

UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO  
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DISSERTAÇÃO DE MESTRADO

**CONTROLE DE UM BRAÇO ROBÓTICO UTILIZANDO  
UMA ABORDAGEM DE AGENTE INTELIGENTE**

**TARIG ALI ABDURRAHMAN EL SHHEIBIA**

Campina Grande - PB

Julho de 2001

UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO  
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

# **CONTROLE DE UM BRAÇO ROBÓTICO UTILIZANDO UMA ABORDAGEM DE AGENTE INTELIGENTE**

*Dissertação apresentada à coordenação de Pós-graduação em informática – COPIN – da Universidade Federal da Paraíba – UFPB, como requisito parcial para a obtenção do grau de Mestre em Informática.*

**TARIG ALI ABDURRAHMAN EL SHHEIBIA**

Orientador: José Homero Feitosa Cavalcanti, Dr.

Campina Grande – PB

Julho de 2001

Ficha catalográfica

---

SHHEIBIA, Tarig Ali Abdurrahman El

S543C

Controle de um Braço Robótico Utilizando uma Abordagem de Agentes Inteligentes

Dissertação (Mestrado) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, Coordenação de Pós-Graduação em Informática, Campina Grande, PB, Julho de 2001.

84 p. Il.

Orientador: José Homero Feitosa Cavalcanti

Palavras Chaves:

1. Inteligência Artificial
2. Robótica
3. Agente Inteligente
4. Lógica Fuzzy

CDU - 007.52

---

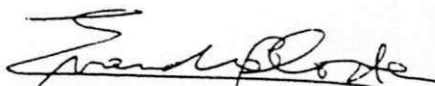
**CONTROLE DE UM BRAÇO ROBÓTICO UTILIZANDO UMA  
ABORDAGEM DE AGENTE INTELIGENTE**

**TARIG ALI ABDURRAHMAN EL SHHEIBIA**


**DISSERTAÇÃO APROVADA EM 06.07.2001**



**PROF. JOSÉ HOMERO FEITOSA CAVALCANTI, D.Sc**  
**Orientador**



**PROF. EVANDRO DE BARROS COSTA, D.Sc**  
**Examinador**



**PROF. JOSÉ ANTÔNIO RIUL, Dr.**  
**Examinador**

**CAMPINA GRANDE – PB**

أهدى هذا العمل  
إلى روح والدي  
على إشتاقية

*Dedico este trabalho,  
à memória de meu pai,  
Ali El Shheibia*

# AGRADECIMENTOS

Agradeço à Deus por ter me iluminado e dado força para a realização e conclusão deste trabalho.

Agradeço à minha família, que mesmo longe, se fizeram presente em cada momento desta etapa, através de seu incentivo e orações.

Sou muito grato ao meu orientador Prof. Dr. Homero Feitosa Cavalcanti pela orientação, esforço, apoio, incentivo e principalmente paciência.

Ao Professor Edilson Ferneda, que me recebeu aqui em Campina Grande e me deu bastante apoio nos momentos iniciais deste trabalho e ao Professor Herman Gomes pelas discussões e dicas sobre o trabalho.

Aos colegas do NEUROLAB, Alexsandro, Everaldo, Bruno, Sérgio, Ricardo, Christiane e aos colegas que me ajudaram na realização deste trabalho, especialmente Eloi, Lelê, Fabrício, Ernesto, Lidiana e Álvaro.

Aos meus amigos que fizeram dessa minha estadia em Campina Grande mais agradável: Renato, Hilmer, Gustavo, Ivette, Leandro, Kyller, Karina, Juliana, Juliana Barbosa, Sandro, Erica, Adriano, Jorge, Giovanni, Tenório (Maceió), Ary César, Daniel.

Meus agradecimentos especiais à Família Serey Guerrero: Dona Julia, Seu Jorge, Dalton, Verônica e Claudia por me acolherem como uma segunda família. E à minha namorada Claudia pela paciência e apoio dados neste período.

À todos os funcionários do DSC e COPIN, em particular Aninha, Vera, D. Inês e Romildo.

Aos meus amigos Ajuad e Mohamed pelo apoio e por tornar a distância entre a família menos difícil.

E à todas as pessoas que contribuíram direta ou indiretamente para a realização deste trabalho.

# SUMÁRIO

<b>LISTA DE FIGURAS</b>	<b>IV</b>
-------------------------	-----------

---

<b>LISTA DE TABELAS</b>	<b>VII</b>
-------------------------	------------

---

<b>LISTA DE QUADROS</b>	<b>VII</b>
-------------------------	------------

---

<b>RESUMO</b>	<b>VIII</b>
---------------	-------------

---

<b>ABSTRACT</b>	<b>IX</b>
-----------------	-----------

---

<b>1. INTRODUÇÃO</b>	<b>1</b>
----------------------	----------

---

<b>1.1 SISTEMAS INTELIGENTES</b>	<b>1</b>
----------------------------------	----------

<b>1.2 OBJETIVOS DO TRABALHO</b>	<b>3</b>
----------------------------------	----------

1.2.1 Objetivo Geral	3
----------------------	---

1.2.2 Objetivos Específicos	3
-----------------------------	---

<b>1.3 JUSTIFICATIVA</b>	<b>3</b>
--------------------------	----------

<b>1.4 ESTRUTURA DA DISSERTAÇÃO</b>	<b>4</b>
-------------------------------------	----------

<b>2. ROBÓTICA</b>	<b>6</b>
--------------------	----------

---

<b>2.1 INTRODUÇÃO</b>	<b>6</b>
-----------------------	----------

<b>2.2 O QUE É ROBÓTICA ?</b>	<b>7</b>
-------------------------------	----------

<b>2.3 CLASSIFICAÇÃO GERAL DOS ROBÔS</b>	<b>8</b>
--	----------

<b>2.4 CONFIGURAÇÃO DE UM ROBÔ</b>	<b>8</b>
------------------------------------	----------

<b>2.5 MANIPULADOR ROBÓTICO</b>	<b>10</b>
---------------------------------	-----------

2.5.1 Base Fixa	10
-----------------	----

2.5.2 Braço Articulado	10
------------------------	----

2.5.3	Unidade de Controle	11
<b>2.6</b>	<b>SISTEMAS DE ACIONAMENTO DE ROBÔS</b>	<b>12</b>
2.6.1	Acionamento Hidráulico	12
2.6.2	Acionamento Elétrico	13
2.6.3	Acionamento Pneumático	14
<b>2.7</b>	<b>APLICAÇÃO ROBÓTICA NA INDÚSTRIA</b>	<b>15</b>
<b>2.8</b>	<b>CONCLUSÃO</b>	<b>17</b>
<b>3. AGENTE INTELIGENTE</b>		<b>18</b>
<hr/>		
<b>3.1</b>	<b>INTRODUÇÃO</b>	<b>18</b>
<b>3.2</b>	<b>AGENTES</b>	<b>19</b>
3.2.1	Definições	19
3.2.2	Características Gerais de Agentes Inteligentes	21
3.2.3	Ambiente	23
3.2.4	Arquitetura de um agente	24
3.2.4.1	Arquiteturas cognitivas	25
3.2.4.2	Arquiteturas Reativas	26
3.2.4.3	Arquiteturas Híbridas	27
<b>3.3</b>	<b>CONCLUSÃO</b>	<b>29</b>
<b>4. LÓGICA FUZZY</b>		<b>30</b>
<hr/>		
<b>4.1</b>	<b>INTRODUÇÃO</b>	<b>30</b>
<b>4.2</b>	<b>FUNDAMENTOS DA LÓGICA FUZZY</b>	<b>32</b>
<b>4.3</b>	<b>OPERAÇÕES</b>	<b>34</b>
4.3.1	Complemento	34
4.3.2	União	35
4.3.3	Interseção	35
<b>4.4</b>	<b>EXPRESSÃO FUZZY DO CONHECIMENTO</b>	<b>36</b>
<b>4.5</b>	<b>SISTEMAS DE CONTROLE FUZZY</b>	<b>37</b>
<b>4.6</b>	<b>UTILIZAÇÃO DA LÓGICA FUZZY</b>	<b>38</b>
<b>4.7</b>	<b>PERSPECTIVAS DA LÓGICA FUZZY</b>	<b>39</b>
<b>4.8</b>	<b>CONCLUSÃO</b>	<b>40</b>
<b>5. MAUTOXAD</b>		<b>41</b>
<hr/>		



<b>5.1</b>	<b>INTRODUÇÃO</b>	<b>41</b>
<b>5.2</b>	<b>DESCRIÇÃO DO ROB OXAD E DO AUTOXAD</b>	<b>42</b>
<b>5.3</b>	<b>SISTEMA DE CONTROLE UTILIZANDO AGENTES INTELIGENTES</b>	<b>49</b>
5.3.1	Descrição do MAUTOXAD	51
5.3.1.1	Os Agentes do MAUTOXAD	51
5.3.2	Implementação do MAUTOXAD	56
5.3.1.2	Interface entre o usuário e o MAUTOXAD	61
5.3.3	Exemplos dos movimentos do MAUTOXAD	63
5.3.3.1	Jogada sem obstáculo	64
5.3.3.2	Jogada com Obstáculo	65
<b>5.4</b>	<b>CONCLUSÃO</b>	<b>69</b>
<b><u>6. CONCLUSÃO</u></b>		<b><u>70</u></b>
<b>6.1</b>	<b>CONTRIBUIÇÕES</b>	<b>70</b>
<b>6.2</b>	<b>DIFICULDADES ENCONTRADAS</b>	<b>71</b>
<b>6.3</b>	<b>SUGESTÕES PARA TRABALHOS FUTUROS</b>	<b>72</b>
<b><u>REFERÊNCIAS BIBLIOGRÁFICAS</u></b>		<b><u>73</u></b>
<b><u>APÊNDICE A: MOTOR DE PASSO</u></b>		<b><u>78</u></b>
<b>A.1</b>	<b>INTRODUÇÃO</b>	<b>78</b>
<b>A.2</b>	<b>FUNCIONAMENTO DOS MOTORES DE PASSO</b>	<b>79</b>
<b>A.3</b>	<b>CARACTERÍSTICAS DOS MOTORES DE PASSO</b>	<b>80</b>
<b>A.4</b>	<b>TIPOS DOS MOTORES DE PASSO</b>	<b>80</b>
<b>A.5</b>	<b>APLICAÇÕES DOS MOTORES DE PASSO</b>	<b>81</b>
<b>A.6</b>	<b>VANTAGENS E DESVANTAGENS DOS MOTORES DE PASSO</b>	<b>81</b>
<b><u>APÊNDICE B: INTERFACE CONTROLADORA DE MOTORES DE PASSO</u></b>		<b><u>82</u></b>
<b>B.1</b>	<b>INTRODUÇÃO</b>	<b>82</b>
<b>B.2</b>	<b>A INTERFACE CONTROLADORA COMPLETA.</b>	<b>82</b>
<b>B.3</b>	<b>INTERFACE PAINEL PARA MOTORES DE PASSO</b>	<b>83</b>

