha^T * {}_A q_1 = \begin{bmatrix} \cos \alpha & -\text{sen } \alpha & 0 \\ \text{sen } \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} a + q_1 \cos \beta \\ q_1 \text{sen } \beta \\ -b \end{Bmatrix} \quad (\text{XII})$$

$${}_I q_1 = \begin{Bmatrix} (a + q_1 \cos \beta) \cos \alpha - q_1 \text{sen } \beta \text{sen } \alpha \\ (a + q_1 \cos \beta) \text{sen } \alpha + q_1 \text{sen } \beta \cos \alpha \\ -b \end{Bmatrix} \quad (\text{XIII})$$

Vetor velocidade do corpo 1 V expresso na base inercial I :

$${}_I V = \frac{d}{dt}({}_I q_1) \quad (\text{XIV})$$

$${}_I V = \begin{Bmatrix} -a \operatorname{sen} \alpha + \dot{q}_1 \cos \beta \cos \alpha - q_1 \dot{\beta} \operatorname{sen} \beta \cos \alpha - q_1 \dot{\alpha} \cos \beta \operatorname{sen} \alpha - \dot{q}_1 \operatorname{sen} \beta \operatorname{sen} \alpha - q_1 \dot{\beta} \cos \beta \operatorname{sen} \alpha - q_1 \dot{\alpha} \operatorname{sen} \beta \cos \alpha \\ a \cos \alpha + \dot{q}_1 \cos \beta \operatorname{sen} \alpha - q_1 \dot{\beta} \operatorname{sen} \beta \operatorname{sen} \alpha + q_1 \dot{\alpha} \cos \beta \cos \alpha + \dot{q}_1 \operatorname{sen} \beta \cos \alpha + q_1 \dot{\beta} \cos \beta \cos \alpha - q_1 \dot{\alpha} \operatorname{sen} \beta \operatorname{sen} \alpha \\ 0 \end{Bmatrix}$$

As equações (XIII) e (XIV) fornecem a posição e a velocidade absolutas do corpo 1 para qualquer combinação dos ângulos α e β , dentro do espaço de trabalho do manipulador, sabendo que o ângulo β é determinado em função do ângulo α e da constante de rigidez da correia definidos na formulação de *Lagrange* (item 3.2.1).

3.2 Modelagem dinâmica

O mecanismo analisado, representado na Figura 3.2, apresenta 3 GDL, onde é considerado um movimento de rotação q_3 , uma nova rotação q_2 e um movimento de translação q_1 , sendo estes acoplados por uma correia flexível de rigidez k . Desconsiderando o amortecimento, pode-se desmembrar a dinâmica do sistema para obter as equações de movimento pela formulação de *Lagrange*.

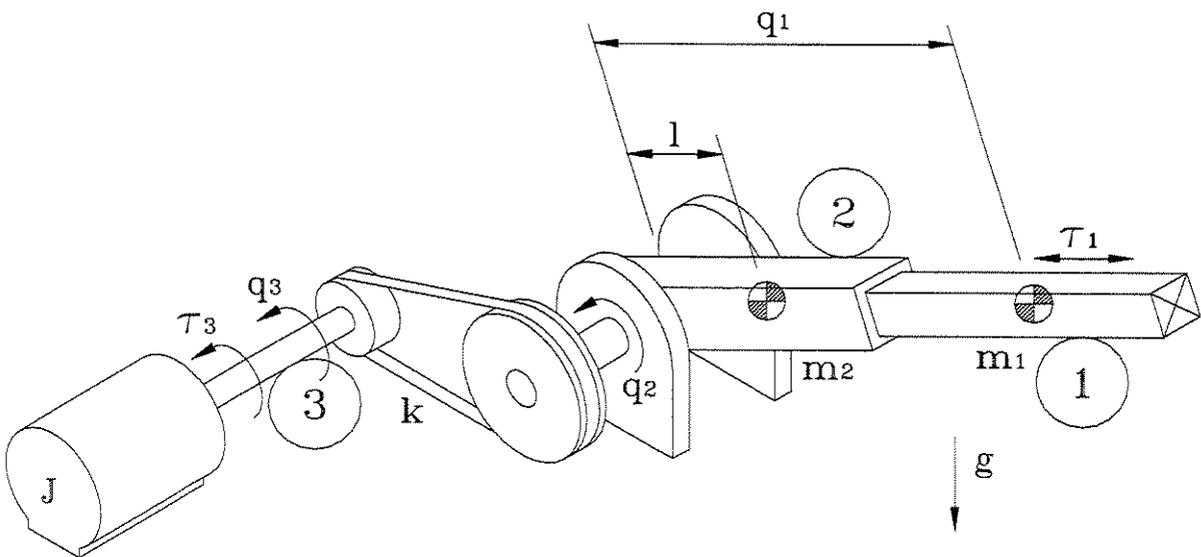


Figura 3.2 – Representação esquemática do sistema de 3 GDL

O mecanismo contém dois motores de acionamento sendo: um motor com inércia J diretamente acoplado ao corpo 3, fornecendo um torque τ_3 e gerando uma rotação q_3 ; e um motor fixo ao corpo 2 e acoplado ao corpo 1 através de uma rosca sem fim, que proporciona um movimento prismático do corpo 1 em relação ao corpo 2, fazendo q_1 ser variável. As massas dos corpos 1 e 2 serão representadas por m_1 e m_2 respectivamente. A distância l , entre o centro de massa do corpo 2 e o eixo de rotação Z_2 , permanece constante. O corpo 2 não está acoplado diretamente a nenhum motor, mas sofre influência da rotação do corpo 3 que é transmitida através da correia dentada, gerando assim a rotação q_2 .

3.2.1 Formulação de Lagrange

Nesta seção, será determinado o conjunto de equações que descreve o comportamento do sistema mecânico. Estas são as equações de movimento de Lagrange, que originam-se da diferença da energia cinética e potencial, conforme exposto por Craig(1989) e Spong(1989).

Energia cinética (K)

A energia cinética é o trabalho que deve ser executado sobre determinado corpo para levá-lo do repouso a uma certa velocidade.

$$K_{rotação} = \frac{1}{2} I w^2$$

$$K_{translação} = \frac{1}{2} m v^2$$

$$K_1 = \frac{1}{2} m_1 (\dot{q}_1^2 + \dot{q}_2^2) + \frac{1}{2} I_1 \dot{q}_2^2$$

Onde:

I_1 = Inércia do corpo 1;

I_2 = Inércia do corpo 2;

J = Inércia do motor;

k = rigidez da correia;

τ_1 = força de translação do corpo 1;

τ_3 = torque do corpo 3;

m_1 = massa do corpo 1;

m_2 = massa do corpo 2;

g = aceleração da gravidade;

q_1 = deslocamento do corpo 1;

\dot{q}_1 = velocidade linear do corpo 1;

l = posição do corpo 2;

q_2 = ângulo do corpo 2;

\dot{q}_2 = velocidade angular do corpo 2;

q_3 = ângulo do corpo 3;

\dot{q}_3 = velocidade angular do corpo 3;

q, \dot{q}, \ddot{q} = variáveis cinemáticas;

v = velocidade linear;

w = velocidade angular;

h = altura de elevação;

x = deformação da mola;

$$K_2 = \frac{1}{2} m_2 l^2 \dot{q}_2^2 + \frac{1}{2} I_2 \dot{q}_2^2$$

$$K_3 = \frac{1}{2} J \dot{q}_3^2$$

Energia potencial (V)

A energia potencial elástica é o trabalho feito sobre uma mola para deformá-la x a partir de seu comprimento inicial, igualando-se à energia que dela pode ser recuperada.

A energia potencial gravitacional é o trabalho mgh feito contra o campo gravitacional para elevar um determinado corpo de uma distância h acima de algum plano arbitrário de referência.

$$V_{elástica} = \frac{1}{2} Kx^2$$

$$V_{gravitacional} = m \cdot g \cdot \Delta h$$

$$V_1 = m_1 g q_1 \text{ sen } q_2 + m_1 g q_{1máx}$$

$$V_2 = m_2 g l \text{ sen } q_2 + m_2 g l$$

$$V_3 = \frac{k}{2} (q_2 - q_3)^2$$

Lagrangeano (L)

$$L = K - V$$

$$L = \frac{1}{2} J \dot{q}_3^2 + \frac{1}{2} m_2 l^2 \dot{q}_2^2 + \frac{1}{2} I_2 \dot{q}_2^2 + \frac{1}{2} m_1 (q_1^2 \dot{q}_2^2 + \dot{q}_1^2) + \frac{1}{2} I_1 \dot{q}_2^2 - \frac{k}{2} (q_2 - q_3)^2 - m_2 g l \text{ sen } q_2 - m_2 g l - m_1 g q_1 \text{ sen } q_2 - m_1 g q_{1máx}$$

Derivadas parciais e equações de movimento (*Euler-Lagrange*):

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau$$

Corpo 1

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} = \tau_1$$

$$m_1 \ddot{q}_1 - m_1 q_1 \dot{q}_2^2 + m_1 g \sin q_2 = \tau_1$$

Corpo 2

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} = 0$$

$$\ddot{q}_2 (m_2 l^2 + I_2 + I_1 + m_1 q_1^2) + k(q_2 - q_3) + 2m_1 q_1 \dot{q}_2 \dot{q}_1 + (m_2 l + m_1 q_1) g \cos q_2 = 0$$

Corpo 3

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_3} - \frac{\partial L}{\partial q_3} = \tau_3$$

$$J \ddot{q}_3 + k(q_3 - q_2) = \tau_3$$

para simplificação:

$$a = m_2 l$$

$$b = m_2 l^2 + I_2 + I_1$$

3.3 Linearização por realimentação

A proposta básica do método de linearização por realimentação é obter uma lei de controle não linear, representada por uma malha interna de controle, que lineariza exatamente um sistema de caso ideal através de uma mudança de coordenadas no espaço de estado. Este método será aplicado nas equações de movimento do manipulador composto por juntas rígidas de rotação e translação e uma junta flexível de acoplamento por correia. Na seqüência apresentar-se-á o controle a partir das equações do sistema linearizado.

De acordo com as equações de movimento do sistema não linear:

$$m_1 \ddot{q}_1 - m_1 q_1 \dot{q}_2^2 + m_1 g \operatorname{sen} q_2 = \tau_1$$

$$\ddot{q}_2 (m_2 l^2 + I_2 + I_1 + m_1 q_1^2) + k(q_2 - q_3) + 2m_1 q_1 \dot{q}_2 \dot{q}_1 + (m_2 l + m_1 q_1) g \cos q_2 = 0$$

$$J \ddot{q}_3 + k(q_3 - q_2) = \tau_3$$

Conforme as equações do modelo dinâmico, pode-se observar que o sistema é alimentado com duas entradas, aplicando-se uma força τ_1 que acarreta a translação do corpo 1 e um torque τ_3 que ocasiona a rotação do corpo 3, sendo assim caracterizado como um sistema de múltiplas entradas e saídas. Definindo-se as matrizes:

$$M(q) = \begin{bmatrix} M_1 & 0 & 0 \\ 0 & M_2 & 0 \\ 0 & 0 & M_3 \end{bmatrix} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & b + m_1 q_1^2 & 0 \\ 0 & 0 & J \end{bmatrix}$$

$$h(q, \dot{q}) = \begin{Bmatrix} h_1 \\ h_2 \\ h_3 \end{Bmatrix} = \begin{Bmatrix} -m_1 q_1 \dot{q}_2^2 + m_1 g \operatorname{sen} q_2 \\ 2m_1 q_1 \dot{q}_2 \dot{q}_1 + (a + m_1 q_1) g \cos q_2 + k(q_2 - q_3) \\ k(q_3 - q_2) \end{Bmatrix}$$

E adotando como vetor de estado:

$$\xi_1 = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix}$$

$$\xi_2 = \dot{\xi}_1 = \begin{Bmatrix} x_4 \\ x_5 \\ x_6 \end{Bmatrix} = \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{Bmatrix}$$

$$\xi_3 = \dot{\xi}_2 = \begin{Bmatrix} \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{Bmatrix} = \begin{Bmatrix} \frac{\tau_1 - h_1}{M_1} \\ -\frac{h_2}{M_2} \\ \frac{\tau_3 - h_3}{M_3} \end{Bmatrix}$$

Lei de controle:

$$u = M(\xi_1)V + h(\xi_1, \xi_2) = \begin{Bmatrix} \tau_1 \\ 0 \\ \tau_3 \end{Bmatrix}$$

$$\dot{\xi}_2 = -M(\xi_1)^{-1}h(\xi_1, \xi_2) + M(\xi_1)^{-1}u$$

Substituindo as variáveis:

$$\dot{\xi}_2 = \begin{Bmatrix} \frac{x_1 x_5^2 - g \operatorname{sen} x_2}{k(x_3 - x_2) - (a + m_1 x_1)g \cos x_2 - 2m_1 x_1 x_5 x_4} \\ \frac{b + m_1 x_1^2}{k(x_2 - x_3)} \\ \frac{1}{J} \end{Bmatrix} + \begin{bmatrix} \frac{1}{m_1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

$$\dot{\xi}_2 = \begin{bmatrix} \frac{1}{m_1} & 0 & 0 \\ 0 & \frac{1}{b + m_1 x_1^2} & 0 \\ 0 & 0 & \frac{1}{J} \end{bmatrix} \begin{Bmatrix} (g \operatorname{sen} x_2 - x_1 x_5^2)m_1 \\ k(x_2 - x_3) + (a + m_1 x_1)g \cos x_2 + 2m_1 x_1 x_5 x_4 \\ k(x_3 - x_2) \end{Bmatrix} + \begin{bmatrix} \frac{1}{m_1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

$$\text{onde: } M(\xi) = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & \frac{1}{b + m_1 x_1^2} & 0 \\ 0 & 0 & J \end{bmatrix}$$

Então a lei de controle será:

$$u = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & \frac{1}{b + m_1 x_1^2} & 0 \\ 0 & 0 & J \end{bmatrix} \begin{Bmatrix} V_1 \\ V_2 \\ V_3 \end{Bmatrix} + \begin{Bmatrix} (g \operatorname{sen} x_2 - x_1 x_5^2)m_1 \\ k(x_2 - x_3) + (a + m_1 x_1)g \cos x_2 + 2m_1 x_1 x_5 x_4 \\ k(x_3 - x_2) \end{Bmatrix}$$

A forma matricial do modelo não linear:

$$\dot{\xi} = f(x) + g(x)u$$

$$\dot{\xi} = \begin{Bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{Bmatrix} = \begin{Bmatrix} x_4 \\ x_5 \\ x_6 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{Bmatrix} = \begin{Bmatrix} x_4 \\ x_5 \\ x_6 \\ \frac{x_1 x_5^2 - g \operatorname{sen} x_2}{k(x_3 - x_2) - (a + m_1 x_1) g \cos x_2 - 2m_1 x_1 x_5 x_4} \\ \frac{b + m_1 x_1^2}{J} \\ \frac{k(x_2 - x_3)}{J} \end{Bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m_1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

com

$$f(x) = \begin{Bmatrix} x_4 \\ x_5 \\ x_6 \\ \frac{x_1 x_5^2 - g \operatorname{sen} x_2}{k(x_3 - x_2) - (a + m_1 x_1) g \cos x_2 - 2m_1 x_1 x_5 x_4} \\ \frac{b + m_1 x_1^2}{J} \\ \frac{k(x_2 - x_3)}{J} \end{Bmatrix}$$

e

$$g(x) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m_1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J} \end{bmatrix}$$

ou então,

$$\dot{\xi} = f(x) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{\tau_1}{m_1} \\ 0 \\ \frac{\tau_2}{J} \end{bmatrix}$$

Nota-se pela análise de $g(x)$ que o sinal de entrada atua somente nas coordenadas ξ_1 e ξ_3 .

Transformação das coordenadas não lineares

$$\psi_1 = T_1(\xi_1) = \xi_1 = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \end{Bmatrix}$$

$$\psi_2 = T_2(\xi_2) = \psi_1 = \dot{\xi}_1 = \xi_2 = \begin{Bmatrix} x_4 \\ x_5 \\ x_6 \end{Bmatrix} = \begin{Bmatrix} y_4 \\ y_5 \\ y_6 \end{Bmatrix}$$

Nas coordenadas ψ e com a lei de controle u , o sistema torna-se:

$$\dot{\psi}_1 = \dot{\xi}_1 = \xi_2 = \psi_2 = \begin{Bmatrix} x_4 \\ x_5 \\ x_6 \end{Bmatrix} = \begin{Bmatrix} y_4 \\ y_5 \\ y_6 \end{Bmatrix}$$

$$\dot{\psi}_2 = \dot{\xi}_2 = V = \begin{Bmatrix} V_1 \\ V_2 \\ V_3 \end{Bmatrix}$$

Com a nova lei de controle linearizada, podem-se escrever as equações do erro e da seguinte forma:

$$\begin{Bmatrix} erro_1 \\ erro_2 \\ erro_3 \end{Bmatrix} = \begin{Bmatrix} \ddot{e}_1 + k_2 \dot{e}_1 + k_1 e_1 \\ \ddot{e}_2 + k_4 \dot{e}_2 + k_3 e_2 \\ \ddot{e}_3 + k_6 \dot{e}_3 + k_5 e_3 \end{Bmatrix}$$

Para se obter um sistema estável, os pólos da equação do erro devem estar no semi-plano esquerdo. Adotando-se todos os pólos na posição -10 , o que assegura uma boa convergência do erro, chega-se a:

$$(s+10)^2 = \ddot{e}_1 + k_2 \dot{e}_1 + k_1 e_1 = s^2 \ddot{e}_1 + k_2 s \dot{e}_1 + k_1 e_1 = e_1 (s^2 + k_2 s + k_1)$$

$$(s+10)^2 = \ddot{e}_2 + k_4 \dot{e}_2 + k_3 e_2 = s^2 \ddot{e}_2 + k_4 s \dot{e}_2 + k_3 e_2 = e_2 (s^2 + k_4 s + k_3)$$

$$(s+10)^2 = \ddot{e}_3 + k_6 \dot{e}_3 + k_5 e_3 = s^2 \ddot{e}_3 + k_6 s \dot{e}_3 + k_5 e_3 = e_3 (s^2 + k_6 s + k_5)$$

$$s^2 + 20s + 100s = s^2 + k_2 s + k_1$$

$$s^2 + 20s + 100s = s^2 + k_4 s + k_3$$

$$s^2 + 20s + 100s = s^2 + k_6 s + k_5$$

onde denomina-se as constantes:

$$k_1 = 100$$

$$k_2 = 20$$

$$k_3 = 100$$

$$k_4 = 20$$

$$k_5 = 100$$

$$k_6 = 20$$

Assim, obtém-se a nova lei de controle V :

$$V = \begin{Bmatrix} V_1 \\ V_2 \\ V_3 \end{Bmatrix} = \begin{Bmatrix} \dot{y}_4^d - k_1(y_1 - y_1^d) - k_2(y_4 - y_4^d) \\ \dot{y}_5^d - k_3(y_2 - y_2^d) - k_4(y_5 - y_5^d) \\ \dot{y}_6^d - k_5(y_3 - y_3^d) - k_6(y_6 - y_6^d) \end{Bmatrix}$$

Capítulo 4

Simulação

4.1 Metodologia aplicada

Nesta etapa, são feitas as análises referentes ao desempenho do sistema robótico através de gráficos correspondentes à identificação do sistema por redes neurais e à linearização por realimentação. Para isso são executadas as simulações, através da ferramenta Simulink do ambiente MATLAB®, comparando-se o comportamento do robô submetido à alteração de seus parâmetros e sinais externos, buscando-se assim a observação e análise dos resultados obtidos.

4.1.1 Representação do sistema

Estão sendo considerados alguns parâmetros construtivos do mecanismo proposto neste trabalho, ou seja, são dados estimados somente para representar o sistema. Estes dados são tomados como padrão em seu desenvolvimento, demonstrando-se também que pequenas mudanças nestes parâmetros podem alterar a trajetória obtida pelo manipulador. Os valores iniciais são:

- $m_1 = 1$ kg (massa do corpo1);
- $m_2 = 1$ kg (massa do corpo2);
- $m_3 = 1$ kg (massa do corpo3);
- $g = 9.8$ m/s² (aceleração da gravidade);
- $k = 1$ N/m (constante de rigidez da correia);
- $\tau_1 = 0$ (força aplicada ao corpo 1)

- $\tau_3 = 0.2 \text{ Nm}$ (torque aplicada ao corpo 3)
- $I_1 = 1 \text{ m}^4$ (momento de inércia do corpo 1);
- $I_2 = 1 \text{ m}^4$ (momento de inércia do corpo 2);
- $I_3 = 1 \text{ m}^4$ (momento de inércia do corpo 3);

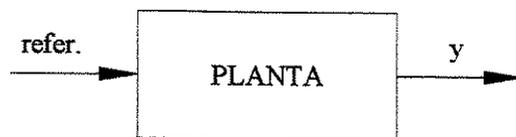


Figura. 4.1 – Representação do modelo não linear

O modelo esquemático acima representa o sistema não linear do manipulador onde o sinal de entrada é composto por um deslocamento ($d = 1\text{m}$), velocidade linear ($v_l = 2 \text{ m/s}$), ângulo do corpo 2 ($a_2 = 10 \text{ rad}$), velocidade angular do corpo 2 ($va_2 = 0 \text{ rad/s}$), ângulo do corpo 3 ($a_3 = 20 \text{ rad}$), velocidade angular do corpo 3 ($va_3 = 0 \text{ rad/s}$) e a excitação de rotação no corpo 3 (τ_3) e translação no corpo 1 (τ_1), sendo estes os parâmetros iniciais do modelo. Sua saída y é representada graficamente no Item 4.2.1 sobre a identificação do sistema por redes neurais.

4.2 Redes neurais em identificação

A identificação consiste na obtenção de uma representação matemática para o processo, sem fazer uso das leis fundamentais que regem seu comportamento dinâmico. Para a identificação são usados pares de dados de entrada e saída (u, y), e um algoritmo para ajuste de um modelo matemático previamente especificado. A Figura 4.2 ilustra o procedimento, onde \hat{y} é o vetor de saída do modelo identificado.

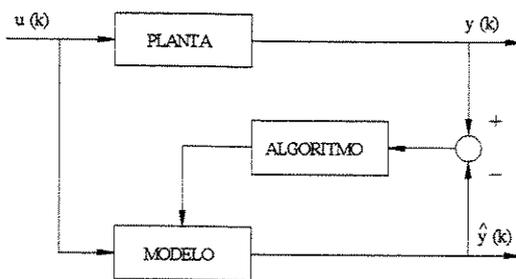


Figura 4.2 – Procedimento de identificação

Na identificação por redes neurais, o modelo dinâmico do processo pode ser representado (Figura 4.3) como um neuro-identificador para um processo de primeira ordem.

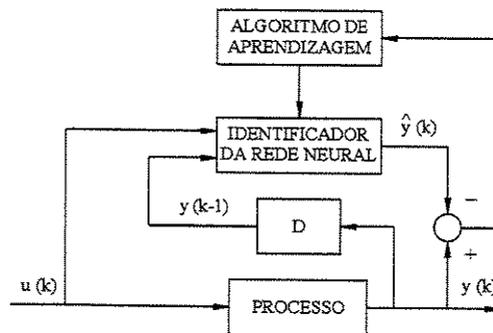


Figura 4.3 – Modelo para identificação com redes neurais artificiais

4.2.1 Identificação do sistema por redes neurais

Para a identificação do sistema, está sendo aplicada uma rede neural de múltiplas camadas, onde seu sinal de entrada é o mesmo da planta e sua saída tende a se aproximar da saída da planta, onde o erro obtido pela diferença das duas saídas gera uma realimentação à rede proporcionando o seu treinamento, buscando-se assim, minimizar este erro. Ao se reproduzir o sistema robótico e sua identificação por redes neurais optou-se pela utilização de algumas funções do “Toolbox” de redes neurais do ambiente MATLAB®, ou parte delas, tais como: INITFF – cria valores iniciais para os pesos e bias de uma rede *feed-forward* para 2 camadas; TRAINLM – treinamento da rede *feed-forward* utilizando o método Levenberg-Marquardt; SIMUFF – simulação da rede *feed-forward* para 2 camadas; TANSIG – função de ativação não linear tangente hiperbólica na camada intermediária; PURELIN – função de ativação linear na camada de saída. Foi utilizado este método pois satisfaz as necessidades do projeto alcançando os resultados esperados.

Para a identificação do sistema não linear estão sendo utilizadas redes neurais de múltiplas camadas, onde neste caso está sendo aplicada uma rede de duas camadas, uma intermediária com oito neurônios e uma última camada composta de seis saídas. O estado inicial é obtido através de várias combinações dos dados de entrada de deslocamento, ângulos, velocidades e forças dos

motores. No treinamento da rede é utilizado o algoritmo Levenberg-Marquardt buscando-se uma somatória do erro quadrático na saída desejada (*target*) de 0.005 em amplitude.

4.2.2 Aplicação

A informação para identificação de sistemas dinâmicos pode ser obtida a partir da observação do comportamento de entrada e saída do processo. Utilizando-se uma rede neural, esta recebe a mesma entrada que o processo e a saída desejada para a rede é a saída do processo.

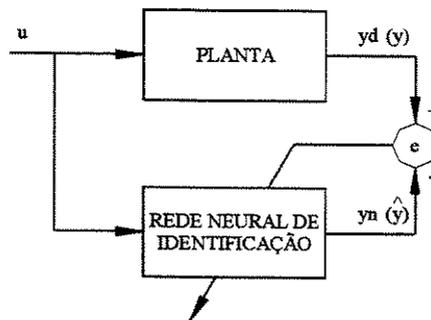


Figura 4.4 – Identificação de sistemas por redes neurais

Para se testar a rede de identificação do movimento robótico, são tomadas as medidas de posição e velocidade dos três corpos do sistema. Com isto, é feita a simulação da rede, onde pode-se analisar a comparação entre os sinais obtidos da rede e o sinal “real” do sistema, proveniente da simulação que representa o comportamento livre do corpos. Nota-se que houve uma boa aproximação da rede em relação a este sinal real.

Os gráficos abaixo representam a identificação do sistema por redes neurais do movimento do manipulador com a aplicação de uma força de translação no corpo 1 e um torque ao corpo 3.

Dados iniciais: $d = 1\text{m}$; $v_1 = 2\text{m/s}$; $a_2 = 10\text{ rad}$; $va_2 = 0\text{ rad/s}$; $a_3 = 20\text{ rad}$; $va_3 = 0\text{ rad/s}$.

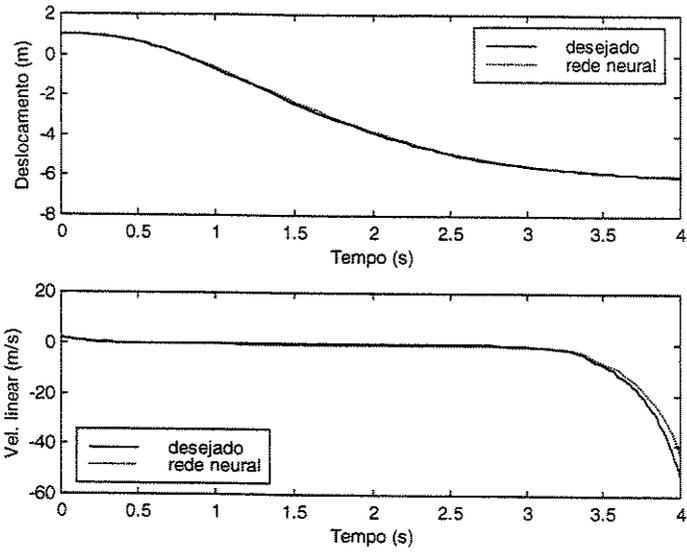


Figura 4.5 – Deslocamento e velocidade do corpo 1

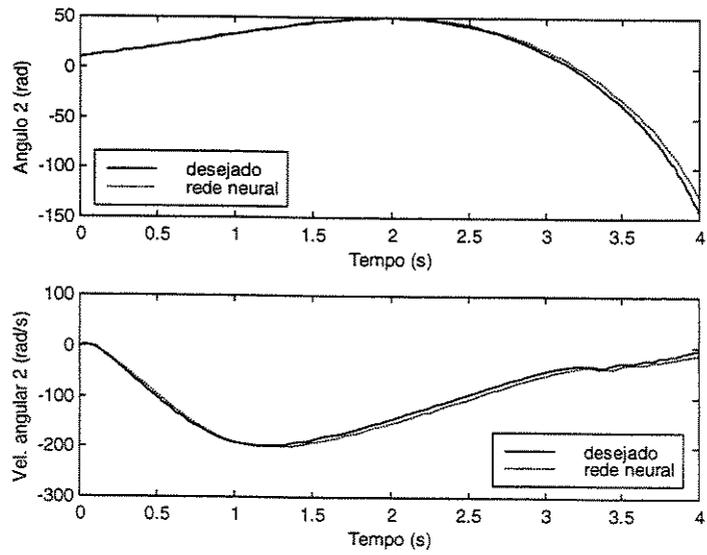


Figura 4.6 – Ângulo e velocidade angular do corpo2

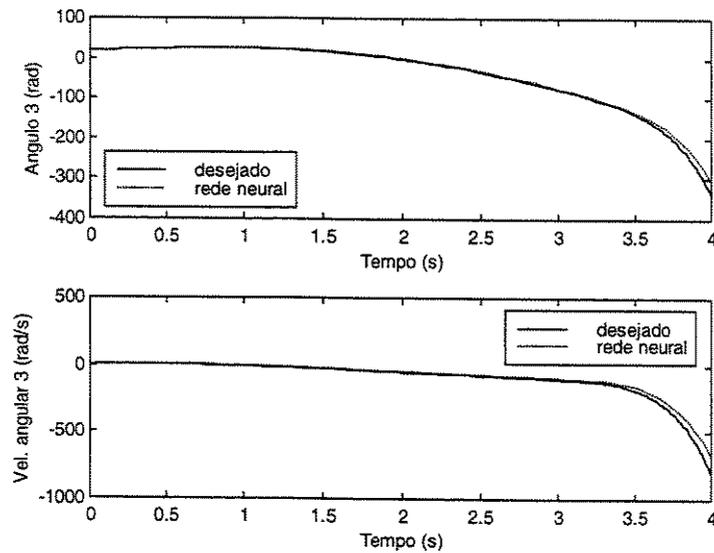


Figura 4.7 – Ângulo e velocidade angular do corpo3

Entende-se como desejado, o sinal representando o desempenho do sistema (linha contínua), onde uma rede neural (linha tracejada) busca acompanhar sua trajetória dentro de limites aceitáveis propostos pelo erro objetivo em seu treinamento. Com os parâmetros estabelecidos, a rede neural obtém êxito nesta aplicação, pois consegue rastrear o sinal original em uma tolerância de aproximação conforme especificado.

Todos os corpos sofrem influência dos movimentos gerados pelo sistema. Convém salientar que os sinais obtidos nas Figuras 4.5, 4.6 e 4.7 apenas representam o papel desempenhado pelos corpos após uma certa excitação, não havendo qualquer tipo de controle que restrinja sua trajetória, ou seja, são aplicadas as forças e espera-se as respostas da posição e velocidade para cada corpo.

4.3 Análise da mudança de parâmetros

As características do sistema influenciam diretamente o desempenho do mecanismo fazendo com que as alterações dos seus parâmetros modifiquem visivelmente sua posição e respectiva velocidade. Sendo assim, propõe-se algumas mudanças de parâmetros no sentido de

demonstrar os desvios causados à trajetória, propositalmente, podendo-se com isto comparar os sinais obtidos à performance esperada.

Cada um dos três corpos do sistema analisado podem ser caracterizados por desempenhos distintos, sendo que qualquer alteração dos parâmetros de massa, inércia dos corpos, dimensões e constantes como a rigidez da correia podem influenciar no desempenho deste sistema, propondo-se assim, fazer uma comparação com a alteração da constante de rigidez para que possa ser observado sua importância no modelo.

A alteração da constante de rigidez considerada é proposta com os valores de $k = 0.001$, ou seja um fator de elasticidade simulando uma mola bastante flexível, $k = 0.5$ um valor intermediário, e $k = 1$ sendo considerada uma mola menos flexível.

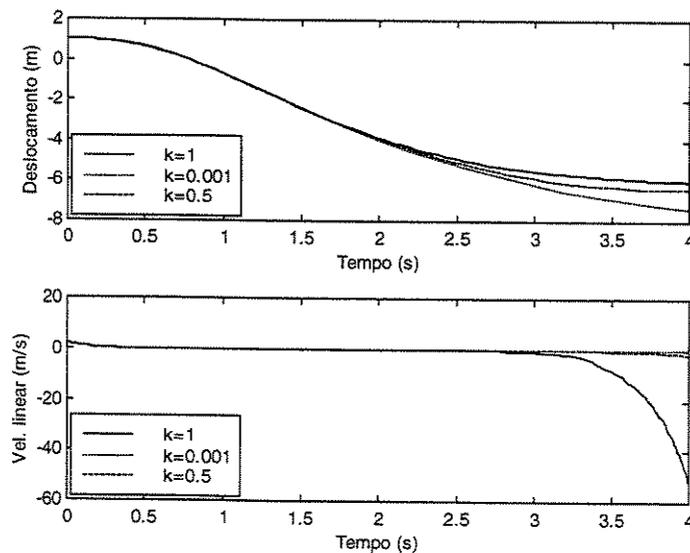


Figura 4.8 – Deslocamento e velocidade linear do corpo 1

Na Figura 4.8 que representa o deslocamento do corpo 1 e sua respectiva velocidade linear, pode-se notar que quanto maior a constante de rigidez, menor será seu deslocamento final, mas perde-se a estabilidade em sua velocidade, havendo portanto um ponto ótimo entre os dois extremos.

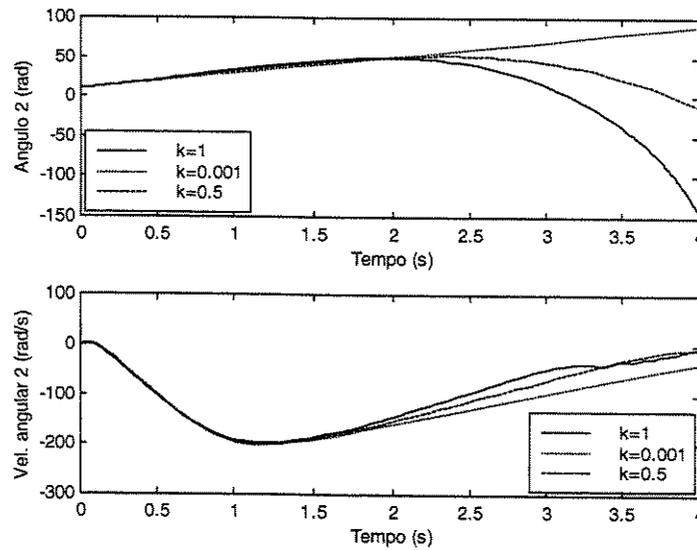


Figura 4.9 – Ângulo e velocidade angular do corpo2

O ângulo e velocidade angular do corpo 2 representado pela Figura 4.9 demonstra a comparação de rigidez, notando-se que diminuindo a rigidez, o ângulo tende a ser uma função linear, fazendo com que sua velocidade angular se estabilize num menor tempo.