# LISTA DE FIGURAS

Figura 2.1 - As quatro anatomias básicas do robô .....	9
Figura 2.2 - Controle em Malha Fechada. ....	11
Figura 2.3 - Controle em Malha Aberta.....	12
Figura 2.4 - Robô hidráulico. HYD - 2800.....	13
Figura 2.5 - Robô elétrico. S400.....	14
Figura 2.6 - Robô pneumático. BOADICEA.....	15
Figura 3.1 - Interação de Agentes com o Ambiente Através de Sensores e Reagentes.....	20
Figura 3.2 - Arquitetura Funcionais .....	26
Figura 3.3 - Arquitetura de suposição.....	27
Figura 3.4 - Arquitetura “Touring machines”.....	28
Figura 4.1 - Representação de valores na lógica fuzzy.....	32
Figura 4.2 - Função de pertinência.....	33
Figura 4.3 - Representação gráfica da operação complemento. ....	34
Figura 4.4 - Representação gráfica da operação união. ....	35
Figura 4.5 - Representação gráfica da operação interseção.....	36
Figura 4.6 - Sistema de Controle Fuzzy.....	37
Figura 5.1 - Foto do ROBOXAD.....	42
Figura 5.2 - Esquema do ROBOXAD .....	42
Figura 5.3 - Eixo base .....	43

Figura 5.4 - Posição das estruturas .....	43
Figura 5.5 - Esquema básico do SI .....	43
Figura 5.6 - Casas simétricas na mesma linha .....	44
Figura 5.7 - Posicionamento dos elos L2 e L3 .....	46
Figura 5.8 - Movimento da garra sobre o plano .....	46
Figura 5.9 - Representação do posicionamento da garra nas posições 2R→2CR. ....	47
Figura 5.10 - Representação de Agentes.....	50
Figura 5.11 - Relações entre os componentes do MAUTOXAD .....	51
Figura 5.12 - Esquema Geral do MAUTOXAD.....	52
Figura 5.13 - Funções de pertinência.....	54
Figura 5.14 - Estratégia IDENTIFICA .....	57
Figura 5.15 - Estratégia MOVE .....	57
Figura 5.16 - Jogada Simples (S).....	59
Figura 5.17 - Jogada em L (L) .....	59
Figura 5.18 - Jogada em C (C) em Y .....	60
Figura 5.19 - Jogada em C (C) em X.....	60
Figura 5.20 - Esquema de Jogadas do MAUTOXAD .....	60
Figura 5.21 - Tela principal.....	61
Figura 5.22 - Divisão do tabuleiro e área de alcance.....	64
Figura 5.23 - Jogada sem obstáculo.....	64
Figura 5.24 - Início da Jogada com obstáculo .....	66

Figura 5.25 - Fim da Jogada com obstáculo .....	67
Figura 5.26 - Início da seqüência de movimento.....	68
Figura 5.27 - Fim da seqüência de movimento.....	68
Figura A.1 - Motor de passo .....	78
Figura A.2 - Motor genérico de quatro fases .....	79
Figura B.1 - Circuito impresso da interface de motor de passo.....	83
Figura B.2 - Circuito impresso do painel p/ motor de passo .....	84

## **LISTA DE TABELAS**

Tabela 5.1 - Coordenadas das posições 2R e 2CR.....	48
Tabela B.1 - Tabela de relação dos componentes utilizados na interface.....	83

## **LISTA DE QUADROS**

Quadro 5.1 - Quadro de regras de simetria .....	45
---	----

# RESUMO

Esta pesquisa constitui-se de um Sistema Inteligente desenvolvido para manipulação de um braço robótico. O sistema é formado por um braço robótico constituído por três elos, uma garra e por um tabuleiro de xadrez móvel, que auxilia o movimento do braço robótico. A abordagem utilizada na concepção do sistema foi orientada a Agentes Inteligentes. O sistema possui seis Agentes Inteligentes: o Agente Mestre que é responsável pela comunicação de seus subordinados; um Agente Inteligente para cada elo mecânico do braço; um Agente Tabuleiro que representa o tabuleiro móvel e um Agente Garra, responsável por pegar ou soltar as peças do tabuleiro. A inteligência de cada agente é implementada utilizando lógica Fuzzy, e o posicionamento do braço do robô é baseado na simetria das casas do tabuleiro de xadrez. A utilização de Agente Inteligente deixou as tarefas do sistema bem distribuídas e organizadas, possibilitando a representação qualitativa e quantitativa da posição dos braços do robô, permitindo o desenvolvimento de um sistema inteligente baseado em agentes para o controle do robô.

# ABSTRACT

This research concerns the construction of an intelligent system developed for the manipulation of a robotic arm. The system is composed of a robotic arm divided in three parts, a magnet and a movable chess board, designed to help the movement of the robotic arm. The system is agent oriented and has six agents: one master agent, responsible for the communication among the slave agents; one intelligent agent for each part of the arm; one board agent, representing the movable chess board; and one magnet agent responsible for catching and releasing the chess pieces. The intelligence of each agent is implemented using Fuzzy Logic and the positioning of the arm is based on the symmetry of the chess board's squares. The use of intelligent agents has let the tasks of the system well distributed and organized, making it possible to have a qualitative as well as a quantitative representation of the positioning of the robotic arm, and to develop an agent based intelligent system for the control of the robotic arm.

# *CAPÍTULO 1*

## **1. INTRODUÇÃO**

### **1.1 Sistemas Inteligentes**

O termo Inteligência Artificial foi proposto em 1956 por John MacCarthy da Universidade de Standford, nos Estados Unidos. Várias discussões para substituição deste nome por outros ocorreram, tais como Processamento Complexo da Informação ou Simulação de Processos Corrompidos, no entanto estes não lograram êxito [TAF95].

O objetivo inicial da Inteligência Artificial (IA) é entender os mecanismos da inteligência [OLI96]. Nas abordagens Clássicas de IA, o entendimento da inteligência é baseado em um comportamento humano individual, o foco de atenção volta-se para a representação de conhecimento e métodos de inferência. Já a Inteligência Artificial Distribuída (IAD) baseia-se na compreensão da inteligência e no comportamento social de entidades, denominadas agentes, enfatizando cooperações e interações entre elas, além do fluxo de conhecimento entre os mesmos.

A Inteligência Artificial (IA) engloba várias áreas do conhecimento, entre elas a Robótica. Um dos entraves na construção de sistemas robóticos está na sua complexidade, que advém da dependência que o projetista possui em relação a especialistas de domínios específicos.

O gerenciamento de sistemas robóticos através de microcomputadores envolve diversas áreas do conhecimento, tais como Engenharia Elétrica, Engenharia Mecânica, Computação, entre outras. Várias ferramentas e tecnologias foram propostas e estão



sendo utilizadas para se ter robôs mais autônomos e eficazes. Pode-se citar: processamento digital de imagem, aquisição de conhecimento, redes neurais artificiais, lógica fuzzy, algoritmos genéticos e controladores adaptativos. Essas são as principais razões que tornaram os sistemas robóticos amplamente utilizados por pesquisadores do mundo todo como plataformas de teste para novas técnicas e pesquisas científicas.

Atualmente uma das linhas mais importantes na área da robótica é o posicionamento de braços robóticos para execução de determinadas tarefas, por exemplo, o movimento de objetos. Essas tarefas devem ser executadas em um tempo determinado e com uma trajetória muito precisa. É por isso que se deseja do robô um comportamento inteligente durante a execução das tarefas. Um exemplo deste tipo de problema seria a manipulação de amostras (ampolas) em análises biomédicas. O grau de flexibilidade do robô e a percepção que ele possui do ambiente onde está inserido são de extrema importância para a realização de movimentos precisos e em tempo hábil. Esses robôs devem perceber o meio e modificar suas ações para executar tarefas específicas através de um comportamento inteligente.

Em robótica, geralmente são as ações que são modeladas a partir do conhecimento de um especialista, podendo-se utilizar controladores baseados em lógica Fuzzy (LF). Assim, torna-se uma forma de abordagem diferente dos métodos convencionais de controle de processos, que são desenvolvidos através da modelagem matemática.

Os Sistemas de Controle Inteligente devem ser capazes de estabelecer seus próprios objetivos baseados no reconhecimento da situação (percepção), e devem ser capazes de operar com incertezas e tomar decisões em ambientes não estruturados [POR89].

Um dos protótipos de robô desenvolvidos no Laboratório NEUROLAB, (Laboratório de Redes Neurais e Automação Inteligente), denominado ROBOXAD, sendo este o protótipo utilizado como base de teste para o desenvolvimento e implementação dos Sistemas Inteligentes (SI) para gerenciamento e controle de robôs [FER99][FER99a].

Um agente é uma entidade que percebe um ambiente e age sobre ele. Para tanto um agente necessita de sensores e efetadores [RUS95]. Através destes sensores e efetadores, pode-se criar um agente capaz de resolver um problema em favor de seu usuário. Um agente pode ou não ser dotado de inteligência. Quando a possui, ele é denominado Agente Inteligente.

Neste trabalho é utilizada a abordagem de Agente Inteligente para o desenvolvimento do Sistema Inteligente que controla a realização de movimentos em um braço robótico, utilizando-se Lógica Fuzzy.

## **1.2 Objetivos do Trabalho**

### **1.2.1 Objetivo Geral**

Utilizar a abordagem de Agente Inteligente para o posicionamento de um braço robótico com cinco graus de liberdade, capaz de se posicionar, mover, e desviar de obstáculos sobre um tabuleiro de xadrez.