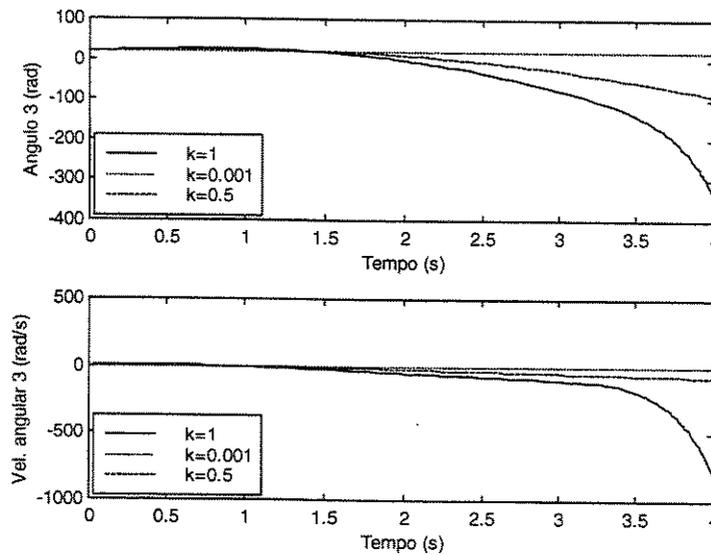


Figura 4.10 – Ângulo e velocidade angular do corpo3

Verifica-se no ângulo e velocidade do corpo 3, na Figura 4.10, que quanto mais rígida a correia, maior será a divergência de seu sinal, fazendo com que direcione-se ao infinito, e assim mostrando apenas a influência da mudança da constante de rigidez no sistema sem alteração em qualquer outro parâmetro. Com isto, pode-se chegar a um valor ideal da constante, visando obter um desempenho desejado

4.4 Linearização por realimentação em tempo contínuo

A modelagem e simulação da linearização por realimentação para o sistema do manipulador é aplicada através da ferramenta interativa Simulink, para análise do sistema dinâmico. Este ambiente proporciona a construção gráfica em forma de diagrama de blocos podendo-se representar o sistema avaliar o seu desempenho. Com a integração ao MATLAB®, o Simulink proporciona acesso imediato a uma variação dos parâmetros de análise e ferramentas de projeto podendo-se facilmente obter os resultados do sistema estudado.

4.4.1 Simulação do sistema com excitação por ondas quadrada e senoidal

Os gráficos abaixo representam a simulação de movimento do manipulador com a aplicação de uma força de translação no corpo 1 representada por uma onda quadrada, e a aplicação de um torque no corpo 3 representada por uma onda senoidal. O diagrama de blocos da Figura 4.11 mostra o esquema da simulação utilizada.

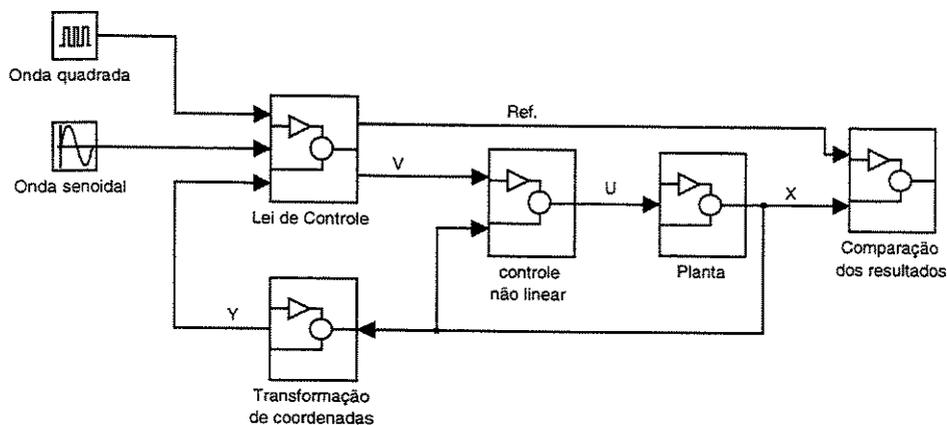


Figura 4.11 – Diagrama de blocos da simulação com ondas quadrada e senoidal em malha fechada

Cada componente do diagrama de blocos da simulação é composto de um sub-sistema que representa sua respectiva função dentro do sistema global, e dentre estes são apresentadas as funções que compõem os parâmetros do problema que é a planta e o controle não linear, onde os mesmos serão substituídos por redes neurais nos próximos capítulos.

A planta é composta pelos parâmetros do sistema robótico e suas variáveis, onde podem ser observados cada constante contida na forma matricial do modelo não linear, tendo os sinais do controle não linear como sua entrada.

Tendo a função não linear: $\dot{\xi} = f(x) + g(x)u$

e a forma matricial do modelo:

$$\dot{\xi} = \begin{Bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{Bmatrix} = \begin{Bmatrix} x_4 \\ x_5 \\ x_6 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{Bmatrix} = \begin{Bmatrix} x_4 \\ x_5 \\ x_6 \\ \frac{x_1 x_5^2 - g \sin x_2}{k(x_3 - x_2) - (a + m_1 x_1)g \cos x_2 - 2m_1 x_1 x_5 x_4} \\ \frac{b + m_1 x_1^2}{J} \\ \frac{k(x_2 - x_3)}{J} \end{Bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m_1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

pode-se representar a função pelo diagrama da Figura 4.12.

O controle não linear também é composto basicamente dos mesmos parâmetros contidos na planta, mas são expressos de forma diferente de acordo com a técnica de linearização por realimentação, onde os sinais da lei de controle estão considerados na entrada deste subsistema, observando-se assim sua forma matricial:

$$u = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & J \end{bmatrix} \begin{Bmatrix} V_1 \\ V_2 \\ V_3 \end{Bmatrix} + \begin{Bmatrix} (g \sin x_2 - x_1 x_5^2) m_1 \\ k(x_2 - x_3) + (a + m_1 x_1)g \cos x_2 + 2m_1 x_1 x_5 x_4 \\ k(x_3 - x_2) \end{Bmatrix}$$

que é representada pelo diagrama de blocos da Figura 4.15.

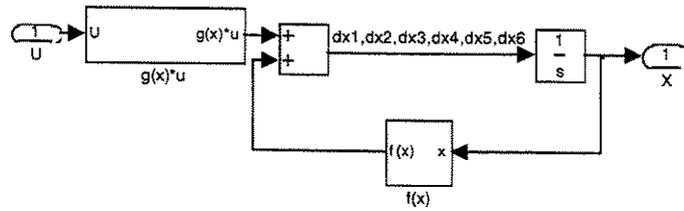


Figura 4.12 – Diagrama de blocos do Simulink que representa a planta.

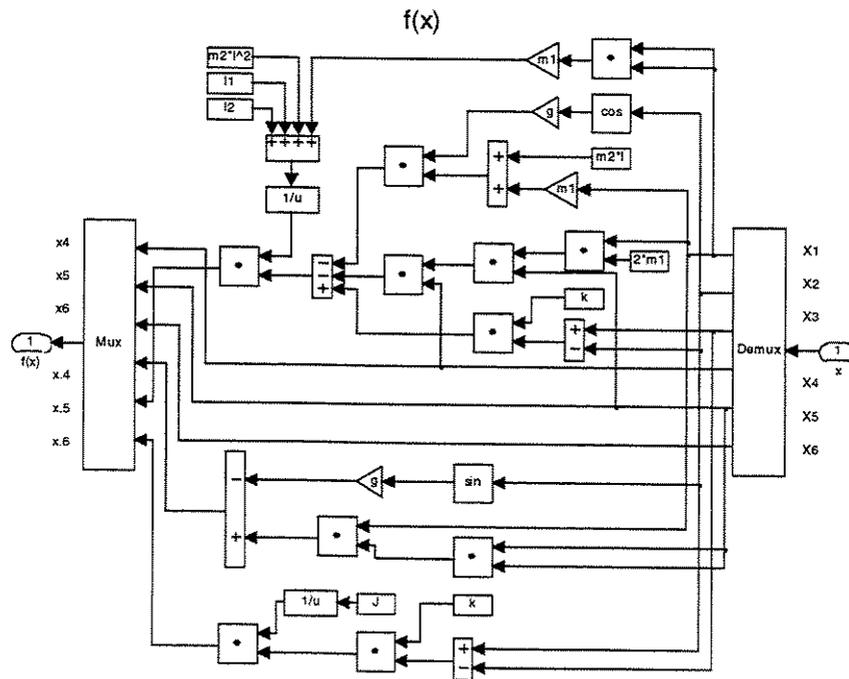


Figura 4.13 – Diagrama de blocos do Simulink que representa $f(x)$.

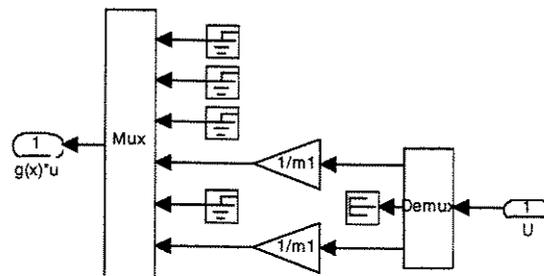


Figura 4.14 – Diagrama de blocos do Simulink que representa $g(x)*u$.

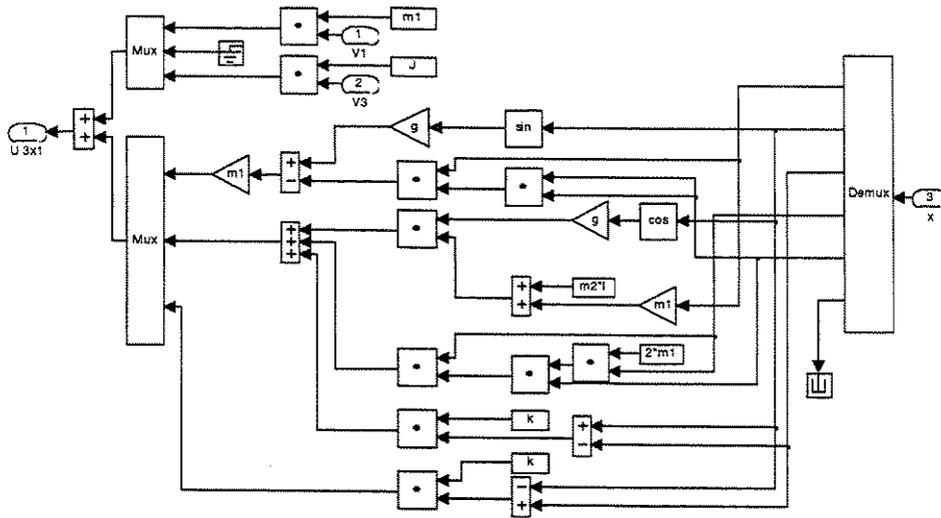


Figura 4.15 – Diagrama de blocos do Simulink que representa o controle não linear.

O resultado obtido para uma saída desejada, para o deslocamento do corpo1, em forma de onda quadrada segue os parâmetros:

Período: $T = 2$ seg;

Proporção do período do ciclo: 50 %;

Amplitude: $A = 1$;

Tempo de início: $t_0 = 0$ seg.

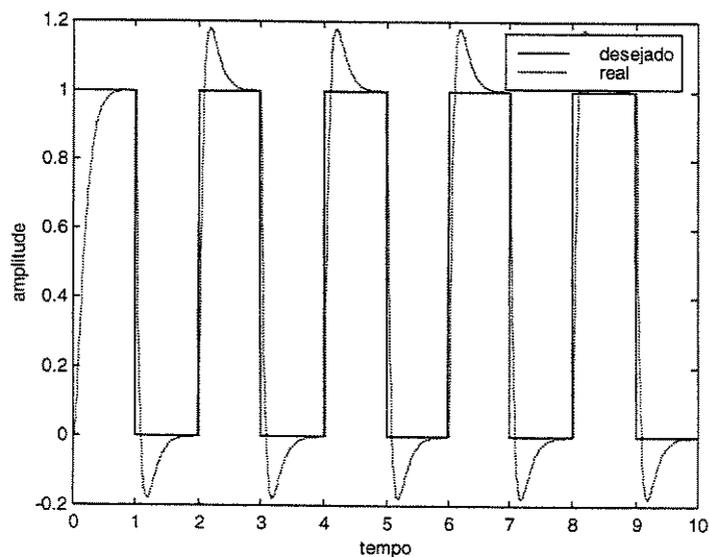


Figura 4.16 – Posição do corpo1

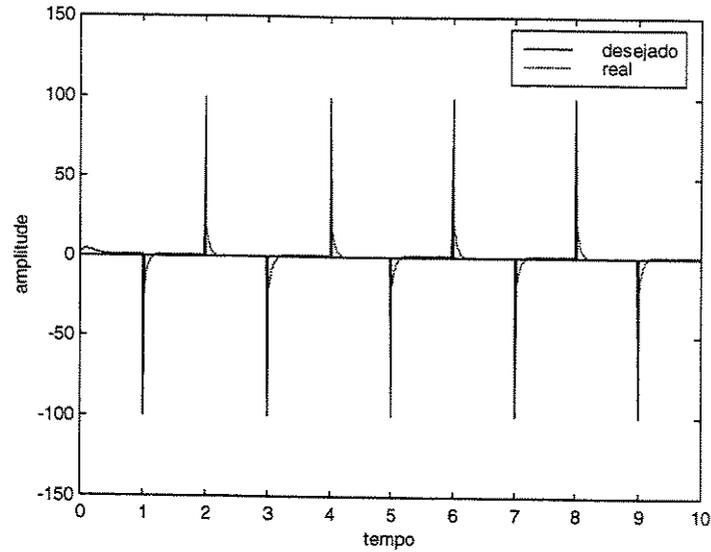


Figura 4.17 – Velocidade linear do corpo1

Nota-se que o controle obtido pelo método de linearização por realimentação apresenta um resultado satisfatório ao acompanhar o sinal da onda quadrada, mesmo apresentando um sobressinal (*overshoot*) significativo, podendo ser observado nas Figuras 4.16 e 4.17.

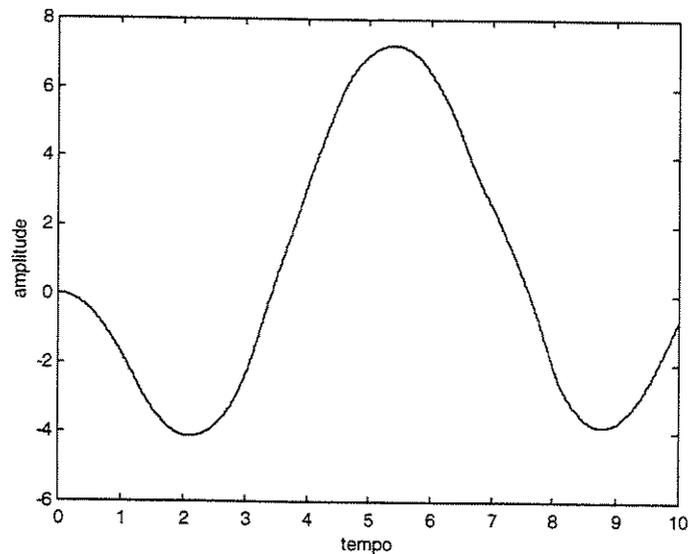


Figura 4.18 – Ângulo do corpo2

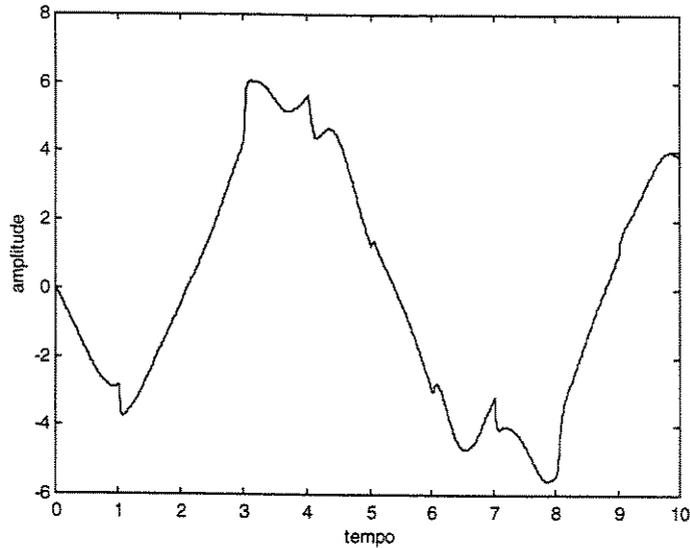


Figura 4.19 – Velocidade angular do corpo2

As Figuras 4.18 e 4.19 representam somente os comportamentos de ângulo e velocidade angular do corpo 2, pois são apenas resultantes dos outros dois movimentos, já que não há controle aplicado diretamente neste corpo.

Outro teste foi realizado, utilizando como função objetivo uma saída senoidal no deslocamento angular do corpo 3 (Figuras 4.20 e 4.21), seguindo os parâmetros da onda senoidal:

Amplitude: $A = 1$;

Frequência: $\omega = 4$ rad/seg;

Fase: 0 rad;

Tempo de início: $t = 0$ seg.

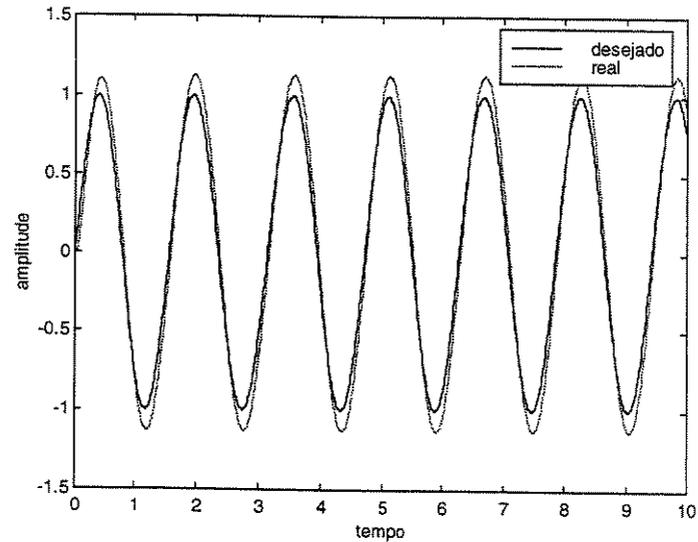


Figura 4.20 – Ângulo do corpo3

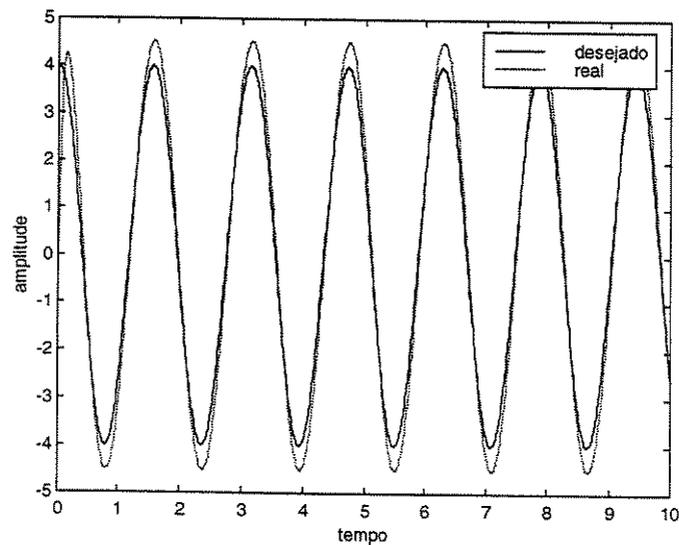


Figura 4.21 – Velocidade angular do corpo3

Sendo neste caso a onda senoidal relativamente mais fácil de se controlar que a onda quadrada, devido sua continuidade na trajetória, ainda assim observa-se uma pequena diferença, ou sobressinal, do sinal obtido (real) para o sinal original (desejado), mas pelas Figuras 4.20 e 4.21 de posição e velocidade angular do corpo 3, respectivamente, pode-se observar que o controlador satisfaz o objetivo de acompanhar a trajetória percorrida em seu desenvolvimento.

4.4.2 Simulação do sistema com resposta ao degrau

Os gráficos abaixo representam a simulação de movimento do manipulador com a aplicação de uma força de translação no corpo 1 representada por um degrau unitário no instante 2 s, e a aplicação de um torque no corpo 3 representada por um degrau unitário no instante 1 s, conforme demonstrado pelo diagrama de blocos da Figura 4.22.

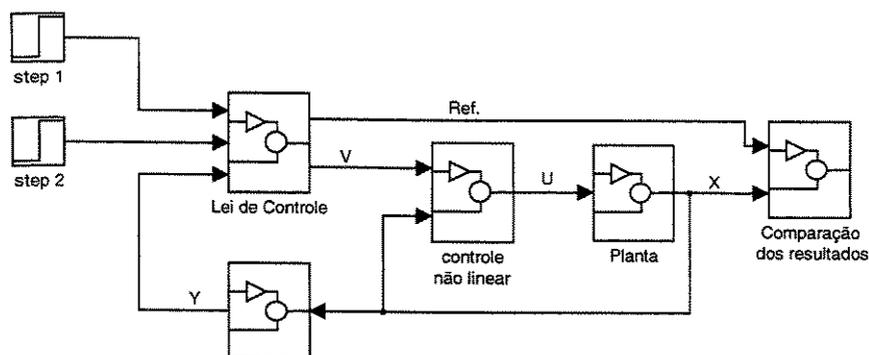


Figura 4.22 – Diagrama de blocos da simulação com resposta ao degrau.

Parâmetros do step 1:

Tempo do step: $t_s = 2$ seg;

Valor inicial: 0;

Valor final: 1;

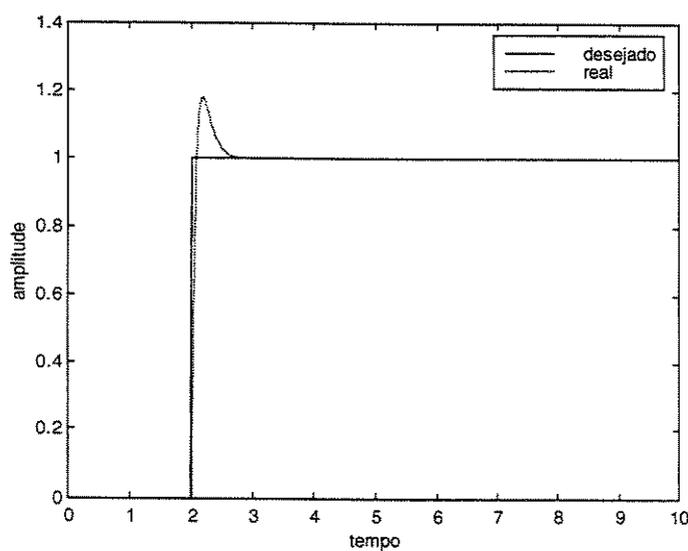


Figura 4.23 – Posição do corpo 1

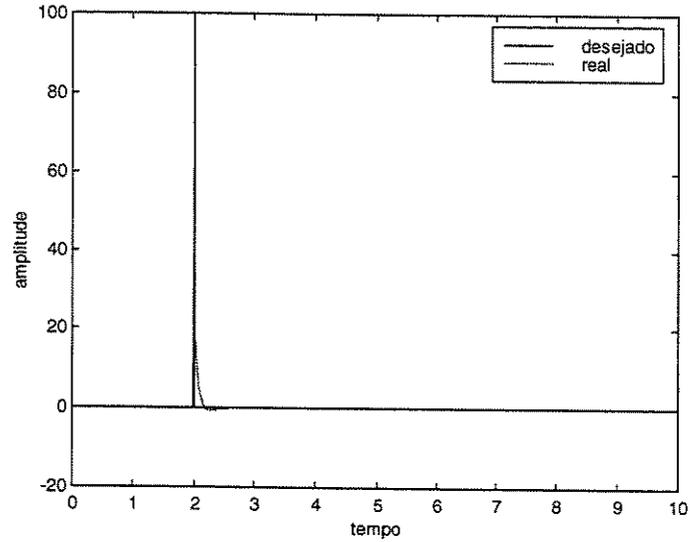


Figura 4.24 – Velocidade linear do corpo1

As Figuras 4.23 e 4.24 representam o comportamento de posição e velocidade linear do corpo 1, onde pode ser notado seu nível de aproximação ao sinal de degrau unitário desejado com um sobressinal razoável.

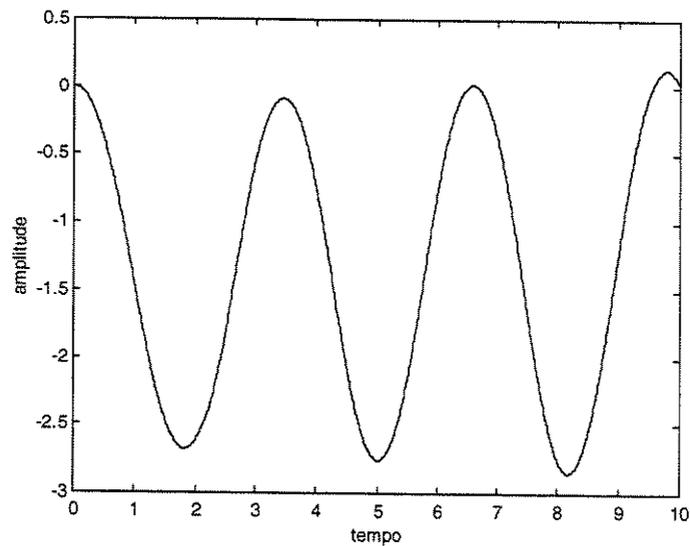


Figura 4.25 – Ângulo do corpo2

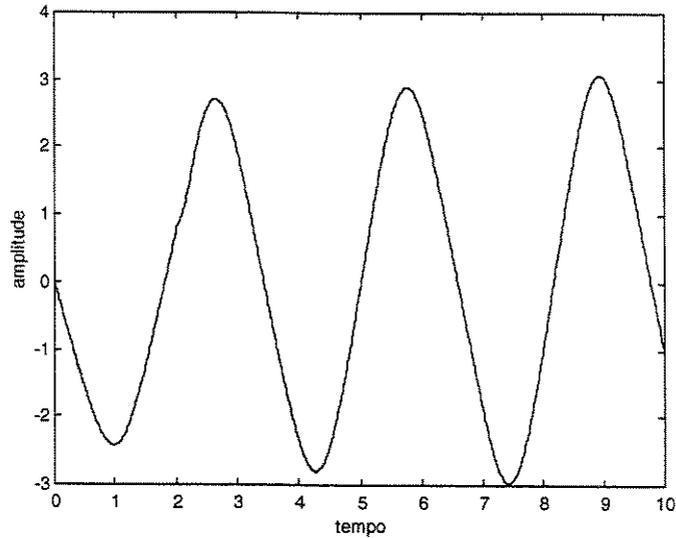


Figura 4.26 – Velocidade angular do corpo2

As Figuras 4.25 e 4.26 representam o comportamento da posição e velocidade angular do corpo 2, já que seu movimento é diretamente influenciado pelas rotação e translação dos corpos 3 e 1, respectivamente, não havendo excitação motora direta, ou seja, a rotação do corpo 2 varia em função dos outros dois corpos, sendo impossível movê-lo separadamente.

Parâmetros do step 2:

Tempo do step: $t_s = 1$ seg;

Valor inicial: 0;

Valor final: 1;

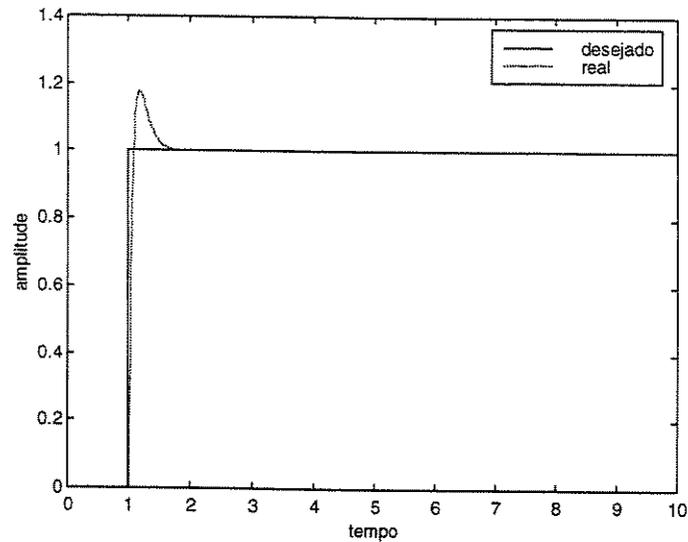


Figura 4.27 – Ângulo do corpo3

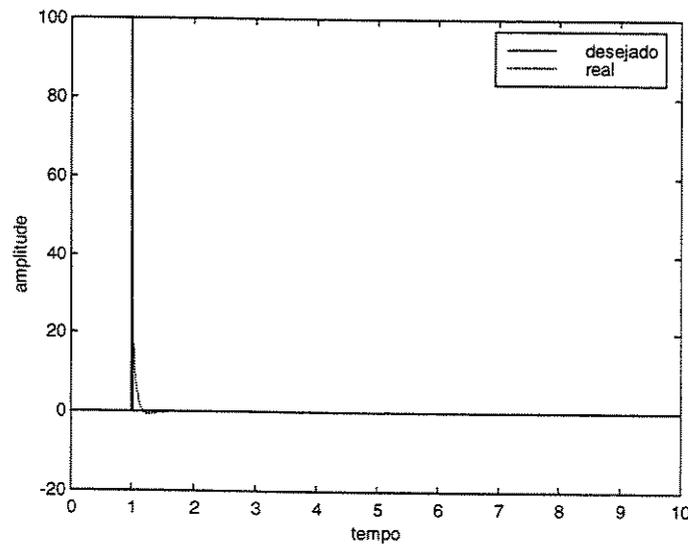


Figura 4.28 – Velocidade angular do corpo3

Da mesma forma observada no corpo 1, pode-se notar pelas Figuras 4.27 e 4.28 que o método de linearização por realimentação é satisfatório como controle ao sistema não linear proposto neste trabalho, pois seu desempenho de aproximação ao sinal do degrau unitário aplicado ao corpo 3, cumpre a proposta de linearização e controle do sistema.

4.4.3 Simulação do sistema em malha aberta

A representação do sistema em malha aberta é para demonstrar a influência da realimentação de estado e a importância de sua presença no desempenho de controle aplicado a planta, conforme pode ser observado em seu esquema na Figura 4.29, propondo uma comparação ao sistema em malha fechada. Entende-se como malha aberta a ausência da realimentação de estado ao controle não linear, ou seja, a malha interna de linearização.

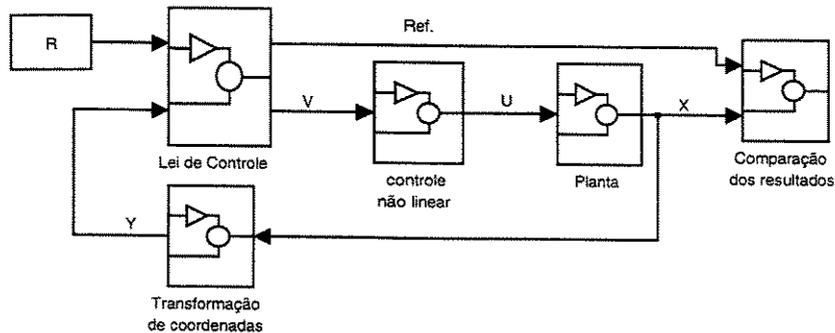


Figura 4.29 – Diagrama de blocos da simulação em malha aberta

A seguir, são apresentados os gráficos da posição do corpo 1 e o comportamento angular do corpo 3 em um sistema sem realimentação de estado no controlador não linear.

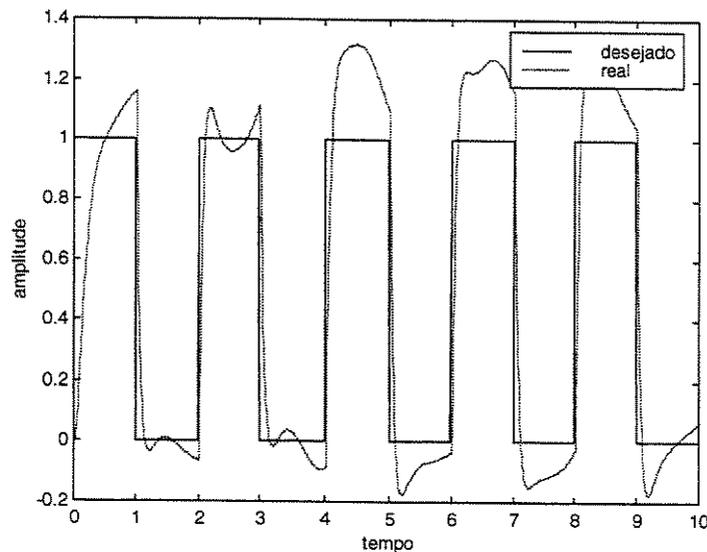


Figura 4.30 – Posição do corpo1 sem realimentação no controle

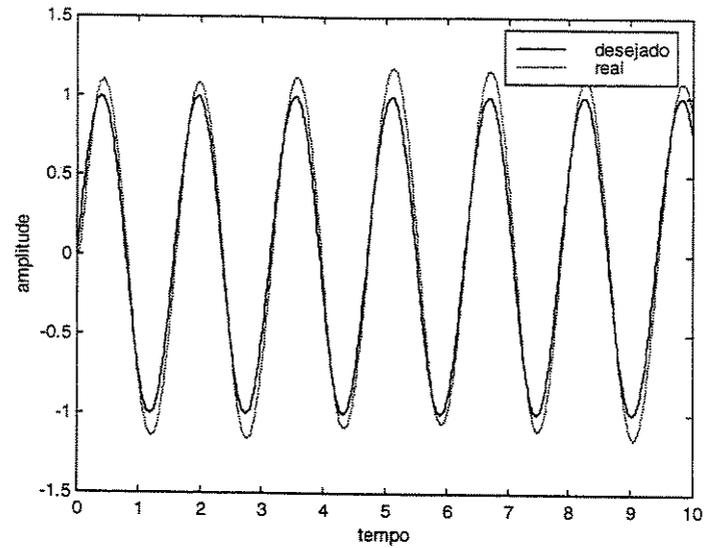


Figura 4.31 – Ângulo do corpo3 sem realimentação no controle

As Figuras 4.30 e 4.31 representam o comportamento dos corpos 1 e 3 com excitação por ondas quadrada e senoidal, respectivamente, como um sistema em malha aberta.