### **1.2.2 Objetivos Específicos**

- Implementar um Sistema Inteligente baseado em agentes para o controle do posicionamento de um manipulador com cinco graus de liberdade (ROBOXAD);
- Utilizar a Lógica Fuzzy para otimizar o posicionamento do manipulador em tempo real;
- Ser capaz de desviar de obstáculos;
- Aumentar a área de alcance do manipulador;
- Testar seus movimentos.

## **1.3 Justificativa**

Este trabalho propõe uma abordagem em uma área importante da robótica, o sistema inteligente de controle do robô.

A utilização da abordagem de agente inteligente aplicado à robótica permitirá maior autonomia aos robôs. Isto, provavelmente diminuirá a complexidade dos modelos

atualmente empregados. Para o desvio de obstáculos, é bastante interessante o uso de agentes inteligentes, pois são várias as tarefas que serão executadas em um determinado espaço de tempo, o uso de agentes distribuídos executa as tarefas de uma forma organizada.

A Lógica Fuzzy sendo utilizada em cada um dos agentes será importante na otimização do posicionamento do braço em cada jogada a ser realizada.

As idéias aqui propostas podem ser utilizadas em outras aplicações semelhantes ao movimento de peças no tabuleiro de xadrez, por exemplo, na manipulação de ampolas em laboratórios de análises bioquímicas.

## **1.4 Estrutura da dissertação**

Esta dissertação está organizada em seis capítulos e 2 apêndices.

No Capítulo 2 apresenta-se uma breve descrição da Robótica, desde seus fundamentos, classificação geral e configuração de robôs. No mesmo capítulo define-se os manipuladores robóticos, sistemas de acionamentos, e comenta-se a utilização dos robôs no setor industrial.

No Capítulo 3 apresenta-se uma sucinta explanação de Agentes Inteligentes. São feitas definições, características gerais, tipos de ambiente onde os Agentes atuam, e as arquiteturas de agentes.

Apresenta-se no Capítulo 4 um breve estudo sobre a Lógica Fuzzy, descrevendo-se os seus fundamentos e destacando-se a definição de conjuntos fuzzy e a sua normalização. Apresenta-se também as principais operações com conjuntos fuzzy, a expressão fuzzy e de sistema de controle fuzzy. Finaliza-se o capítulo com um estudo de utilização desta lógica.

Finalmente, apresenta-se no Capítulo 5 o sistema de controle utilizando agentes inteligentes (MAUTOXAD). Descreve-se brevemente o ROBOXAD que é a base de teste do sistema, o sistema AUTOXAD e apresenta-se em detalhes a descrição do MAUTOXAD, apresentando seus agentes e sua implementação. Seu funcionamento é ilustrado através de alguns exemplos de movimentos (jogadas).

Conclui-se a dissertação no capítulo 6, no qual é feita uma análise das principais

vantagens obtidas com a utilização do agente inteligente e dificuldades encontradas. Encerra-se a conclusão com a apresentação de algumas sugestões para trabalhos futuros.

No Apêndice A, apresenta-se um breve resumo sobre motores de passos, como funcionam, principais características, os diferentes tipos de motores, suas aplicações, e algumas vantagens e desvantagens desses motores.

No Apêndice B, apresenta-se um breve resumo sobre a interface controladora dos motores de passos.

# CAPÍTULO 2

## 2. ROBÓTICA

### 2.1 Introdução

Muitos anos atrás, os robôs faziam parte apenas da ficção científica, fruto da imaginação do homem. No início dos anos 60, os primeiros robôs começaram a ser usadas com o objetivo de substituir o homem em tarefas que ele não podia realizar por envolverem condições desagradáveis, tipicamente contendo altos níveis de: calor; ruído; gases tóxicos; esforço físico extremo; trabalhos monótonos, "chatos".

Nos últimos 20 anos, as tendências que garantem a evolução dos robôs são: o constante aumento dos níveis salariais dos empregados; o extraordinário avanço tecnológico no ramo de computadores, que induz à redução dos preços do robô e uma significativa melhoria em seu desempenho.

A palavra robô ("robot") tem origem da palavra tcheca robotnik, que significa servo. O termo robô foi utilizado inicialmente por Karel Capek em 1923, nesta época a idéia de um "homem mecânico" parecia vir de alguma obra de ficção. [SAL90]

O grande escritor americano de ficção científica Isaac Asimov estabeleceu quatro leis muito simples para a robótica:

1ª lei: "Um robô não pode ferir um ser humano ou, permanecendo passivo, deixar um ser humano exposto ao perigo".

2ª lei: "O robô deve obedecer às ordens dadas pelos seres humanos, exceto se tais ordens estiverem em contradição com a primeira lei".

3ª lei: "Um robô deve proteger sua existência na medida em que essa proteção não estiver em contradição com a primeira e a segunda lei".

4ª lei: "Um robô não pode causar mal à humanidade nem permitir que ela própria o faça".

A quarta e última lei foi escrita por Asimov em 1984.[SAL90]

A idéia de se construir robôs começou a tomar força no início do século XX com a necessidade de aumentar a produtividade e melhorar a qualidade dos produtos. É nesta época que o robô industrial encontrou suas primeiras aplicações.

Atualmente devido aos inúmeros recursos que os sistemas de microcomputadores, os atuadores e os sensores nos oferece, a robótica atravessa uma época de contínuo crescimento que permitirá, em um curto espaço de tempo, o desenvolvimento de robôs inteligentes fazendo assim com que a ficção do homem antigo torne-se a realidade do homem atual.

## **2.2 O Que é Robótica ?**

Podemos definir como robótica o controle de mecanismos electro-electrônicos através de um computador, transformando-o em uma máquina capaz de interagir com o meio ambiente e executar ações decididas por um programa criado pelo programador a partir destas interações.[AMI93]

Podemos exemplificar o uso da robótica em diversas áreas de conhecimento. Na engenharia temos os robôs que mergulham a grandes profundidades para auxiliar em reparos nas plataformas de petróleo; na medicina, os robôs já auxiliam as cirurgias de alto risco. Outras aplicações podem ser menos percebidas, tal como a impressora que também é um robô.[SAL90]

## 2.3 Classificação Geral dos Robôs

Existem diversas classes de robôs que se diferenciam em suas aplicações e formas de trabalhar, possuindo diferentes características e propriedades. Os robôs podem ser classificados por tipo. Os quatro tipos principais de robôs são relacionados a seguir: [SAL90]

1. **Robôs Inteligentes:** são manipulados por sistemas multifuncionais controlados por computadores, são capazes de interagir com seu ambiente através de sensores e de tomar decisões em tempo real. Atualmente dedica-se grande esforço no desenvolvimento desse tipo de robô.
2. **Robôs com controle por computador:** são semelhantes aos robôs inteligentes, porém não tem a capacidade de interagir com o ambiente. Se estes robôs forem equipados com sensores e programas computacionais adequado, podem se transformarem em robôs inteligentes.
3. **Robôs de aprendizagem:** limitam-se a repetir uma seqüência de movimentos, realizados com a intervenção de um operador ou memorizados.
4. **Manipuladores:** são sistemas mecânicos multifuncionais, cujo sensível sistema de controle permite governar o movimento de seus membros das seguintes formas:
  - a) manual, quando o operador controla diretamente os movimentos;
  - b) de seqüência variável, quando é possível alterar algumas das características do ciclo de trabalho.

## 2.4 Configuração de um Robô

Existem quatro configurações básicas de robôs: Cartesiano, Cilíndrico, Polar e Articulado. Estas configurações estão ilustradas no diagrama da Figura 2.1 [SPO89].

Robôs polares, possuem duas juntas rotacionais e uma junta prismática. (ver Figura

2.1 (a)).

Os robôs cilíndricos podem ser representados por duas linhas perpendiculares e uma base de rotação. O braço do robô está ligado ao suporte para que possa ser movido radialmente em relação a coluna do robô. Um exemplo de configuração cilíndrica esta ilustrada na Figura 2.1 (b).

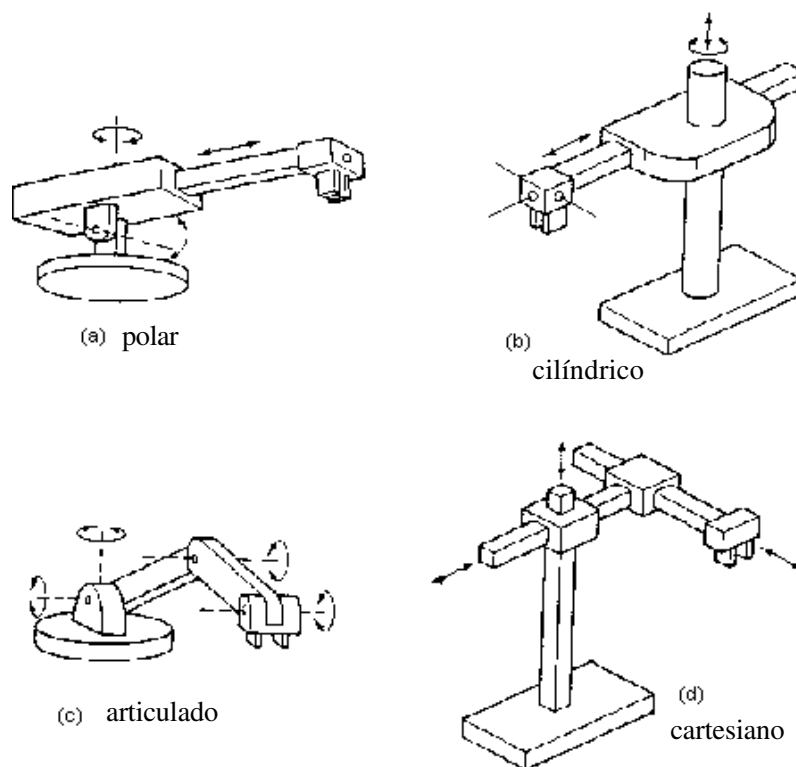


Figura 2.1 - As quatro anatomias básicas do robô

O robô articulado possui juntas que se movem similarmente aos movimentos das juntas de um ser humano. Consta de dois componentes retos, correspondendo ao antebraço e braço humano, ambos montados num pedestal vertical. Esses componentes estão conectados por juntas rotacionais correspondendo ao ombro e ao cotovelo. Um punho está unido à extremidade do antebraço, o que propicia diversas juntas adicionais (ver Figura 2.1 (c)).

Nos robôs cartesianos, os seus movimentos são realizadas junto a um eixo com três linhas perpendiculares representando os eixos xyz. A Figura 2.1 (d) ilustra este tipo de configuração.