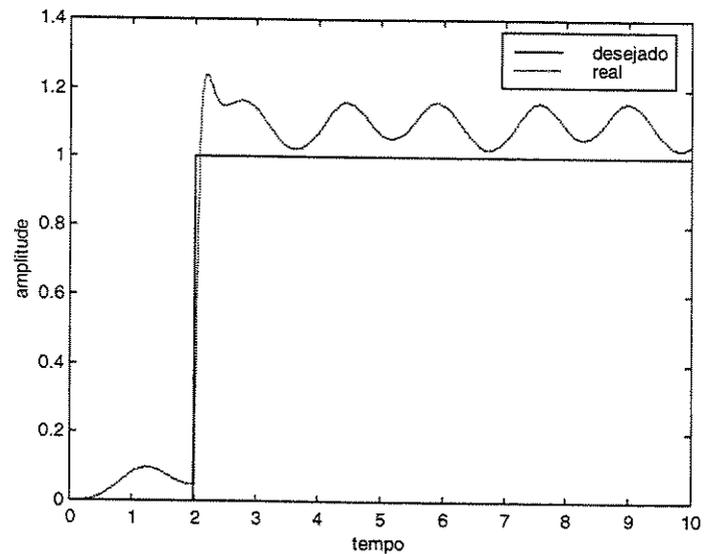


Figura 4.32 – Posição do corpo1 sem realimentação no controle

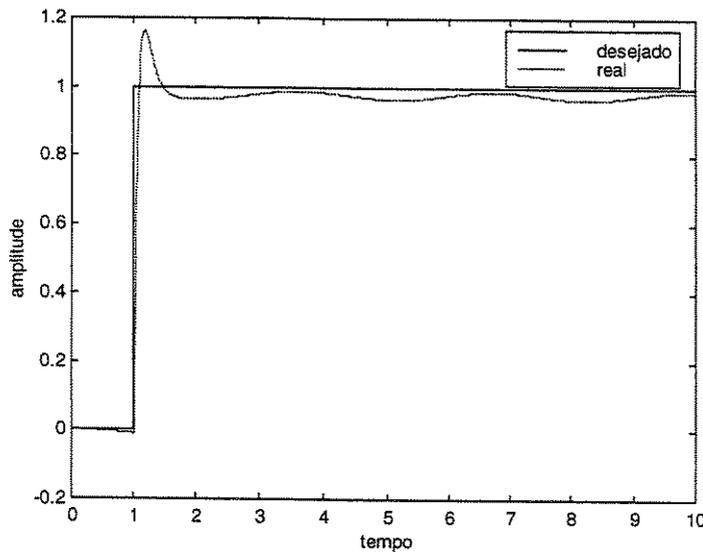


Figura 4.33 – Ângulo do corpo3 sem realimentação no controle

As Figuras 4.32 e 4.33 representam o comportamento dos corpos 1 e 3, respectivamente, com resposta ao degrau como um sistema em malha aberta.

Pode-se notar nas Figuras 4.32 e 4.33 que o sistema sem a realimentação de estado ainda consegue acompanhar a trajetória nos movimentos, mas com um desempenho de menor eficiência comparado ao caso com realimentação, mostrado nas Figuras 4.23 e 4.27.

4.5 Comparação dos polos estimados

Dentre todas as características estudadas neste sistema, propõe-se analisar a variação de posição dos polos em relação a seu tempo de estabilização e do sobressinal (*overshoot*), observando e comparando seu desempenho na mudança e determinação dos ganhos obtidos pelo polinômio característico do sistema .

As Figuras 4.34 e 4.35 representam um deslocamento dos pólos ainda mais para o lado esquerdo do semi plano, ou seja $(s+20)^2$, onde pode-se observar um sobressinal maior, porém nota-se que o tempo de estabilização é menor em relação ao utilizado no sistema $(s+10)^2$.

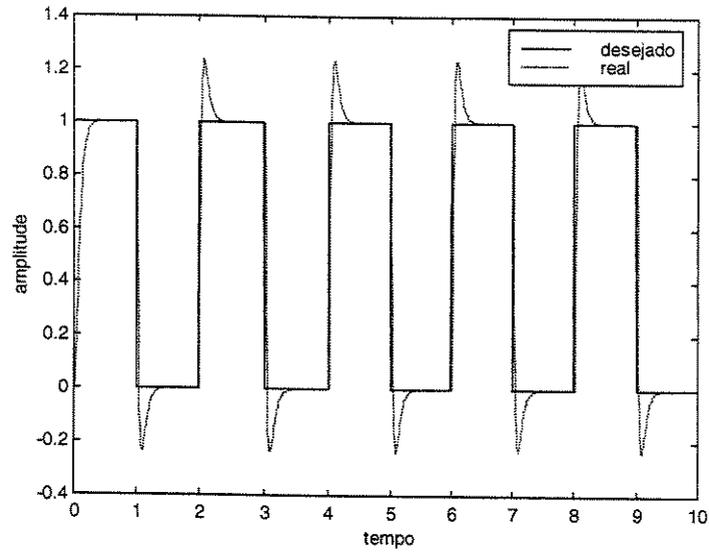


Figura 4.34 – Posição do corpo 1 com sinal de onda quadrada

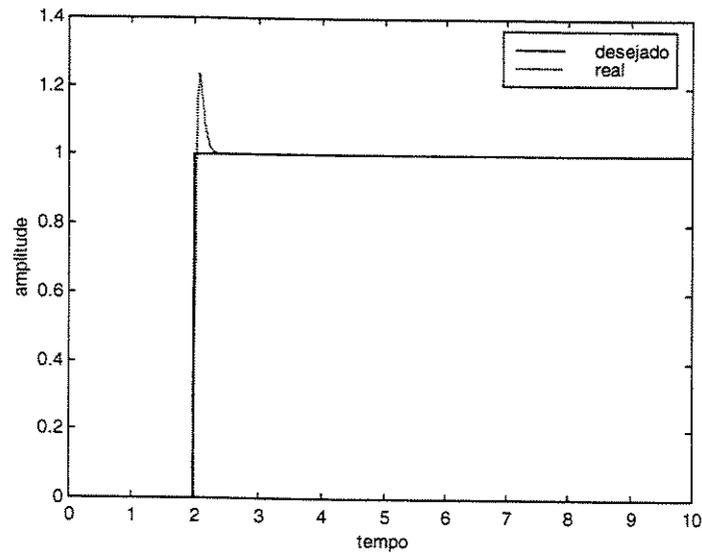


Figura 4.35 – Posição do corpo 1 com resposta ao degrau

Já em pólos localizados no semi-plano direito, como as raízes de $(s-1)^2$, nota-se que sua resposta não converge ao sinal desejado, todavia vale ressaltar que este resultado é somente representativo, sendo portanto descartado sua aplicação.

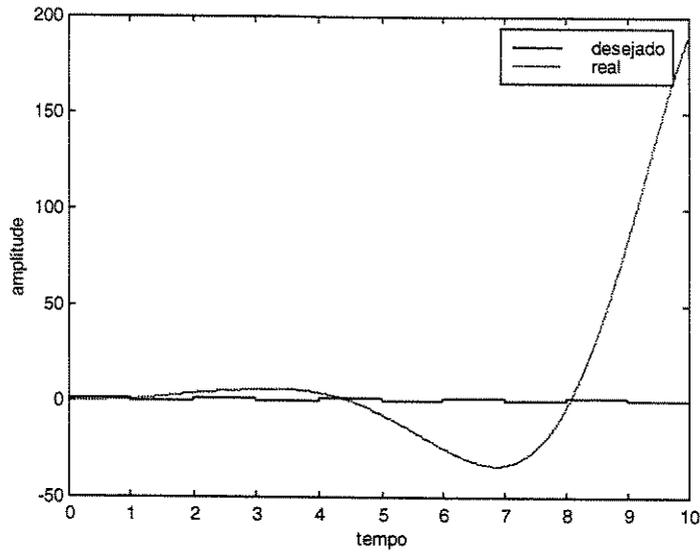


Figura 4.36 – Posição do corpo 1 com sinal de onda quadrada

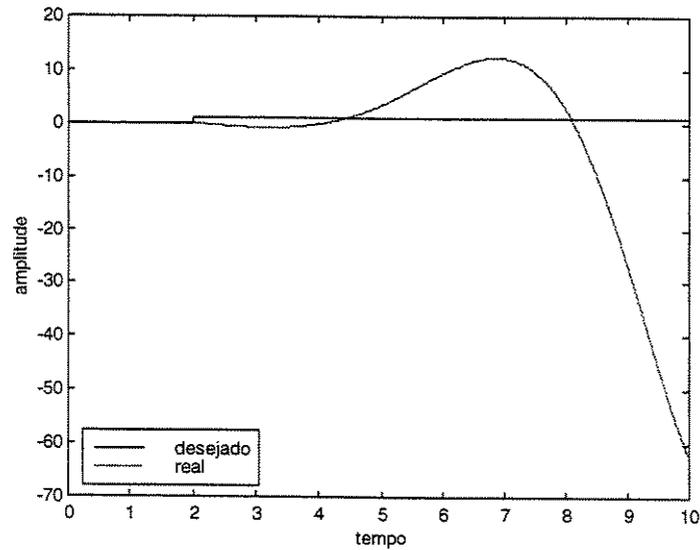


Figura 4.37 – Posição do corpo 1 com resposta ao degrau

Pode-se dizer que há um ponto ótimo na estimação dos pólos, mas neste trabalho busca-se apenas a utilização de qualquer opção satisfatória dentro de uma tolerância razoável para a representação do sistema.

Capítulo 5

Aplicação de redes neurais em linearização

5.1 Estrutura de uma rede neural

Basicamente, a rede neural é composta de múltiplas camadas para atuar como uma função de aproximação de um mapeamento não linear, relacionando entradas e saídas da rede. A estrutura da rede neural é estabelecida através de tentativa e erro. Não existe até o momento, dado o conhecimento do autor, um método que permita o projeto de uma estrutura de rede a partir das condições operacionais e do desempenho desejado para um dado sistema.

Encontram-se vários problemas para se definir estruturas de redes neurais, buscando-se conseguir um melhor desempenho com o mínimo de tempo necessário para a aproximação da função. Entre os diversos aspectos a serem considerados, alguns podem ser citados:

- Número de amostras na entrada da rede pode ser inadequada para se obter o comportamento real da função;
- A rede neural pode ser subdimensionada em elementos nas camadas intermediárias;
- Pode haver um superdimensionamento desnecessário em seus parâmetros prejudicando e aumentando o tempo de operação.

Não há uma regra básica para se montar uma estrutura para solucionar cada problema, e a rede pode ser estruturada de forma diferente a cada pequena modificação na planta. Nota-se que é necessário treinar inúmeras vezes uma mesma arquitetura de rede, pois encontram-se diferentes

resultados a cada reinicialização do programa. Além disso, não há garantia de que um número de elementos seja suficientemente adequado para representar a estrutura do sistema.

Como não há um método específico, faz-se inúmeras tentativas e mudanças na estrutura da rede em relação ao número de camadas intermediárias, número de amostras, número de neurônios na camada intermediária, variação do momento e taxa de aprendizagem. Quando os dados são abusivos em quantidade ou tamanho, o tempo torna-se um fator considerado, pois depende-se de sua rapidez para tornar-se um método eficiente e que compense sua utilização. Em suma, deve-se usar o próprio discernimento baseado em bom senso e critérios de engenharia para se chegar a bons resultados.

5.2 Metodologia utilizando redes neurais

Neste capítulo, são feitas as análises referentes ao desempenho do sistema robótico através de gráficos correspondentes à linearização do sistema por redes neurais. Para isso são feitas as representações do sistema utilizando rede neural na substituição dos blocos da planta e controle não linear, utilizando-se para isto o ambiente MATLAB®.

Os blocos são substituídos por duas redes neurais, onde uma rede é treinada para representar o bloco de linearização, e uma segunda rede é treinada para representar a planta, conforme Figura 5.1, sendo posteriormente consideradas em conjunto na seção 5.3.

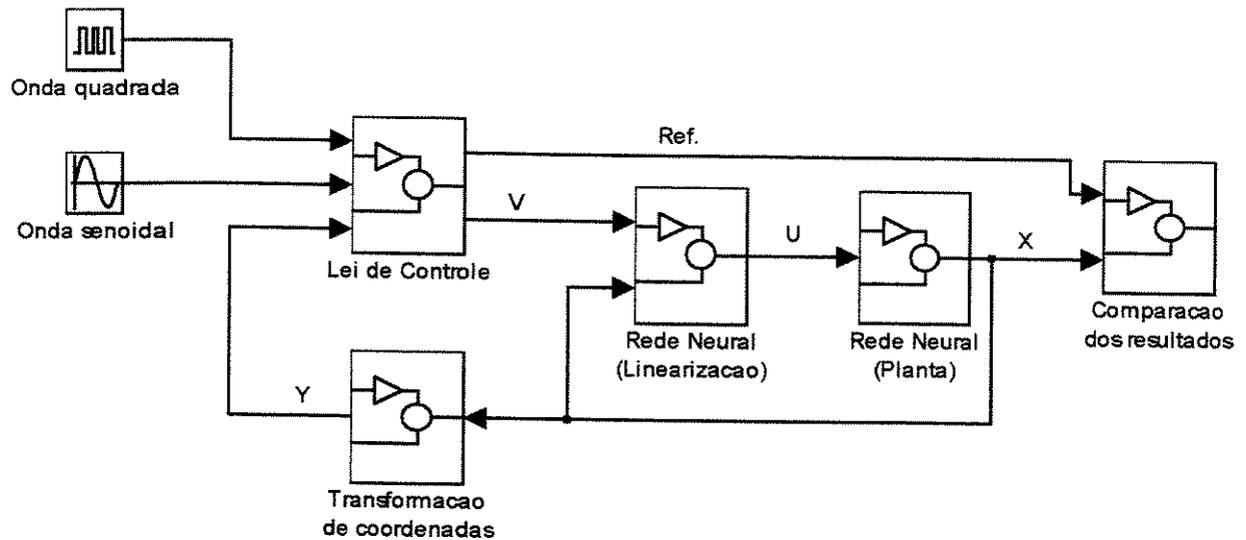


Figura 5.1 – Diagrama de blocos das redes neurais aplicadas ao sistema de linearização e representação da planta

5.2.1 Rede neural de linearização

A rede neural de linearização resume-se em uma estrutura composta de múltiplas camadas, sendo uma camada intermediária com dez neurônios, oito entradas, sendo: seis de realimentação do estado X e uma para cada sinal da Lei de Controle V_1 e V_2 , e três saídas, representadas nas Figuras 5.2, 5.3 e 5.4, chamando-se este sinal de saída desejada ou *target*. A rede foi treinada com um máximo de cinquenta iterações para atingir uma somatória do erro ao quadrado de cinco milésimos. Os sinais utilizados na rede neural de linearização são obtidos a partir do modelo aplicado pela linearização por realimentação do capítulo anterior.

Como pode ser observado nas Figuras 5.2, 5.3 e 5.4, a amplitude dos sinais apresenta-se relativamente alta para um tratamento por redes neurais, tornando-se conveniente transformar os sinais desejados com amplitude no intervalo ± 1 ($-1 < A < +1$), possibilitando uma melhor atuação da rede.

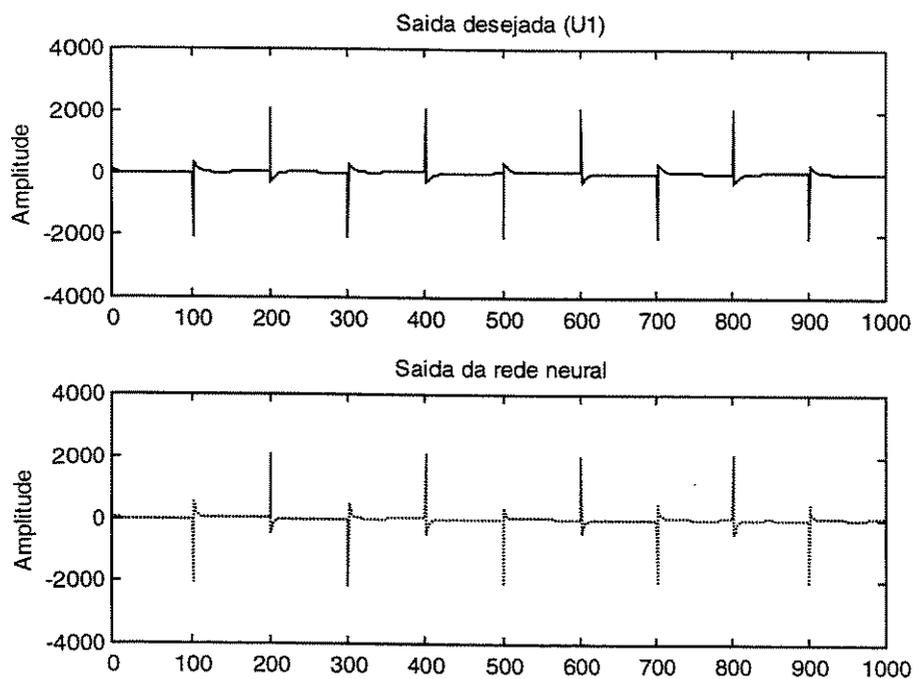


Figura 5.2 – Sinal da primeira saída da rede de linearização

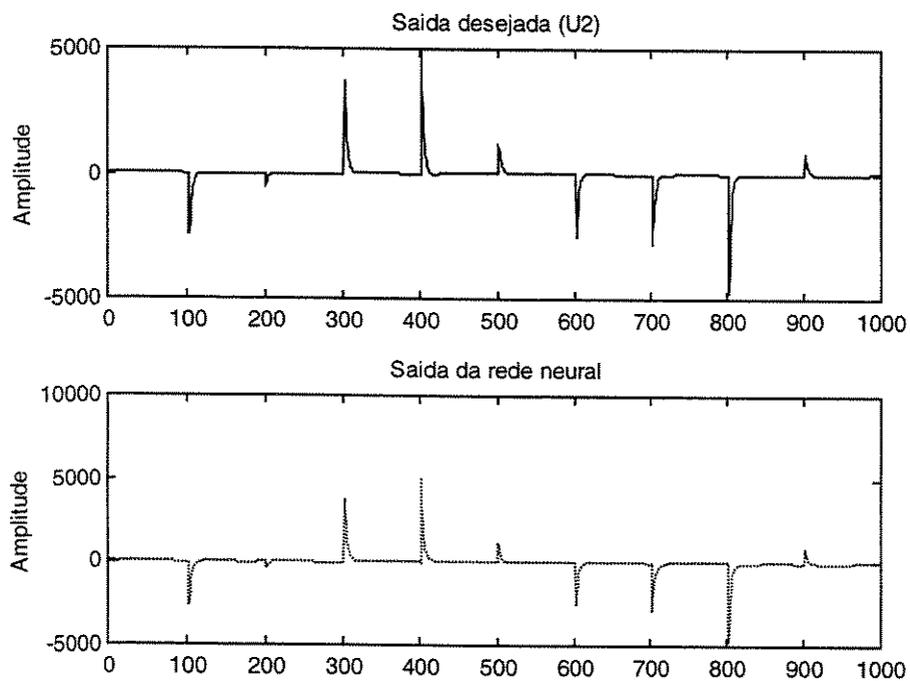


Figura 5.3 – Sinal da segunda saída da rede de linearização

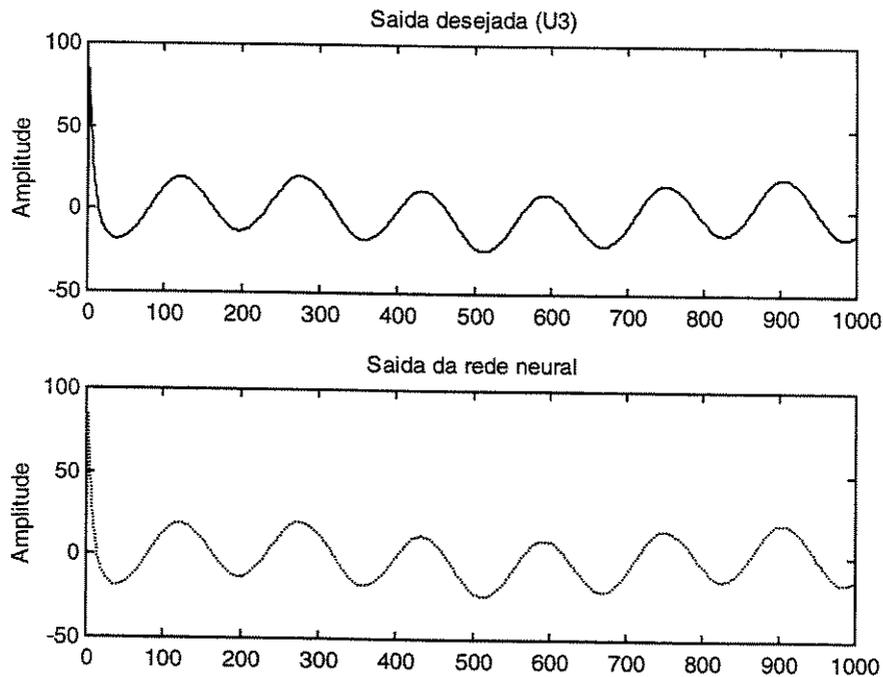


Figura 5.4 – Sinal da terceira entrada da rede de linearização

O que se pode observar, é que a rede neural obteve um bom resultado de aproximação ao sinal desejado. Os três gráficos acima representam cada um dos sinais que compõe a saída da linearização do sistema, utilizados como entrada para a rede neural que representa a planta.

5.2.2 Rede neural da planta

A estrutura da rede neural que representa a planta contém múltiplas camadas, sendo uma camada intermediária com dez neurônios, a última camada que contém seis saídas, as quais são representadas pelos gráficos das Figuras 5.5 a 5.10, e três entradas, sendo representadas pelas três saídas da rede de linearização. A rede foi treinada com um máximo de cinquenta épocas, buscando atingir uma somatória dos erros ao quadrado de saída equivalente a cinco milésimos. Os gráficos abaixo representam a comparação dos sinais desejados e os sinais obtidos na saída da rede.

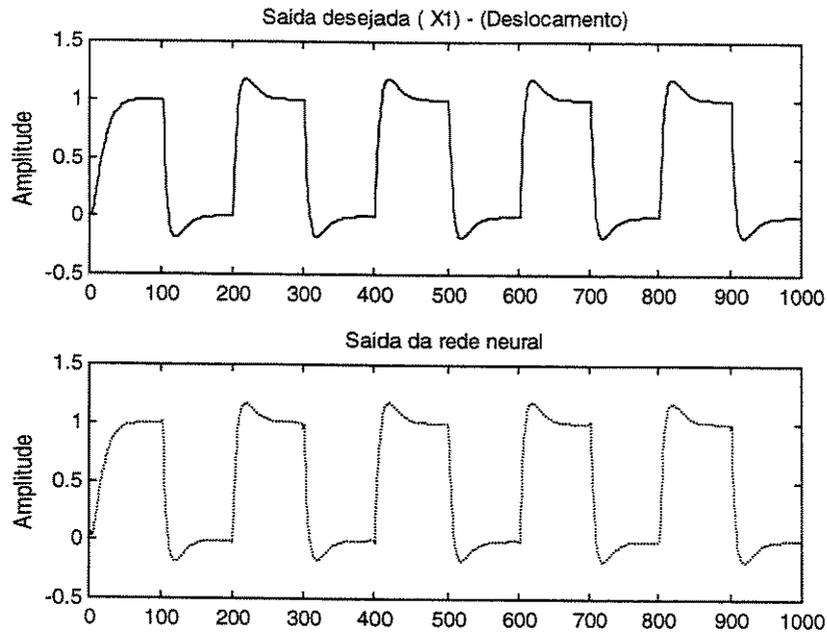


Figura 5.5 – Comparação das saídas desejada e da rede, representando a posição do corpo1

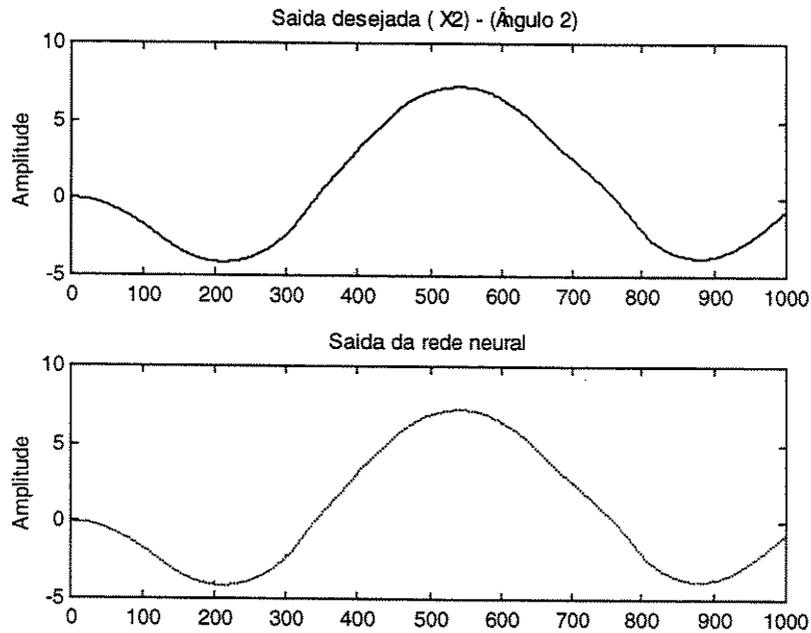


Figura 5.6 – Comparação das saídas desejada e da rede, representando a posição angular do corpo2

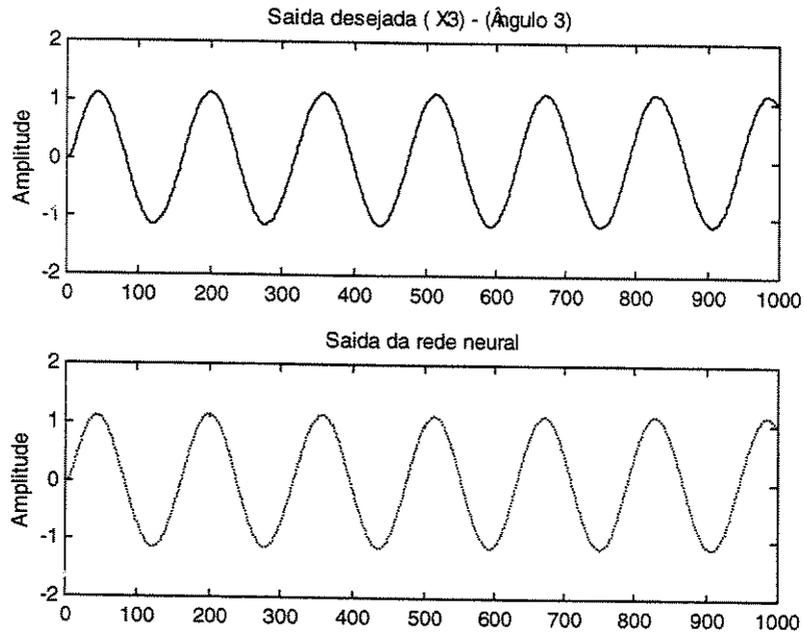


Figura 5.7 – Comparação das saídas desejada e da rede, representando a posição angular do corpo3

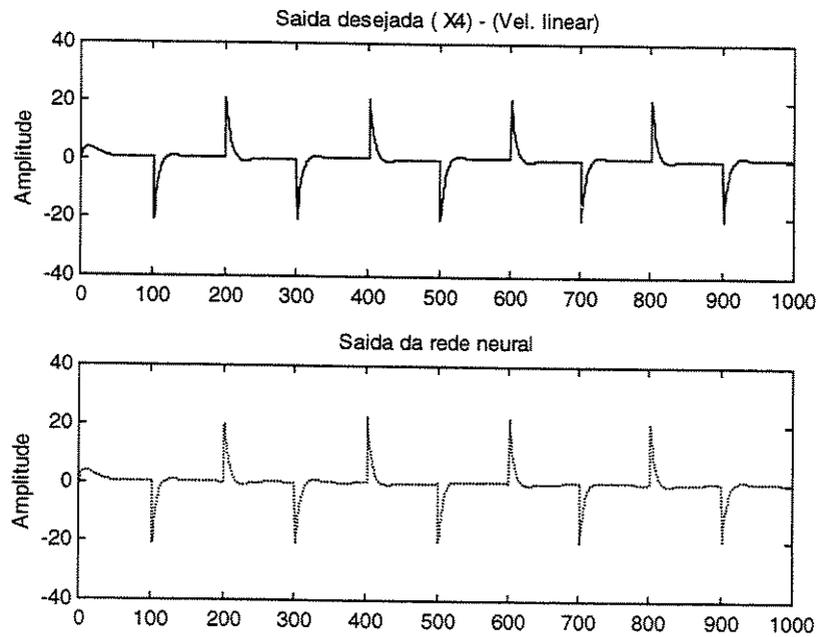


Figura 5.8 – Comparação das saídas desejada e da rede, representando a velocidade angular do corpo1

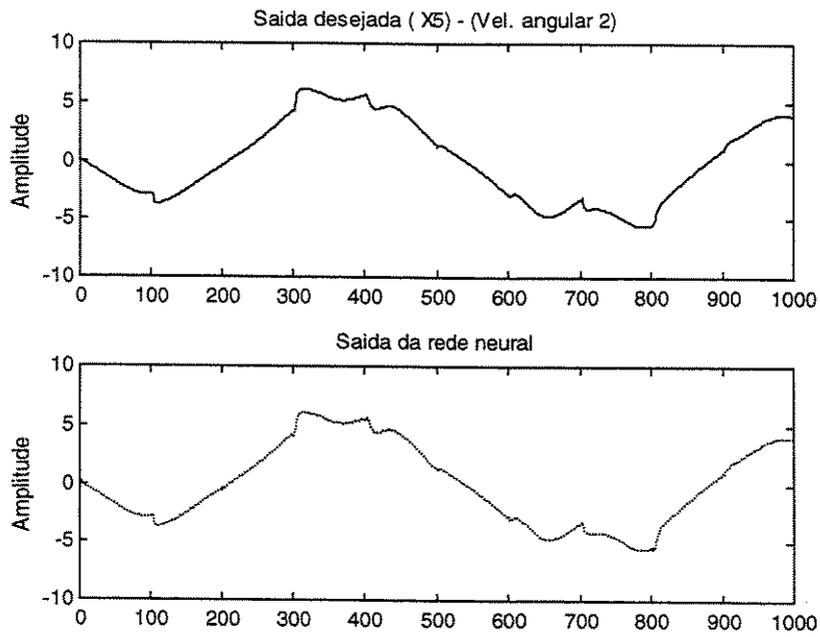


Figura 5.9 – Comparação das saídas desejada e da rede, representando a velocidade angular do corpo2

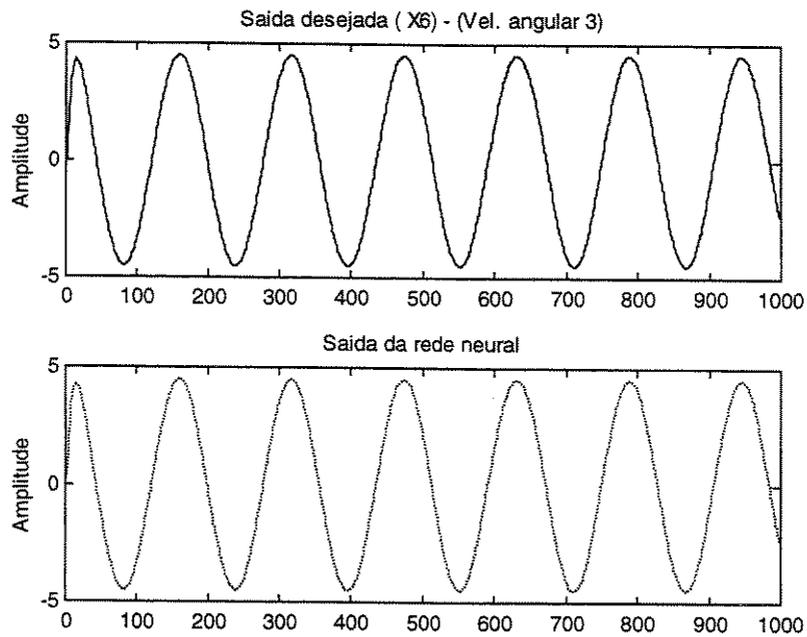


Figura 5.10 – Comparação das saídas desejada e da rede, representando a velocidade angular do corpo3

Novamente, pode-se notar o bom desempenho da rede neural que representa a planta, onde observa-se o pequeno erro obtido pela diferença dos sinais, estes que representam as posições e respectivas velocidades dos três corpos.

5.3 Substituição do linearizador por uma rede neural representada no Simulink

Para substituir o linearizador do diagrama de blocos da linearização por realimentação pela rede neural, foram utilizados os pesos e os *bias* semelhantes à rede treinada anteriormente, apresentada neste capítulo. A pequena alteração é que a nova estrutura é composta por vinte neurônios na camada intermediária e o sinal de entrada da rede é o original, ou seja, não houve necessidade de normalizá-lo. O número de épocas foi aumentado de cinqüenta para duzentos e dobrou-se o número de neurônios da camada intermediária visando um menor erro de aproximação da função ($SQE = 0.0002$), para que o sinal de entrada da planta, gerado pela rede, não interfira na resposta desejada, como foi constatado pela utilização da mesma arquitetura de rede da aplicação anterior.

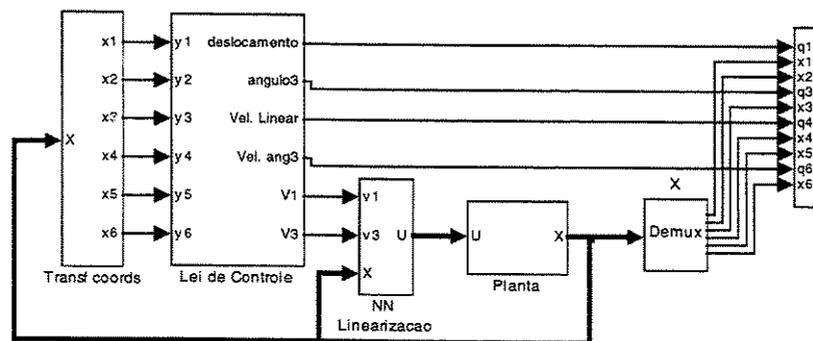


Figura 5.11 – Diagrama de blocos do sistema com uma rede neural de linearização

Normalmente, o sinal de entrada da rede é formado pelos sinais de referência e pela realimentação de seu próprio sinal de saída, tendo-se em vista que, quanto mais informações para a rede, melhor deveria ser seu desempenho. Uma pequena diferença apresentada nesta montagem é a realimentação feita com o sinal de saída da planta, mostrando-se com isto, a não linearidade

dos subsistemas pela sua relação de entrada-saída, ou seja, rede e planta não lineares, onde se houvesse um ou outro bloco linear, esta relação não seria possível.