## **2.5 Manipulador Robótico**

Os manipuladores robóticos possuem três partes principais: uma base fixa, um braço articulado, e uma unidade de controle. Os manipuladores robóticos são frequentemente descritos como tendo um certo número de graus de liberdade, que é o número de variáveis que podem ser modificadas de modo a alterar sua posição.

### **2.5.1 Base Fixa**

A base fixa consiste normalmente em um pedestal preso ao chão, mas que também pode estar presa às paredes ou ao teto, ou montada em outra máquina ou mesmo sobre uma plataforma móvel.

### **2.5.2 Braço Articulado**

O braço articulado é formado por várias partes: elos, juntas, atuadores de juntas, sensores de posição de juntas, punho e órgão terminal (a “mão” do robô).

- Elos são as partes rígidas de um braço de robô, comparáveis aos ossos do braço de uma pessoa.

- Juntas são as partes do braço de um robô que permitem uma conexão móvel entre dois elos. As juntas podem ser de dois tipos: deslizantes e rotativas. As juntas deslizantes movem-se em linha reta, sem girar. As juntas rotativas giram em torno de uma linha imaginária estacionária chamada eixo de rotação.

- Um atuador é uma versão mecânica de um músculo. Ele produz movimento quando recebe um sinal de entrada. Atuadores são chamados rotacionais ou lineares, dependendo se eles produzem um movimento giratório ou em linha reta. Os três principais tipos de atuadores são: eletromecânicos (acionados por motores elétricos), hidráulicos (acionados por líquidos comprimidos) e pneumáticos (acionados por gases comprimidos).

- Os sensores de posição de juntas são frequentemente chamados codificadores rotacionais ou lineares, porque codificam informações sobre as posições das juntas em uma

forma que pode ser facilmente enviada como sinais ao controlador do robô.

- O punho é o nome dado às três últimas juntas do robô. Estas são sempre rotacionais, e seus eixos de rotação são mutuamente perpendiculares.

- Os órgãos terminais podem ser classificados em dois grandes grupos denominados garras e ferramentas especializadas. Robôs usam garras para mover objetos e usam ferramentas especializadas para fazer tarefas especiais.

### 2.5.3 Unidade de Controle

A unidade de controle representa o "cérebro" do robô. Ela recebe sinais de entrada dos sensores do robô e transmite sinais de saída para os atuadores do robô. Há dois tipos de sistemas de controle de robôs: malha aberta e malha fechada. No sistema de malha fechada, depois que o controlador envia sinais ao atuador para mover o manipulador, um sensor no manipulador retorna um sinal ao controlador, fechando a malha controlador-atuador-manipulador-sensor-controlador. A Figura 2.2 mostra um esquema de controle do tipo malha fechada.

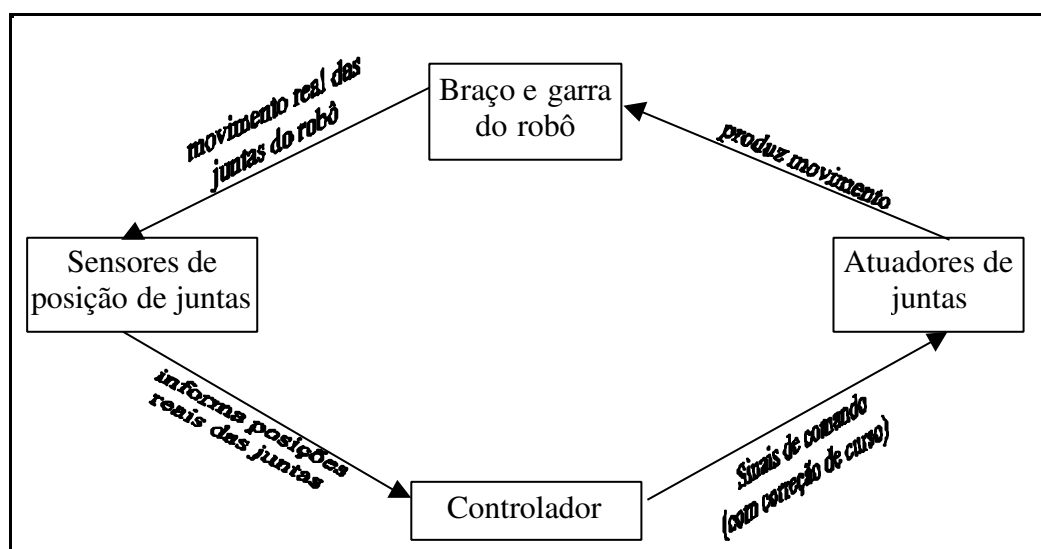


Figura 2.2 - Controle em Malha Fechada.

No esquema do tipo malha aberta da Figura 2.3, não há sensor medindo como o manipulador realmente se moveu em resposta aos sinais enviados para os atuadores, e consequentemente não há sinais de realimentação do manipulador para o controlador. A

“malha de controle” é aberta, indo do controlador para o atuador e deste para o manipulador. Portanto não há maneira de se saber a posição atual do manipulador. Tudo o que se sabe é onde ele deveria estar e não se ele realmente alcançou o alvo.

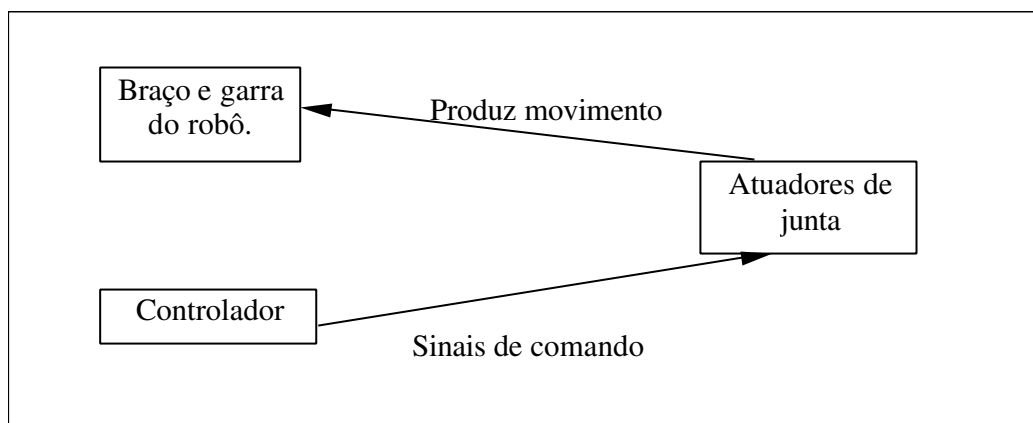


Figura 2.3 - Controle em Malha Aberta.

## 2.6 Sistemas de acionamento de robôs

A capacidade do robô de mover o seu corpo, braço e punho é determinada pelo sistema utilizado para acioná-lo. O sistema de acionamento determina a velocidade dos movimentos do braço, a força do robô e seu desempenho dinâmico. Em certa medida, o sistema de acionamento determina os tipos de aplicação que o robô pode realizar.

Os tipos de sistemas de acionamento podem ser divididos em:

### 2.6.1 Acionamento Hidráulico

O uso de acionamento hidráulico é geralmente associado a robôs de maior porte. Os robôs hidráulicos foram a fonte original de força nos primeiros robôs [AMI93]. Este sistema propicia ao robô maior velocidade e força; porém, ele se soma ao espaço útil do piso requerido pelo robô. O sistema de acionamento hidráulico pode ser projetado para acionar juntas rotacionais ou lineares. Acionadores de pás ou aletas rotativas podem ser utilizados para suprir movimento rotacional, e os pistões hidráulicos podem ser utilizados

para realizarem movimento linear.

Um exemplo de robô hidráulico é o HYD-2800 ilustrado na Figura 2.4. É um robô hidráulico utilizado em laboratórios por estudantes para simular uma aplicação industrial na vida real. Os robôs hidráulicos tendem a vazarem óleo, o que é um inconveniente.

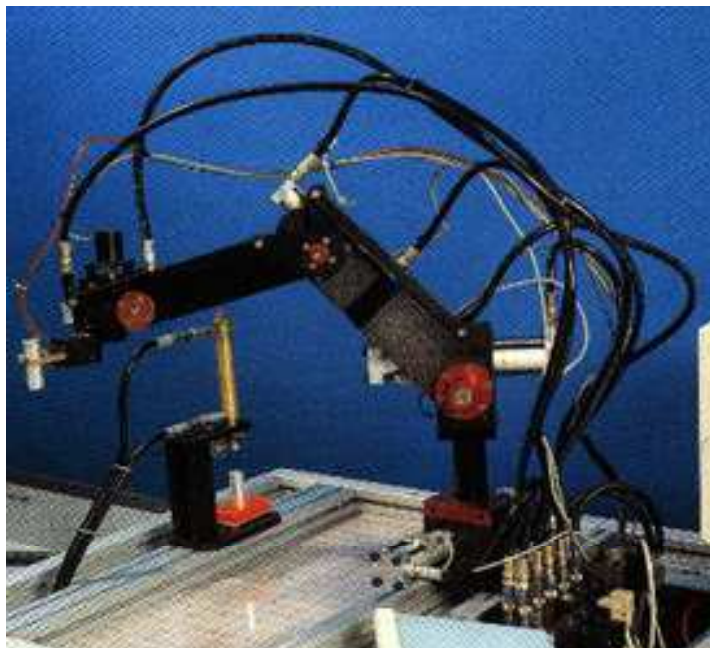


Figura 2.4 - Robô hidráulico. HYD - 2800

## 2.6.2 Acionamento Elétrico

O acionamento elétrico geralmente não propicia tanta velocidade ou potência quanto os sistemas hidráulicos. Todavia, a precisão e repetibilidade dos robôs de acionamento elétrico são geralmente melhores. Conseqüentemente, os robôs elétricos tendem a ser menores, requerendo menos espaço útil no piso, e suas aplicações tendem para um trabalho mais preciso como, por exemplo, montagem.

Os robôs de acionamento elétrico são acionados por motores de passo ou servo motores de corrente contínua. Estes motores são adequados idealmente para o acionamento de juntas rotacionais através de sistemas de eixos e engrenagens de acionamento.

Os motores elétricos podem também ser utilizados para acionar juntas lineares (por exemplo, braços telescópicos) por meio de sistemas de polia ou por outros mecanismos

translacionais.

Um exemplo de robô elétrico é o S400 ilustrado na Figura 2.5. O S400 é um robô articulado produzido pela G. M. Manuc, com seis eixos, que utiliza servos motores. Este robô é utilizado por empresas para soldagem a ponto e a arco, para carga e descarga e outros.



Figura 2.5 - Robô elétrico. S400

### **2.6.3 Acionamento Pneumático**

Este tipo de acionamento é geralmente reservado para robôs de menor porte com poucos graus de liberdade (movimentos de duas a quatro juntas). Estes robôs estão freqüentemente limitados a simples operações de "pega-e-põe" com ciclos rápidos. O acionamento pneumático pode ser facilmente adaptado ao acionamento de pistão, para suprir movimento de translação das juntas de deslizamento. Pode também ser utilizado para operar órgãos terminais rotacionais para juntas rotacionais.

Na Figura 2.6 está ilustrado o robô BOADICEA. Ele é um tipo de robô pneumático. O BOADICEA possui seis pernas, sua estrutura mecânica foi construída com fibra de carbono, tubos de alumínio e tubos de plástico de injeção.

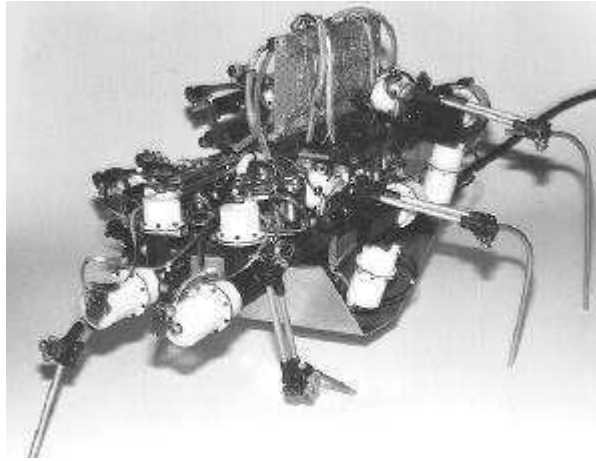


Figura 2.6 - Robô pneumático. BOADICEA

## 2.7 Aplicação Robótica na Indústria

Na época em que foram lançados no mercado, na década de 1960, os robôs eram caros e acessíveis a pouquíssimas empresas existentes em países mais desenvolvidos, principalmente no Japão e nos Estados Unidos. No entanto, a partir de 1976 os seus preços começaram a baixar de uma forma acelerada.

O grande responsável por esta brutal redução de custos que ocorreu na informática e na robótica, foi a microeletrônica. Com o avanço desta disciplina, por exemplo, foi possível colocar toda a capacidade do ENIAC, o primeiro computador construído à válvula e desenvolvido em 1950, em uma pastilha de silício de menos de 0,5 cm<sup>2</sup>. Ressaltando que isso se consegue com velocidade de processamento muito superior e a um custo infinitamente menor.

Desta forma, os microprocessadores, influenciaram diretamente a capacidade de todas as máquinas industriais, tendo impacto decisivo nas tecnologias associadas à robótica, permitindo que a capacidade de processamento de informações se multiplicasse de forma estrondosa, além de baratear o custo dos robôs, tornando-os mais acessíveis.

Este crescimento de tecnologia relacionados a robótica gerou grandes benefícios. A automação possibilita grandes incrementos na produtividade do trabalho, possibilitando que as necessidades básicas da população possam ser atendidas. Além de aumentar a produção, os equipamentos automatizados possibilitam uma melhora na qualidade do produto,

uniformizando a produção, eliminando perdas e refugos.

A automação também permite a eliminação de tempos mortos, ou seja, permite a existência de "operários" que trabalhem 24 horas por dia sem reclamarem, o que leva a um grande crescimento na rentabilidade dos investimentos.

A microeletrônica flexibiliza o processo de fabricação, ou seja, permite que os produtos sejam produzidos conforme as tendências do mercado, evitando que se produzam estoques de produtos invendáveis.

As características citadas acima, mostram que a microeletrônica, possibilita que não haja nem escassez nem desperdício, com melhor qualidade de vida e de produção, aliada a um menor esforço. Sem dúvida a automação industrial foi e é um grande impulsionador da tecnologia de robótica.

Cada vez mais tem se procurado aperfeiçoar os dispositivos, dotando-os com inteligência para executar as tarefas necessárias. Por exemplo, usando Redes Neurais procura-se a linearização de acionamentos eletromecânicos; com Lógica Fuzzy pode-se fazer o planejamento de trajetória para robôs redundantes; ou utilizando Sistemas Especialistas é possível a detecção de vazamento de água a partir da aquisição remota de consumo.

Fala-se em evolução dos robôs, mas não se pode esquecer dos impactos sociais que eles podem causar a sociedade. E quando se fala em impactos causados pela robótica o primeiro fator que nos vem a cabeça é o desemprego.

As transformações que ocorrem, causadas pelo advento dos robôs, muitas vezes podem não estar visíveis para grande parte das pessoas que não convivem no ambiente fabril, contudo a ascensão da robótica nas fábricas faz parte da mesma tendência que vem determinando, nos últimos anos, a crescente automatização dos bancos, do comércio e das empresas em geral, causados pelo advento da informática.

No que se refere ao meio fabril pode-se observar dois lados de uma moeda: Por um lado, as indústrias recrutam robôs e computadores. Elas são guiadas por uma necessidade crucial para sobrevivência no mercado de forma a conquistar maior produtividade e qualidade para seus produtos de forma barata e assim assegurar competitividade frente aos

concorrentes.

Por outro lado os trabalhadores, ficam aterrorizados com a possibilidade de perda de emprego, causados pelos impactos que os robôs exercem sobre o nível de emprego. Certamente os robôs se instalam no lugar dos homens, muitas vezes, um robô substitui dezenas ou até centenas de homens em uma linha de produção.

Este temor de desemprego vem aumentando a cada dia que passa. A queda nos custos dos robôs tornando-os acessíveis para muitos setores das indústrias, fez com que eles (os robôs) pudessem competir com a mão de obra barata, como a existente nos países do terceiro mundo, ameaçando o emprego de muitos trabalhadores.

Muitas empresas multinacionais, que se instalavam em países subdesenvolvidos para utilizar-se do recurso "mão de obra barata", já estão pensando em reverter essa tendência e concentrar suas operações nos seus próprios países de origem, utilizando robôs para baratear seus custos.

O uso de robôs para as indústrias passa a ser uma questão de sobrevivência, assim, resistir ao seu uso dos robôs é uma batalha perdida, principalmente devido a forma acelerada com que eles caem de preço. Além disso, o sucesso que as empresas e países usuários de robôs vem obtendo é alto. O Japão por exemplo, em 10 anos conseguiu quadruplicar a sua produção de automóveis, mantendo praticamente a mesma força de trabalho.

## **2.8 Conclusão**

Assim como a Revolução Industrial, o desenvolvimento de robôs modernos prevê uma revolução na vida das pessoas. As máquinas serão responsáveis por trabalhos e serviços em diversas áreas, permitindo ao homem dedicar-se a pesquisa científica, hobbies e lazer.



# *CAPÍTULO 3*

## **3. AGENTE INTELIGENTE**

### **3.1 Introdução**

Durante os anos 80 a comunidade de Inteligência Artificial, desencorajada pela falta de progresso, após 30 anos de pesquisa em projetos de sistema inteligentes começou a explorar novas áreas onde sistemas de IA pudessem ter um domínio mais dinâmico de aprendizagem. Ao invés de olhar para resultados simulados, simbólicos em mundos artificiais, começaram a explorar as possibilidades de interações complexas com o mundo físico, através de um mecanismo denominado agentes.[OLI96]

A tecnologia de agentes inteligentes é uma das áreas de maior crescimento de pesquisa e desenvolvimento. Ela proporciona ao usuário facilidades que são baseadas em conceitos da Inteligência Artificial Distribuída (IAD), desenvolvidos a partir de métodos de representação de conhecimentos, de resolução de problemas e de inferência em ambientes distribuídos.[OLI96] A IAD estuda os comportamentos sociais, onde os sistemas computacionais são vistos como sociedades de agentes inteligentes e tendo como ênfase as cooperações, interações e o fluxo de conhecimento entre unidades distintas.

As aplicações e/ou investigações das tecnologias de agentes incluem administração de rede, controle de tráfego aéreo, controle de robôs, recuperação/ administração de informação, comércio eletrônico, educação, assistente digital pessoal, elaboração de agenda, organização de “e-mail”, etc.

Uma das definições de agentes refere-se a eles como entidades reais ou virtuais que emergem num ambiente onde podem tomar algumas ações. Eles capazes de perceber e representar parcialmente esse ambiente, de comunicar-se com outros agentes e possuem um comportamento autônomo, consequência de sua observação, seu conhecimento e suas interações com outros agentes.[FER91]

“Embora não haja ainda um consenso sobre uma definição formal da idéia de agente inteligente, pode-se defini-lo agente inteligente como uma entidade cognitiva, ativa e autônoma, ou seja, que possui um sistema interno de tomada de decisões, que age sobre o mundo e sobre os outros agentes que o rodeiam e, por fim, que é capaz de funcionar sem necessitar de algo ou de alguém para o guiar (tem mecanismos próprios de percepção do exterior)”. [MAE94]

Quando diversos agentes inteligentes estão associados pela interação num ambiente comum, este ambiente é chamado multi-agente. “Em Sistemas multi-agentes, o projetista não volta sua atenção para um problema específico, mas para um domínio específico. Nesta abordagem, a idéia consiste em coordenar o comportamento inteligente de um conjunto de agentes autônomos, cuja existência pode ser anterior ao surgimento de um problema particular” [OLI96].

## **3.2 Agentes**

### **3.2.1 Definições**

Pode-se definir um agente como sendo qualquer coisa que percebe seu ambiente através de sensores e atua neste ambiente através de reagentes ou efetadores [RUS95] (Figura 3.1). Agente inteligente refere-se a uma entidade inteligente e autônoma. A palavra autônoma, neste caso, significa que cada agente possui sua própria existência, a qual não é dependente da existência de outros agentes .