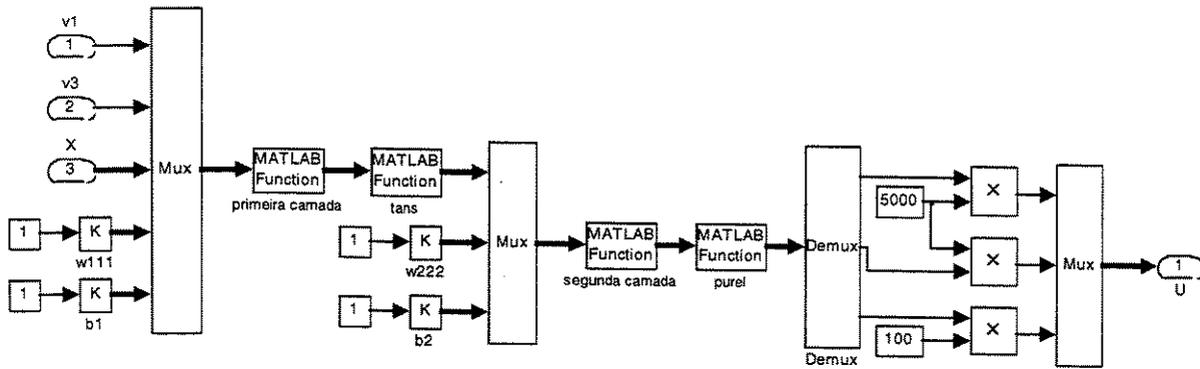


Figura 5.12 – Diagrama de blocos do subsistema da rede neural de linearização

O sinal de saída da rede deve ser multiplicado pelo mesmo fator que foi utilizado na normalização do sinal, aplicado anteriormente no treinamento da rede. Neste caso são usados os fatores 5000 e 100 para que os respectivos sinais de entrada da rede permaneçam entre o intervalo de -1 e $+1$ para o treinamento, e depois retornando-se ao valor original para a simulação.

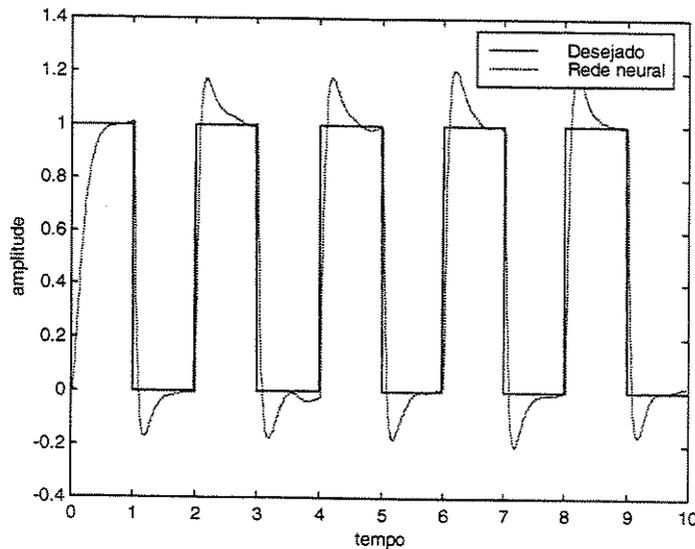


Figura 5.13 – Aproximação da curva que representa o deslocamento do corpo 1

O gráfico da Figura 5.13 torna-se crítico na análise de aproximação da rede ao sinal desejado, devido ao sobressinal acentuado em relação ao sinal real. No entanto, a rede demonstra um bom desempenho na sua função de acompanhamento da trajetória desejada.

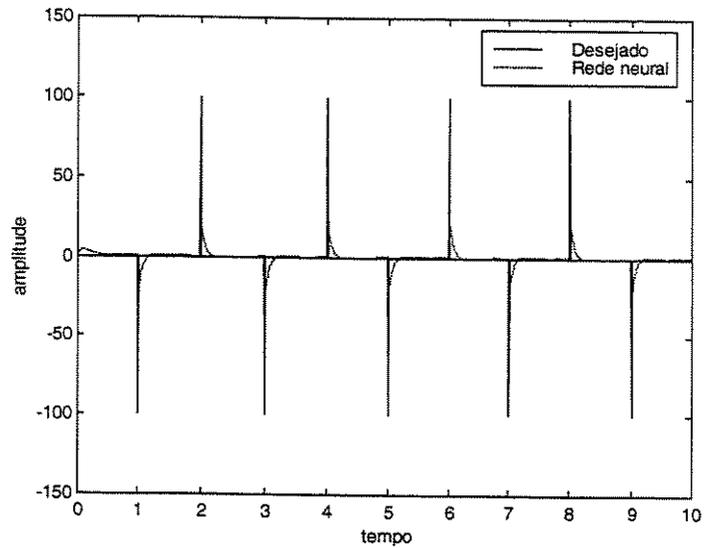


Figura 5.14 – Aproximação da curva que representa a velocidade do corpo 1

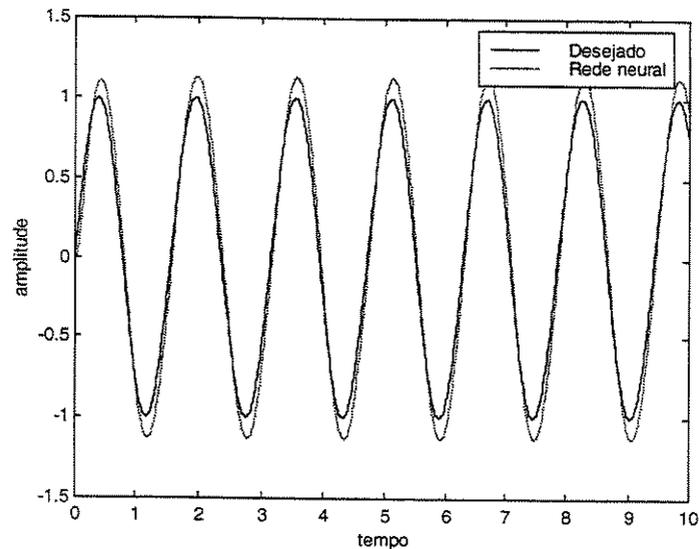


Figura 5.15 – Aproximação da curva que representa o deslocamento do corpo 3

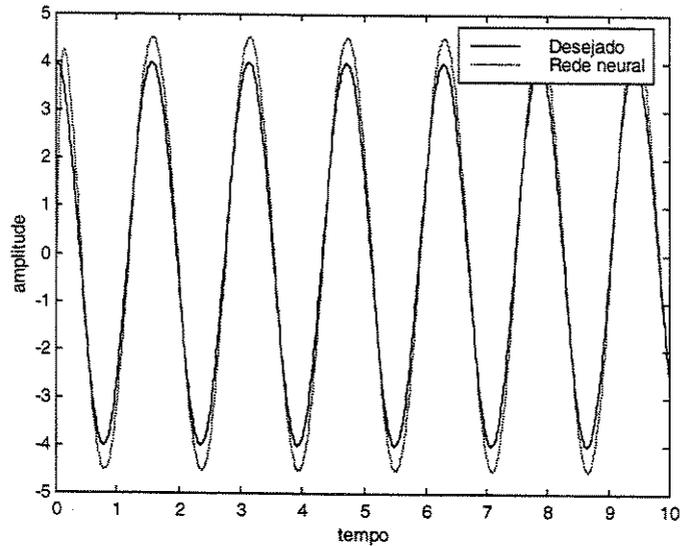


Figura 5.16 – Aproximação da curva que representa a velocidade do corpo 3

Novamente, é apresentada a facilidade com que a rede neural acompanha um sinal de onda “senoidal”, onde representam o deslocamento e a velocidade do corpo 3 nas Figuras 5.15 e 5.16, respectivamente.

A rede alcançou excelente desempenho na aproximação de sinais de altas frequências, mas mostrou deficiência em acompanhar sinais de baixas frequências, pois não foi treinada para acompanhar este tipo de sinal. Esta limitação pode ser analisada pelo gráfico da Figura 5.17, onde o sinal de excitação é representado por uma onda senoidal de baixa frequência. Já a linearização por realimentação mostrou-se eficiente para altas e baixas frequências, não acusando qualquer deficiência na alteração de seu sinal de excitação.

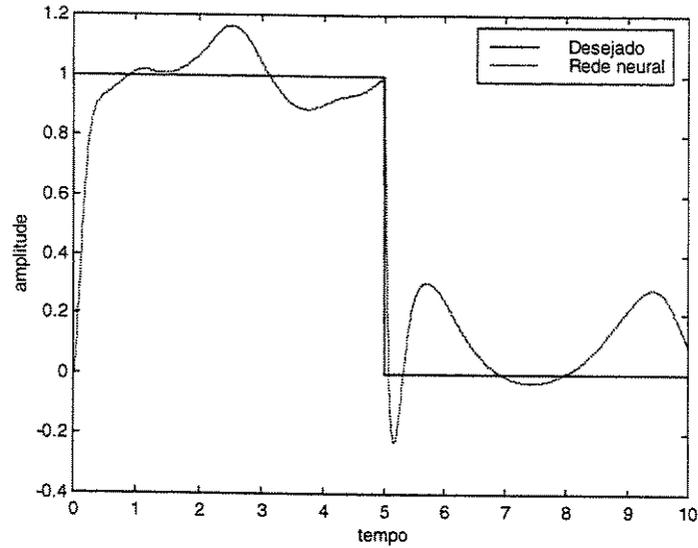


Figura 5.17 – Rede neural em sinais de baixa frequência representando o deslocamento do corpo 1

A frequência do sinal de entrada foi alterado de 0.5 Hz para 0.1 Hz, exatamente para exemplificar a limitação da rede em baixas frequências.

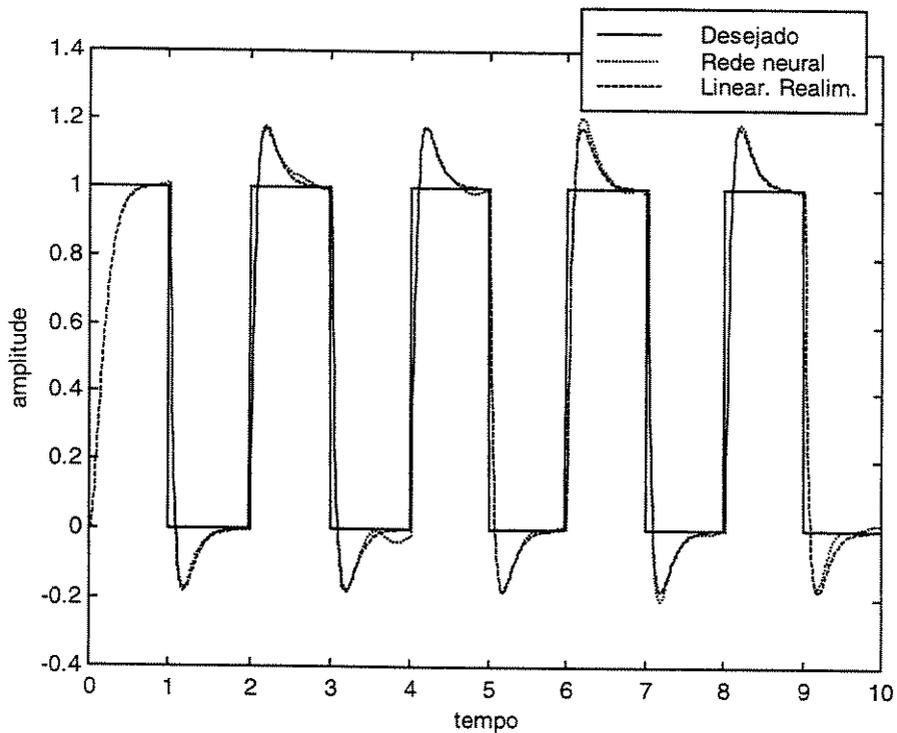


Figura 5.18 – Comparação dos sinais que representa o deslocamento do corpo 1

Finalizando e completando a análise comparativa entre os métodos aplicados neste trabalho, a Figura 5.18 ilustra a os três sinais sobrepostos, sinal desejado, rede neural de linearização e a linearização por realimentação, onde é feita uma completa comparação entre eles, podendo-se com isto, observar um sobressinal normal aos sinais resultantes de linearização, sendo também notada uma pequena distinção entre estes dois últimos.

Capítulo 6

Conclusão e propostas para futuros trabalhos

6.1 Conclusão

A estratégia do trabalho inicia-se pela utilização das bases em controle adaptativo usando redes neurais aplicadas ao método de linearização de sistemas não lineares, mostrando-se assim, que uma rede neural consegue acompanhar a trajetória desejada do modelo de referência. As técnicas de tratamento dos sinais decorrentes dos sistemas dinâmicos não lineares mostraram-se totalmente eficazes, buscando seguir as linhas de pesquisa na área de inteligência artificial e apresentando resultados animadores em relação às expectativas do trabalho. Este método demonstrou a atual empregabilidade de redes neurais como diferentes soluções a problemas de sistemas não lineares.

A aplicação de redes neurais de múltiplas camadas para a identificação do sistema corresponde a uma ferramenta ideal, pela obtenção da resposta desejada, devido a sua capacidade de acompanhar a trajetória obtida na simulação do movimento robótico, ainda quando o sistema encontra-se sem controle, ou seja, seus movimentos são caracterizados por seus próprios parâmetros e dados iniciais de entrada, não sofrendo qualquer intervenção que limite sua trajetória.

O método de linearização por realimentação está sendo utilizado para estabelecer uma estrutura analítica, mostrando-se como uma ferramenta prática e simples em comparação ao

complexo problema de sistema não linear presente neste trabalho. A técnica de linearização por realimentação tem sido uma poderosa técnica para linearização de sistemas robóticos. De acordo com os gráficos, pode-se observar que a resposta do sistema linearizado por realimentação foi decisivo na performance do sistema e satisfatório de uma forma geral. Isto demonstra que, apesar de uma pequena variação na resposta relativos ao processo de linearização, o método é completamente capaz e eficiente.

O linearizador e a planta substituídos por redes neurais demonstram um excelente desempenho de aproximação da curva, comprovando que a técnica apresentada pode ser aplicada em sistemas não lineares de manipuladores. A resposta obtida através da rede neural é semelhante à resposta obtida pela linearização por realimentação, possuindo ainda uma característica de se adaptar à diferentes situações de entrada e saída do sistema.

Pode-se mostrar a aplicação de redes neurais a um sistema não linear através dos resultados do trabalho exibidos através dos gráficos obtidos com o ambiente MATLAB® e sua ferramenta Simulink, utilizados para simular os movimentos do manipulador e avaliar o desempenho do sistema. As técnicas de controle adaptativo por modelo de referência formam a base da aplicação das redes neurais por apresentar uma relação entrada/saída mostrando-se ideal em sistemas com parâmetros desconhecidos e com alto nível de complexidade como os sistemas não lineares.

6.2 Propostas para trabalhos futuros

O manipulador estudado contém um espaço de trabalho planar, ou seja, bidimensional, onde propõe-se aumentar um grau de liberdade à base deste ou colocá-lo sobre uma plataforma móvel, possibilitando um movimento giratório ou deslizante, deixando este de ser planar, passando a ter movimentos nos três eixos de coordenadas.

Propõe-se a implantação de outros métodos de controle do sistema como controle adaptativo ou controle robusto, podendo-se fazer uma comparação de sua performance com a aplicada em linearização por realimentação e redes neurais.

Poder ser estudado a construção de um robô que se aproxime às características propostas no trabalho, para validação das técnicas aplicadas e comparação dos métodos utilizados, através de experimentos de controle e movimentos, podendo ser considerado também o amortecimento dos corpos.

Uma outra proposta seria a implementação de programação em linguagem C, eliminando algumas limitações do programa executado no ambiente MATLAB®, como a limitação dos dados do modelo e a velocidade de execução do programa. Pode-se ainda analisar seus movimentos por outras ferramentas de simulação como o ambiente *Working Model*®, apenas como ilustração e representação gráfica dos movimentos.

Referências Bibliográficas

- Arbib, Michael A. editor. *The Handbook of Brain Theory and Neural Networks*. Massachusetts Institute of Technology, 1995.
- Asada, H. Haruhiko. Slotine, Jean-Jacques E. *Robot Analysis and Control*. John Wiley & Sons, Inc, 1986.
- Åström, Karl Johan / Wittenmark, Björn *Computer Controlled Systems*. Englewood Cliffs, New Jersey, Prentice Hall Inc, 1984.
- Åström, Karl Johan / Wittenmark, Björn *Adaptive Control*. Addison-Wesley Publishing Company, 1989.
- Beer, F. P. & Johnston Jr., E. R. *Mecânica Vetorial para Engenheiros: Dinâmica*. Vol. 2, McGraw-Hill do Brasil Ltda, 1973.
- Chalan, V. V. *Adaptive control systems*. Marcel Dekker, Inc, 1987.
- Craig, John J. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Publishing Company, Inc, 1989.
- D'Azzo, John J. Houpis, Constantine H. *Análise e Projeto de Sistemas de Controles Lineares*. Guanabara Dois, 1984.