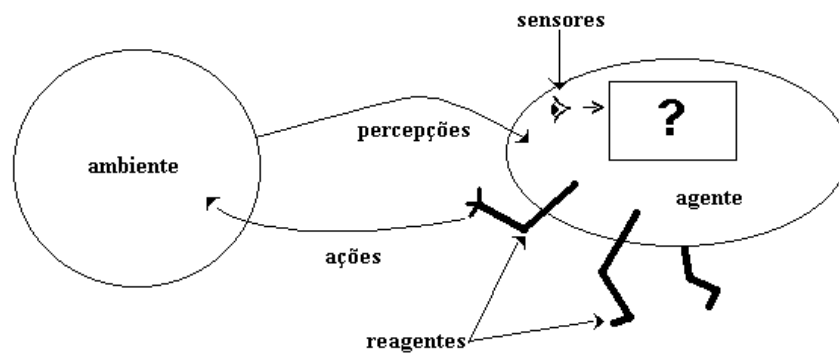


Figura 3.1 - Interação de Agentes com o Ambiente Através de Sensores e Reagentes

Todas as propriedades que um agente apresenta devem estar implícitas em sua arquitetura. A arquitetura de um agente pode ser definida como uma metodologia particular para definir agentes ou simplesmente como a porção de um sistema que provê e gerencia recursos de um agente.

A escolha por uma determinada arquitetura é freqüentemente feita seguindo alguma declaração metodológica explícita, freqüentemente direcionada para o domínio e ambiente no qual a arquitetura será usada. A seguir apresentam-se algumas definições de arquiteturas de agentes :

“é uma metodologia particular para a construção de agentes. Ela especifica como o agente pode ser decomposto na construção de um conjunto de módulos componentes e como esses módulos devem ser construídos para interagir. O conjunto total de módulos e suas interações têm que fornecer uma resposta a questão de como o sensor de dados e o estado interno atual do agente determinam suas ações. Uma arquitetura abrange técnicas e algoritmos que suportam essa metodologia.” [MAE91]

“é uma coleção específica de módulos de programas computacionais (ou computadores), tipicamente designado por caixas com setas indicando o controle e fluxo de dados entre os módulos. Uma visão abstrata de uma arquitetura é como uma metodologia para desenhar decomposições particulares para tarefas particulares.” [WOO94]

### 3.2.2 Características Gerais de Agentes Inteligentes

Alguns atributos que caracterizam os agentes são os seguintes:

**Autonomia:** autonomia é a capacidade do agente executar o controle sobre suas próprias ações [FRA96]. Nissen [NIS95] relaciona o controle somente ao usuário final quando define que, um agente inteligente deve possuir a habilidade de praticar ações para desenvolver tarefas ou alcançar objetivos, sem necessitar da interferência do usuário final. Wooldridge e Jennings [WOO94] acrescentam que o agente, além de possuir controle sobre seu comportamento, deve também possuir controle sobre seu estado interno.

**Mobilidade:** a característica de mobilidade é especificada por Franklin e Graesser [FRA96] como a capacidade do agente de transportar-se de uma máquina à outra.

**Comunicabilidade:** segundo Franklin e Graesser [FRA96], a comunicabilidade é a capacidade do agente de comunicar-se com os outros agentes ou pessoas.

**Inteligência:** Auer [AUE95] identifica a inteligência como a propriedade de um agente que o habilita a negociar efetivamente com ambigüidades. Durante o processo de determinação da ação mais adequada à situação, o agente defronta-se com ambigüidades nos mais diversos níveis. Neste contexto, a inteligência pode ser considerada como um conjunto de recursos, atributos e características que habilitam o agente a decidir que ações executar.

**Reatividade:** Wooldridge e Jennings [WOO94] definem reatividade como sendo a propriedade que permite aos agentes perceberem seus ambientes e responderem adequadamente às mudanças neles ocorridas.

**Habilidade Social:** Paraiso [PAR96] descreve esta característica como a habilidade que os agentes possuem de interagir com os outros agentes ou pessoas, no momento adequado, para concluir suas tarefas ou ajudar outros agentes.

**Flexibilidade:** tendo em vista as funções executadas, a flexibilidade é fundamental, visto que não são preestabelecidos roteiros para o desenvolvimento de suas atividades Franklin e Graesser [FRA96]. Auer [AUE95] complementa afirmando que a flexibilidade

reside na habilidade dos agentes de escolher dinamicamente as ações e a seqüência de execução das mesmas, em resposta a um estado do ambiente.

**Cooperatividade:** Gilbert e Manny [GIL96] afirma que cooperatividade é a capacidade dos agentes inteligentes trabalharem juntos para concluírem tarefas mutuamente benéficas e complexas. Para tanto, os agentes devem possuir "espírito de colaboração" a fim de criarem e obterem êxito nos sistemas orientados a agentes.

**Comportamento adaptativo:** a capacidade do agente de modificar seu comportamento em função de experiências anteriores é chamada por Franklin e Graesser [FRA96] de Comportamento Adaptativo. Segundo Gilbert e Manny [GIL96], os agentes devem ser capazes de examinar o ambiente externo bem como os procedimentos efetuados anteriormente sob condições similares, e adaptar então suas ações, objetivando aumentar a probabilidade de alcançarem seus objetivos.

**Aprendizagem:** Belgrave [BEL95] relaciona as propriedades de aprendizado e comportamento adaptativo e as define como, a habilidade apresentada pelo agente de acumular conhecimento baseado em experiências anteriores, e conseqüentemente, modificar seu comportamento em resposta à novas situações. Segundo Auer [AUE95], aprendizagem é a capacidade que um agente deve possuir para executar uma tarefa com maior eficiência do que em execuções anteriores. Sem a capacidade de aprendizagem o agente reagirá sempre da mesma maneira para um mesmo ambiente e uma mesma situação.

**Coerência:** Lemon [LEM96] define coerência como sendo a propriedade que o agente possui de resolver conflitos entre objetivos concorrentes ou conflitantes. O Agente é dito coerente se mantém um comportamento íntegro, mesmo quando inserido em situações ambientais que indiquem a possibilidade de diferentes respostas adequadas.

**Planejamento:** Belgrave [BEL95] afirma que planejamento é a habilidade do agente de sintetizar e escolher diferentes cursos de ações, com o propósito de alcançar seus objetivos.

### 3.2.3 Ambiente

Sociedade é o conjunto de entidades ativas, os agentes. Já as entidades passivas formam o ambiente. Um agente raciocina sobre os outros agentes e o ambiente. Comparando com a área de Sistemas Distribuídos, os agentes são os processos, a sociedade é o conjunto de processos e o ambiente corresponde às entidades do mundo, com exceção dos processos, com os quais a sociedade mantém relação.[ALV97]

Quando se diz que um agente está inserido num determinado ambiente é no sentido que ele se encontra exposto a interações com ele, por exemplo: um robô numa fábrica, um agente de software que negocia na Internet, entre outros.

Esta interação é essencial pois é determinante na definição e na atuação de um agente (se a sua existência fosse autônoma até em relação aos seus objetivos, um agente poderia ser visto como uma caixa preta, sem qualquer utilidade).

Assim, as interações com o ambiente são essenciais para um agente: permitem alterar o ambiente de forma e obter informações que podem ser úteis ao cumprimento do seu objetivo (incluindo a avaliação do seu próprio desempenho).

#### **Propriedades de ambientes**

Ambientes aparecem em várias formas [RUS95]. As principais distinções a serem feitas são as listadas a seguir:

**Acessível versus inacessível:** Se os sensores de um agente dão acesso ao estado completo do seu ambiente, então dizemos que este ambiente é acessível ao agente. Um ambiente é efetivamente acessível se os sensores detectam todos os aspectos relevantes à escolha da ação apropriada.

Um ambiente acessível é conveniente porque o agente não precisa manter qualquer estado interno para ter informação sobre o mundo.

**Determinístico versus não determinístico:** Se o próximo estado do ambiente é determinado completamente pelo estado atual e as ações selecionadas pelos agentes, dizemos que o ambiente é determinístico. Em princípio, o agente não precisa se preocupar

com a incerteza em um ambiente acessível e determinístico.

Se o ambiente é inacessível, ele pode parecer não determinístico. Isso se torna particularmente verdade se o ambiente é complexo, tornando-se difícil controlar todos os aspectos não acessíveis. Assim, é sempre melhor pensar num ambiente como determinístico ou não determinístico do ponto de vista do agente.

**Episódico versus não episódico** – Num ambiente episódico, a experiência do agente está dividida em “episódios”. Cada episódio consiste na percepção e conseqüente ação do agente. A qualidade da ação depende apenas do episódio em si, porque episódios subsequentes não dependem de que ações ocorram em episódios anteriores. Ambientes episódicos são muito mais simples porque os agentes não precisam pensar adiante.

**Estático versus dinâmico** – Se o ambiente pode mudar enquanto o agente estiver atuando, então dizemos que o ambiente é dinâmico para o agente, do contrário ele é estático. Ambientes estáticos são fáceis de se tratar porque o agente não precisa ficar observando o mundo enquanto toma uma decisão sobre uma ação, nem precisa se preocupar com a passagem do tempo.

Se o ambiente não muda com o passar do tempo, mas a medida de desempenho do agente sim, dizemos que este ambiente é semidinâmico.

**Discreto versus contínuo** – Se existe um número distinto, definido claramente de estímulos e ações, dizemos que o ambiente é discreto. Xadrez é discreto – existe um número fixo de movimentos possíveis a cada jogada. Dirigir um táxi é contínuo – a velocidade e distância entre outros veículos variam entre uma faixa de valores contínuos.

### 3.2.4 Arquitetura de um agente

Outro aspecto importante num agente é a sua arquitetura interna. A arquitetura interna de um agente está associada à própria definição e mecanismos de decisão do agente que determinam e influenciam a sua atuação.

As arquiteturas dos agentes podem ser divididas em cognitivas, reativas e híbridas [WOO96].

### 3.2.4.1 Arquiteturas cognitivas

Os agentes cognitivos já têm conhecimento do estado do mundo onde estão inseridos; têm metas próprias a atingir. Estes agentes baseiam-se nesse conhecimento para decidirem que ações vão realizar, isto é, estes agentes têm controle sobre o seu próprio comportamento e não se limitam a reagir a estímulos exteriores.

Possuem uma representação simbólica de si próprios (as suas capacidades e os seus objetivos), bem como dos outros agentes e do estado do mundo, o que lhes permite operando sobre essas representações, deduzir ações a tomar no sentido de maximizar uma utilidade.

No contexto de um Sistema Multi-Agente, os agentes sabem que estão inseridos numa comunidade de agentes, em que especificamente cada um deles tem objetivos e capacidades próprias diferentes (ou não), mas todos cooperam entre si, na tentativa de alcançarem a resolução de um determinado problema.

Os agentes têm uma racionalidade social visto que, estando inseridos numa comunidade, podem ter para além de objetivos principais, outros objetivos de resolução de um problema global.

Eles também têm uma racionalidade individual uma vez que possuem conhecimentos específico-se e objetivos individuais a atingirem que não beneficiam o resto da comunidade. Um agente pode possuir diferentes estruturas de conhecimento como fatos, crenças, objetivos ou intenções, preferências, motivos, desejos, etc.

Normalmente um agente adquire novos fatos ou crenças, como resultado do envio de mensagens de outros agentes ou, através de modificações do ambiente onde ele está inserido. Os agentes não só executam ações, como resultado das percepções, mas também essas ações podem ser o resultado dos objetivos por ele estabelecidos.

Segundo Demazeau [DEM95] a arquitetura do agente cognitivo possui: conhecimento, percepção, comunicação, raciocínio e decisão ( Figura 3.2).

Um exemplo da arquitetura cognitiva é a arquitetura Funcional [OLI96]. Nas arquiteturas funcionais o agente é dividido em módulos que implementam as



funcionalidades consideradas necessárias a sua operação.

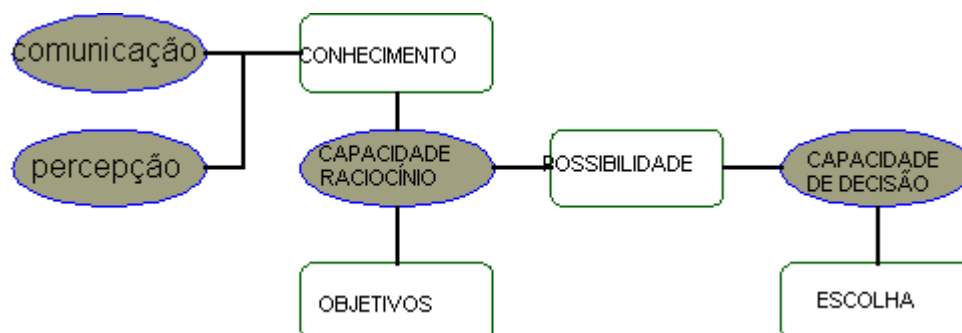


Figura 3.2 - Arquitetura Funcionais

### 3.2.4.2 Arquiteturas Reativas

Os modelos reativos encontram-se no lado oposto dos agentes cognitivos. O seu mecanismo de "inferência" é simplificado ao máximo e especializado na resposta a determinados "estímulos" do contexto. Rodney Brooks [BRO91], um defensor desta abordagem sintetiza o apoio à abordagem reativa:

- melhor modelo do mundo é o próprio mundo.
- comportamentos inteligentes emergem da dinâmica das interações do agente com contexto.

Assim, a ênfase é colocada não na definição de modelos abstratos do mundo, mas em definir capacidades simples e úteis para os agentes. A arquitetura baseia-se no comportamento realização e tarefas, como mecanismos simples de reação dos agentes [WOO95].

A base desta visão é que se espera que pela conjugação destas reações e interações com o exterior possam nascer "abordagens" inteligentes por parte dos agentes. No âmbito deste tipo de arquiteturas surgiu uma corrente que introduz a noção de comportamento.

A noção de comportamento pode ser vista como um mecanismo de controle que permite alcançar ou manter dado objetivo ao nível do agente. Os comportamentos podem

envolver a conjugação de várias “reações” dos agentes na sua definição.

Nesta linha lógica, Brooks [BRO91] sugere uma arquitetura de suposição, como base para a organização funcional dos comportamentos. A idéia base é que inerente a dado agente reativo está associado um esquema de combinação dos comportamentos que permite descrever prioridades, ações conjuntas, etc (Figura 3.3).

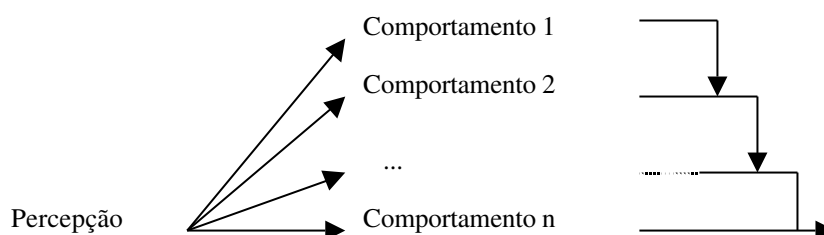


Figura 3.3 - Arquitetura de suposição

No extremo das abordagens reativas estão as abordagens puramente reativas (conhecidas também por reflexivas) onde a idéia base consiste em um agente possuir um conjunto de ações pré-programadas com as quais reage a estímulos bem definidos, sem qualquer mecanismo de "análise" (à semelhança dos reflexos humanos).

### 3.2.4.3 Arquiteturas Híbridas

Os modelos híbridos estruturam-se em níveis [WOO96]. Nível essencialmente reativo e nível mais cognitivo (planejamento e "inferência" sobre os conhecimentos de contexto).

A idéia básica é que as tarefas que asseguram a sobrevivência do agente têm uma estrutura mais reativa e todo mecanismo cognitivo está encarregado de delinear as estratégias para alcançar os objetivos do agente.

A arquitetura híbrida mistura componentes das arquiteturas deliberativas e reativas com o objetivo de torná-la mais adequada e funcional para a construção de agentes. Nesse caso, os agentes são dotados de comportamento reativo com relação aos eventos que ocorrem no ambiente e comportamento deliberativo onde existe uma definição simbólica do mundo para a tomada de decisões.

Exemplo da arquitetura híbrida é a arquitetura “Touring machines” (Figura 3.4) formada por três camadas além do subsistema de percepção e ação que agem diretamente com o ambiente do agente. As camadas são as seguintes: reativa, de planejamento e de modelagem. [WOO95]

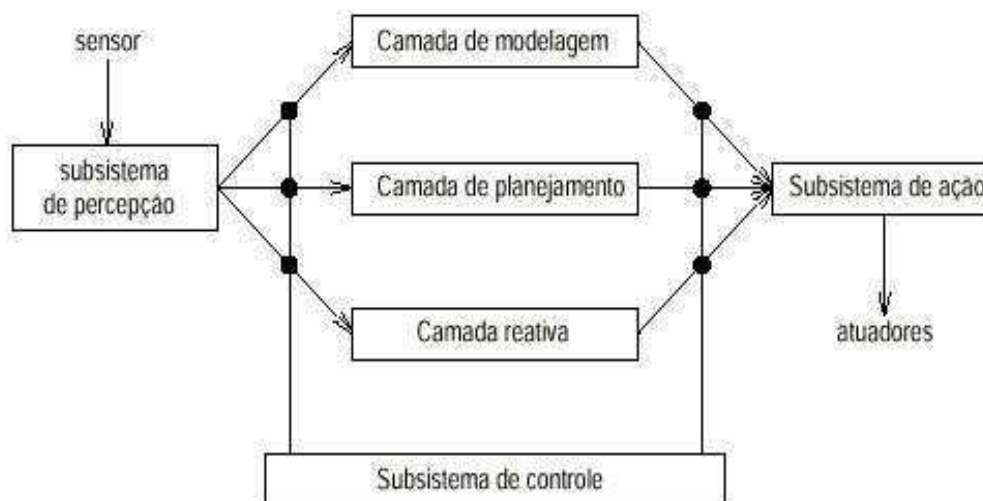


Figura 3.4 - Arquitetura “Touring machines”

A camada reativa é implementada com um conjunto de regras de situação-ação, onde a partir dos eventos que ocorrem no ambiente executam-se ações potenciais.

A camada de planejamento constrói planos e seleciona ações para executar, a fim de alcançar os objetivos do agente. Possui dois componentes, o planejador e o mecanismo de foco de atenção.

O primeiro integra a geração e execução dos planos e usa uma biblioteca de planos juntamente com um mapa ou visão do mundo, a fim de construir planos que executarão na busca dos objetivos do agente.

O último tem o papel de limitar a quantidade exagerada de informações para o planejador, fazendo com que o mesmo possa tratar somente as informações que lhe competem.

Por fim, a camada de modelagem, contém modelos ou representações dos estados cognitivos de outras entidades no ambiente dos agentes. Esses modelos são manipulados para identificar conflitos nos objetivos em virtude do aparecimento de uma interferência

quando o agente os está percorrendo.

### **3.3 Conclusão**

Este Capítulo apresentou uma visão geral sobre agentes inteligentes. Foram vistos , as diversos conceitos dados para o termo agente, suas principais características, o ambiente onde atuam, assim como as propriedades desse ambiente. Finalizamos apresentando os três diferentes tipos de arquitetura dos agentes: cognitiva, reativa, e híbrida. Este estudo possibilitou a definição dos agentes do nosso sistema.

# CAPÍTULO 4

## 4. LÓGICA FUZZY

### 4.1 Introdução

Aristóteles, filósofo grego (384 - 322 a.C.), foi o fundador da ciência da lógica, e estabeleceu um conjunto de regras rígidas para que conclusões pudessem ser aceitas como logicamente válidas. O emprego da lógica de Aristóteles levava a uma linha de raciocínio lógico baseado em premissas e conclusões. Como um exemplo: se é observado que "todo ser vivo é mortal" (premissa 1), a seguir é constatado que "Sarah é um ser vivo" (premissa 2), como conclusão temos que "Sarah é mortal".

Desde então, a lógica Ocidental, assim chamada, tem sido binária, isto é, uma declaração é falsa ou verdadeira, não podendo ser ao mesmo tempo parcialmente verdadeira e parcialmente falsa. Esta suposição e a lei da não contradição, que coloca que "U e não U" cobrem todas as possibilidades, formam a base do pensamento lógico Ocidental. A lógica fuzzy viola estas suposições.

A lógica de Aristóteles trata com valores "verdade" das afirmações, classificando-as como verdadeiras ou falsas. Não obstante, muitas das experiências humanas não podem ser classificadas simplesmente como verdadeiras ou falsas, sim ou não, branco ou preto.

Por exemplo, é aquele homem alto ou baixo? A taxa de risco para aquele empreendimento é grande ou pequena? Um sim ou um não como resposta a estas questões é, na maioria das vezes, incompleta.

Na verdade, entre a certeza de ser e a certeza de não ser, existem infinitos graus de incerteza [BEZ94]. Esta imperfeição intrínseca à informação representada numa linguagem natural tem sido tratada matematicamente no passado com o uso da teoria das probabilidades. Contudo, a lógica fuzzy, com base na teoria dos conjuntos fuzzy, tem se mostrado mais adequada para tratar imperfeições da informação do que a teoria das probabilidades[BEZ94].

De forma mais objetiva e preliminar, podemos definir lógica fuzzy como sendo uma ferramenta capaz de capturar informações vagas, em geral descritas em uma linguagem natural, e convertê-las para um formato numérico, de fácil manipulação pelos computadores de hoje em dia. Considere a seguinte afirmativa: se o tempo de um investimento é longo e o sistema financeiro tem sido não muito estável, então a taxa de risco do investimento é muito alta. Os termos "longo", "não muito estável" e "muito alta" trazem consigo informações vagas. A extração (representação) destas informações vagas se dá através do uso de conjuntos fuzzy.

Devido a esta propriedade e a capacidade de realizar inferências, a lógica fuzzy tem encontrado grandes aplicações nas áreas: sistemas especialistas, computação com palavras, raciocínio aproximado, linguagem natural, controle de processos, robótica, modelamento de sistemas parcialmente aberto, reconhecimento de padrões, processos de tomada de decisão.

O conceito de conjunto Fuzzy foi introduzido, em 1965, por Lotfi A. Zadeh da Universidade da Califórnia, Berkeley [ZAD65]. No meio da década de 60 do século XX, Zadeh observou que os recursos tecnológicos disponíveis eram incapazes de automatizar as atividades relacionadas a problemas de natureza industrial, biológica ou química, que compreendessem situações ambíguas, não passíveis de processamento através da lógica computacional fundamentada na lógica booleana. Procurando solucionar esses problemas o Professor Zadeh publicou em 1965 um artigo resumindo os conceitos dos conjuntos Fuzzy, revolucionando o assunto com a criação de sistemas Fuzzy.

Uma representação gráfica convencional de valores na lógica fuzzy é ilustrada na Figura 4.1, em que a altura das pessoas é representada na abscissa e três funções (baixo, mediano e alto) representam a classificação das pessoas quanto à altura.

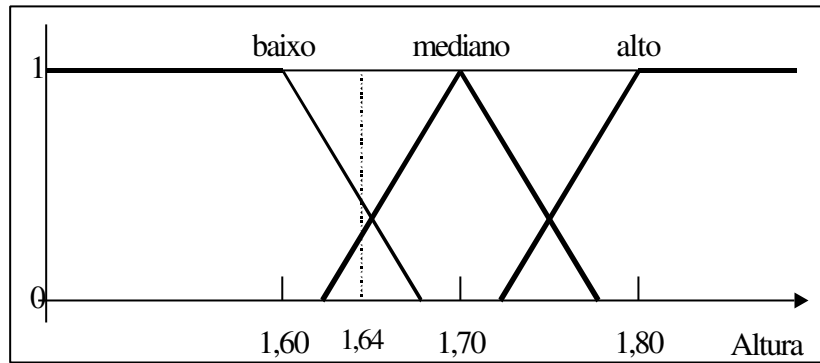


Figura 4.1 - Representação de valores na lógica fuzzy.

Uma pessoa medindo 1,64 m de altura é considerada baixa, de acordo com o gráfico da Figura 4.1, mas está muito próxima de ser considerada de altura mediana. Entretanto, é possível observar que uma pessoa é totalmente baixa de 0 até 1,60 m (a curva “baixo” indica valor 1 e as demais valor 0 na faixa de valores indicada). A partir de 1,60 m a reta que define o valor baixo começa a decrescer, enquanto a reta que define a altura mediana começa a crescer.

Uma pessoa com 1,70 m é considerada de altura mediana e acima de 1,80 m é considerada alta. Essas medidas são totalmente imprecisas, variando de acordo com os conceitos de cada pessoa, região, cidade, país e outros. Em muitas situações porém, os valores inexatos são mais importantes e possuem significados mais expressivos do que os valores exatos.

## 4.2 Fundamentos da Lógica Fuzzy

A lógica fuzzy foi proposta por Lotfi A. Zadeh em 1965 como uma matemática que podia representar as incertezas do cotidiano [BEZ94]; é basicamente uma linguagem que serve para descrever e analisar dependências imprecisas [ZAD94]. Diante dos problemas da lógica booleana e dos recursos oferecidos pela lógica fuzzy, muitos pesquisadores passaram a utilizá-la como ferramenta para o desenvolvimento de sistemas inteligentes. Atualmente, há uma grande variedade de pesquisas envolvendo a lógica fuzzy.

Na teoria clássica dos conjuntos um elemento do universo de discurso (domínio) pertencente ou não ao referido conjunto, assume valores 0 ou 1 [KLI88]. Na teoria dos conjuntos fuzzy existe um grau de pertinência de cada elemento que pode assumir qualquer

valor dentro do intervalo  $[0,1]$  [KLI88].

O valor 1 representa completa pertinência e o valor 0 indica uma completa exclusão. Essa generalização aumenta significativamente o poder de expressão da função característica, onde esta função, diz o grau de pertinência  $\mu_A(x)$  de um elemento  $x$  pertencente a um universo  $U$  com respeito a um conjunto  $A$ , onde  $A \subseteq U$ .

Assim, se  $Y$  é uma variável que toma valores num universo  $U$ , então, a distribuição de possibilidades associada com  $Y$  pode ser vista como uma condição elástica nos valores que podem ser atribuídos a  $Y$ . Por exemplo, se  $F$  é um subconjunto fuzzy de  $U$  caracterizado pela função de pertinência  $\mu_f:U \rightarrow [0,1]$ , então a declaração “ $Y$  é  $F$ ” é traduzida numa distribuição de possibilidade para  $Y$  sendo igual a  $F$ . Então pode-se dizer que o conjunto fuzzy  $A$  é definido como o par ordenado  $A = \{x, \mu_A(x)\}$  [KLI88], Onde  $x \in X$  e  $0 \leq \mu_A(x) \leq 1$ . A função de pertinência  $\mu_A(x)$  descreve o grau a que o objeto  $x$  pertence ao conjunto  $A$ . Quando  $\mu_A(x) = 0$ , representa nenhuma pertinência e  $\mu_A(x) = 1$ , representa pertinência total.

Por exemplo, seja  $X$  a representação da altura de uma pessoa. O subconjunto  $A$  de  $X$  que representa as pessoas baixas ( $\mu_A(x)$ ) é o conjunto fuzzy com a função de pertinência mostrada na Figura 4.2. Observe que  $\mu_A(x) \cong 1$  para pessoas com até 1,60m de altura. A partir de 1,60m,  $\mu_A(x)$  começa a decrescer. Para  $X = 1,60m$ ,  $\mu_A(x) \cong 1$ .

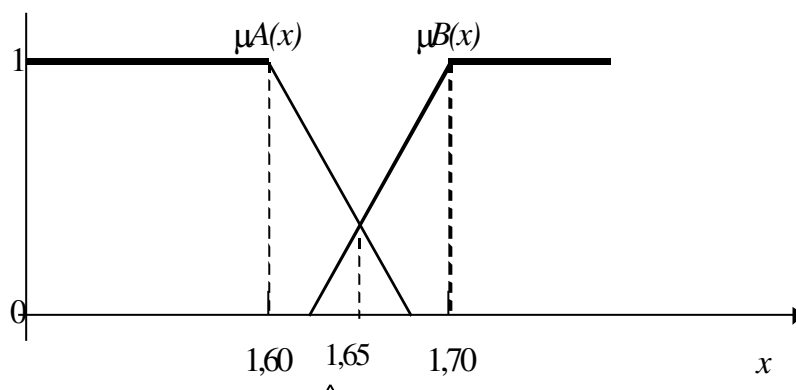


Figura 4.2 - Função de pertinência



Um conjunto fuzzy pode ter mais de uma representação, assim podemos descrever a altura de uma pessoa também por um subconjunto B de X representando as pessoas medianas ( $\mu_B(x)$ ), cuja função de pertinência esta ilustrada na Figura 4.2.

Observe que  $\mu_B(x) \equiv 1$  para pessoas com altura acima de 1,70. Também observe que para  $X=1,65m$ ,  $\mu_A(x)=\mu_B(x)$ , ou seja, uma pessoa com 1,65m tem o mesmo grau de altura tanto para baixo como para mediano[KL188].

## 4.3 Operações

A teoria originária dos conjuntos fuzzy foi fundamentada nos termos das três operações feitas com conjuntos (*complemento, união e interseção*) que são equivalentes as operações da lógica booleana (*negação, ou, e*). Assumindo que  $A \in X$  e  $B \in X$ , apresentam-se a seguir essas operações.

### 4.3.1 Complemento

A operação complemento é utilizada para definir a função de pertinência oposta de um subconjunto, ou seja, o complemento do subconjunto A, definido como  $A'$ , é formado pelos pontos opostos de A de dentro do intervalo  $[0, 1]$ . Essa operação, quando tratada nos extremos desse intervalo, é equivalente à operação “negação” da lógica booleana. A representação formal da operação complemento é descrita na Equação 4.1. A sua representação gráfica é ilustrada na Figura 4.3.

$$\mu_{A'}(x) = 1 - \mu_A(x) \quad \forall x \in X \quad (4.1)$$

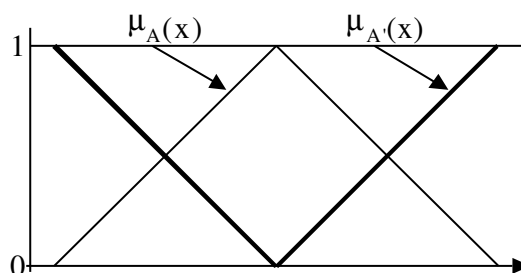


Figura 4.3 - Representação gráfica da operação complemento.

### 4.3.2 União

A operação união é utilizada para associar dois subconjuntos, ou seja, a união do subconjunto A com B resulta em um subconjunto abrangendo os pontos máximos dos dois subconjuntos unidos. Essa operação, quando tratada nos extremos do