- Demuth, Howard & Beale Mark. *Neural Network Toolbox For Use with MATLAB*. The Math Works, Inc, 1992-1993.
- Dorf, R. C. & Bishop, R. H. *Modern Control Systems*. Addison-Wesley Publishing Company, 1995.
- Elgerd, O. I. *Control Systems Theory*. McGraw-Hill, 1967.
- Eykhoff, Pieter. *System Identification Parameter and State Estimation*. John Wiley & Sons Ltd, 1974.
- Ferreira, E. P. *Robótica Básica*. Versão preliminar publicada para a V Escola Brasileiro-Argentina de Informática. Rio de Janeiro, 1991.
- Fu, K. S. Gonzales, R. C. Lee, C. S. G. *Robotics: control, sensing, vision and Intelligence*. McGraw-Hill, Inc, 1987.
- Haykin, Simon. *Neural Networks. A Comprehensive Foundation*. Prentice-Hall, Inc, 1999.
- Hecht-Nielsen, Robert. *Neurocomputing*. Addison-Wesley Publishing Company, Inc, 1990.
- Hsu, Hwei Piao. *Análise de Fourier*. Livros Técnicos e Científicos, Rio de Janeiro, 1973.
- Irwin, G. W., Warwick, K., and Hunt, J. K., eds., "Neural Networks for Control Systems". *IEEE Control Engineering Series 46*. London: Peter Peregrinus, 1992.
- Isidori, Alberto. *Nonlinear control systems*. Springer-Verlag Berlin, 1989.
- Kalman, R. E. "On the general theory of control systems". *1st IFAC congress, Moscow*, pp 481-492, 1960.
- Kenjo, Takashi. *Stepping motors and their microprocessor controls*. Oxford, Clarendon, 1984.

- Klafter, R. D., Chmielewski, T. A. & Negin, Michael. *Robotic engineering: an integrated approach*. Robotica. Englewood Cliffs, Prentice-Hall, 1989.
- Koivo, A. J. *Fundamentals for control of robotic manipulators*. New York, J. Wiley, 1989.
- Kovács, Z. L. *Redes neurais artificiais: fundamentos e aplicações*. 2ª ed. São Paulo: Edição Acadêmica, 1996.
- Kuo, B. C. *Automatic Control Systems*. Prentice-Hall, Inc. 7th ed, 1995.
- Kuo, B. C. *Incremental motion control*. Volume II. SRL Publishing Company, Champaign, 1979.
- Kuo, B. C. *Digital Control Systems*. Saunders College Publishing, 1992.
- Lee, R. C. K. *Optimal Estimation, Identification, and Control*. M. I. T. Press, Cambridge, Massachusetts, 1964.
- Li, W. & Slotine, J. *Neural Network Control of Unknown Nonlinear Systems*. American Control Conference, Pittsburgh, 1989
- Ljung, Lennart. *System Identification*. Prentice-Hall, Inc, 1987.
- McCulloch, W. S., Pitts, W. "A logical calculus of the ideas immanent in nervous activity". *Bulletin of Mathematical Biophysics*, vol 5, pp. 115-133, 1943
- Meriam, James L. *Dinâmica*. Livros Técnicos e Científicos S. A, 1994
- Minsky, M. and Papert, S. *Perceptrons: An Introduction to Computational Geometry*. M. I. T. Press, Cambridge, Massachusetts, 1969
- Murray, Alan F. *Applications of Neural Networks*. Kluwer Academic Publishers, 1995.

- Narendra, K. S. and Parthasarathy, K. "Identification and Control of Dynamical Systems Using Neural Networks". *IEEE Transactions on Neural Networks*, vol. 1, pp 4-27, no. 1, march 1990.
- Nguyen, D. & Widrow, B. "Neural Networks for self-Learning Control systems", *IEEE Control Magazine Systems*, 1990.
- Noriega, J. R. and Wang, Hong. "A Direct Adaptive Neural Network Control for Unknown Nonlinear Systems and Its Application". *IEEE Transactions on Neural networks*, vol. 9, n° 1, pp 27-33, January 1998.
- Ogata, K. *Discrete-time control systems*. Prentice-Hall, 1987
- Ogata, K. *Engenharia de Controle Moderno*. Prentice-Hall do Brasil 1982
- Pal, C. / Kayaba, N. Morishita, S. and Hagiwara, I. "Dynamic System Identification by Neural network". *JSME International Journal*, vol. 38, n° 4, pp 686-692, 1995.
- Paul, Richard P. *Robot manipulators: mathematics, programming, and control*. Massachusetts Institute of Tecnology, 1981.
- Rumelhart, D. E., & McClelland, J. L. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT, 1986.
- Santos, Ilmar. F. *Cinemática e Dinâmica de Sistemas Dinâmicos*. FEM-UNICAMP, 1998
- Simpson, Patrick K., editor. "Neural networks application". *IEEE Technology Update Series*, 1996
- Slotine, J. J. and Li, W. *Applied non-linear control*. Prentice Hall, Englewood Cliffs, New Jersey, 1991

- Spong, Mark W. "Modeling and control of elastic joint robots". *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 109, n° 4, pp 310-319, Dec 1987.
- Spong, Mark W. & Vidyasagar, M. *Robot Dynamics and Control*. John Wiley & Sons, Inc, 1989. [bae 629.892/Sp66r]
- Stone, Henry W. *Kinematic Modeling, Identification, and Control of Robotic Manipulators*. Kluwer Academic Publishers, 1987.
- Suykens, Johan A. K. / Vandewalle, Joos P. L. / Moor, Bart L. R. De. *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*. Kluwer Academic Publishers, 1997.
- Tham, M. T., Morris, A. J. and Montague, G. A. "Soft-sensing: A solution to the problem of measurement delays". *Chem. Eng. Res. and Des.*, 67, 6, pp 547-554, 1989.
- Valavanis, K. P. & Saridis George N. *Intelligent robotic systems : theory, design and applications*. Boston, Kluwer, 1992.
- Yuan, M./ Poo, A. N. & Hong, G. S. "Direct neural control system: Nonlinear extension of adaptive control". *IEE Proc.-Control Theory Appl.*, vol. 142, n° 6, pp 661-667, Nov 1995.
- Zhihong, M./ Wu, H. R. & Palaniswami, M. "An Adaptive Tracking Controller Using Neural Networks for a Class of Nonlinear Systems". *IEEE Transactions on Neural Networks*, vol. 9, n° 5, pp 947-955, Sep 1998.

Anexo I

A evolução das redes neurais artificiais por assunto

1943	McCulloch-Pitts neuron	McCulloch, Pitts
1957	Perceptron	Rosenblatt
1960	Madaline	Widrow
1969	Cerebellatron	Albus
1974	Backpropagation network	Werbos, Parkwr, Rumelhart
1977	Brain state in a box	Anderson
1978	Neocognitron	Fukushima
1978	Adaptive resonance theory	Carpenter, Grossberg
1980	Self-organizing map	Kohonen
1982	Hopfield net	Hopfield
1985	Bidirectional assoc. mem.	Kosko
1985	Boltzmann machine	Hinton, Sejnowsky, Szu
1986	Counterpropagation	Hecht-Nielsen
1988	Cellular neural network	Chua, Yang

Anexo II

Pesquisadores em redes neurais

1943	McCulloch e Pitts
1948	Wiener
1949	Hebb
1957	Rosenblatt
1958	Widrow e Hoff
...	...
1969	Minsky e Papert
...	
1960 - 1980	Kohonen, Grossberg, Widrow, Anderson, Caianiello, Fukushima,...
...	...
1974	Werbos
...	...
1982	Hopfield
1986	Rumelhart e McClelland

Anexo III

Redes neurais – Aplicações e métodos (somente para referência)

Relação dos métodos conforme Arbib, 1995:

1. *Perceptron* de simples camada (*adaline*);
2. Algoritmo adaptativo do método dos mínimos quadrados (*LMS*); Regra de aprendizado Widrow-Hoff;
3. Filtros Wiener; Filtros Kalman; Algoritmo dos mínimos quadrados recursivo;
4. *Perceptron* de múltiplas camadas com resposta ao impulso em tempo finito;
5. *Perceptron* de múltiplas camadas treinado por retro-propagação (*MLP*);
6. Retro-propagação em redes recorrentes;
7. *Neocognitron*;
8. *Wavelets*;
9. Acompanhamento de projeção;
10. Rede com teoria de ressonância adaptativa;
11. Funções de figura de classificação de mérito e erro mínimo de classificação;
12. Análise dos componentes principais;
13. Classificador com rede neural Bayesian;
14. Modelos de média de movimento auto-regressivo; Modelos de regressão linear múltipla;
15. Redes de convolução *feedforward*;
16. Programação não linear;
17. Algoritmo Viterbi para encontrar a trajetória mais curta de uma função;
18. Modelos de regressão não linear aditiva tal como expectativa condicional aditiva;

19. Mapeamento topográfico (Kohonen);
20. *Learning Vector Quantization (LVQ)*;
21. Modelos Markov;
22. Redes neurais de tempo de atraso;
23. Aprendizado de reforço;
24. Otimização via Hopfield;
25. Função de bases radiais.

APLICAÇÕES	MÉTODOS																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
FILTRO ADAPTATIVO	x	x																							
PROCESS. DE SINAL ADAPTATIVO		x	x	x		x																			
RECONHECIMENTO AUTOMÁTICO		x			x		x	x	x	x	x														
ASTRONOMIA					x							x	x												
PROGNÓSTICO						x								x											
SISTEMAS INTELIGENTES					x																				
RECONHECIMENTO DE CALIGRAFIA					x										x	x	x								
FÍSICA DE ALTA TENSÃO					x	x																			
GERENCIAMENTO DE INVESTIMENTO					x	x						x		x					x						
REDUÇÃO DE RUÍDO	x	x	x																						
ANÁLISE DE ESTRUTURA PROTEICA					x															x					
RECONHECIMENTO DE VOZ					x	x						x								x	x				
IDENTIFICAÇÃO ACÚSTICA														x								x			
PRODUÇÃO DE AÇO					x																				
TELECOMUNICAÇÕES		x	x		x																			x	x
PROCESSAMENTO VISUAL DE FORMATO DE OBJETO										x										x	x				x

Anexo IV

1. Dados característicos do robô projetado

A proposta de analisar o desempenho de um mecanismo robótico requer uma aproximação do modelo com parâmetros reais, buscando-se uma simulação com dados baseados em características de alguns componentes do projeto. Apresenta-se para isto, um descritivo com a montagem do equipamento somente para obtenção de dados para o desenvolvimento do projeto, ou seja, os dados fornecidos a seguir são apenas ilustrativos, e não são considerados na simulação deste trabalho. Descreve-se a seguir os componentes principais do mecanismo, que são: a motorização, a transmissão por correia e o fuso.

2. Opção por utilizar motor de passo

Após muita pesquisa, o motor de passo deixa os laboratórios e torna-se comercial no início dos anos 60 com o advento do transistor. Nos anos 70 sofreu rápido desenvolvimento graças ao seu emprego em “drives” de computadores pessoais (*PC's*). Nos anos 80, aumentou-se o interesse no âmbito mundial devido a sua precisão e versatilidade, principalmente nas aplicações industriais. A preocupação ambiental decorrente da emissão de gases pelos veículos automotivos deram nova impulsão aos motores de passo nos anos 90, a ponto de cada novo veículo ter um desses motores em seu sistema de injeção eletrônica. Pode-se dizer atualmente, que o motor de passo é a síntese da automação industrial em movimentos precisos e programáveis, além de proporcionar uma excelente relação custo-benefício.

É um mecanismo eletro-mecânico que concede movimentos angulares pré estabelecidos de maneira uniforme e altamente precisa. O tipo de sinal de entrada é em sua maioria “digital”, onde um pulso corresponde a um preciso movimento angular (incremental). O motor de passo trabalha em malha aberta, isto é, não sofre uma manutenção corretiva de sensores como *encoders* ou *resolvers* (Kenjo-1984, Kuo-1979).

Sua estrutura é composta de um estator onde são enroladas as bobinas (rotor), dois rolamentos e mancais (tampas). Os motores atualmente usados apresentam oito bobinas e apenas duas fases de energização. O rotor apresenta um polo norte e um polo sul definidos, e é a magnetização seqüencial do estator que faz o rotor girar e deslocar sua carga. O ângulo de deslocamento do rotor varia conforme o fabricante, mas é comum encontrar o de 1,8 graus (o motor completa uma volta a cada duzentos pulsos). Limita-se a torques de no máximo 31,5 Nm e rotações até 1500 rpm.

Vantagens:

- Baixo custo;
- Mecanismo simples, não necessita manutenção;
- Não danifica em *stall*;
- Erro de posicionamento não cumulativo;
- Alto torque em baixas rotações;
- Resistente a choques mecânicos;
- Fácil de programar.

Desvantagens:

- Malha aberta, sem monitoração corretiva (*feedback*);
- Baixas e médias velocidades de trabalho;
- Potências relativamente baixas ($\approx 4\text{HP}$);
- Baixas acelerações e desacelerações;
- Resonância em certas velocidades

3. Transmissão por correia

As correias sincronizadoras, também conhecidas como correias “dentadas”, atuam pelo princípio de transmissão de movimento através de dentes, semelhante aos da engrenagem. Os sulcos das polias são envolvidos pelos dentes da correia proporcionando um alto rendimento à transmissão. A escolha deste tipo de transmissão deve-se às vantagens de possuir um engrenamento anti derrapante e preciso, não necessita lubrificação, ausência de ruído, baixo custo, além de absorver os choques mecânicos causados pelos movimentos.

Características dos componentes:

Relação de transmissão: 1:2.

Correia trapezoidal: passo = 5.08 mm; largura = 9.5 mm; 70 dentes; circunferência primitiva = 355 mm.

Polia do corpo 3: passo = 5.08 mm; 18 sulcos; diâmetro primitivo = 29 mm.

Polia do corpo 2: passo = 5.08 mm; 36 sulcos; diâmetro primitivo = 58 mm.

4. Fuso

O fuso é embutido no corpo 2 proporcionando o deslocamento do corpo 1 com a função de transformar o movimento rotativo do motor em um movimento prismático.

Características:

Rosca M10; passo da rosca 1.5 mm; comprimento = 250 mm

5. Dados ilustrativos

Considerando o robô em posição de repouso ($L = 210$ mm), propõe-se executar um movimento vertical para cima de 50 mm conforme indicado na figura abaixo.

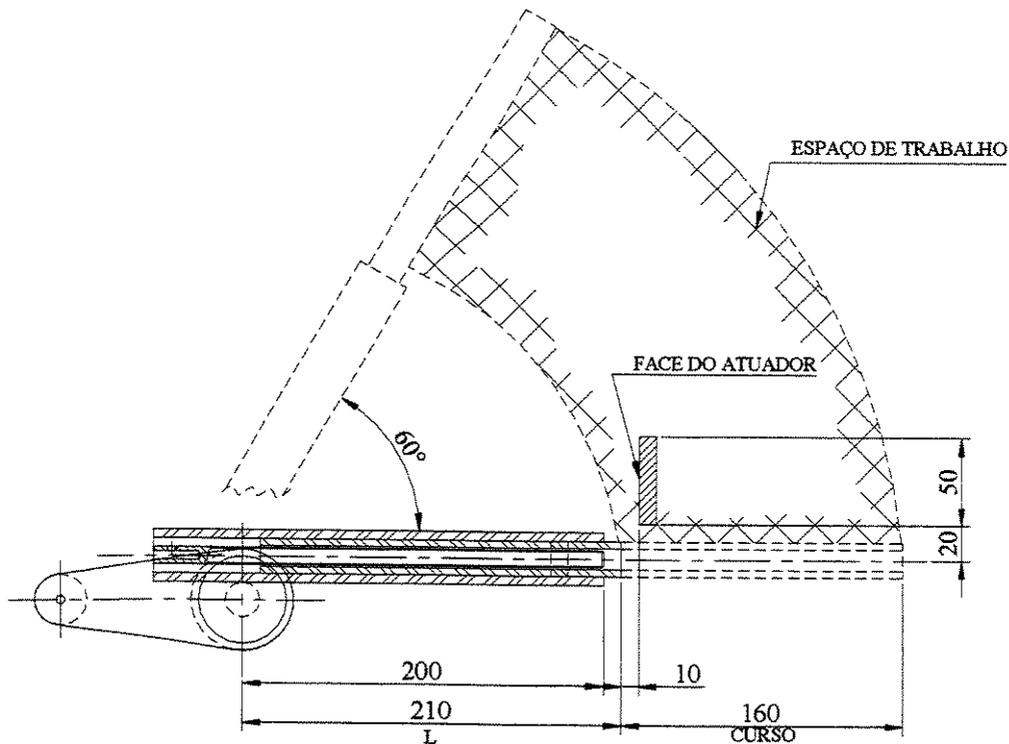


figura 7.3 – Representação esquemática da atuação do manipulador

Movimento da posição de repouso até o início da operação:

Motor 3: rotação positiva em 5.4°

→ 6 períodos;

Motor 1: avanço em 13.16 mm

→ 8.77 voltas;

→ 1754 períodos;

Movimento desejado de operação:

Motor 3: rotação positiva em 11.7°

→ 13 períodos;

Motor 1: avanço em 15.17 mm

→ 10.11 voltas;

→ 2022 períodos;

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE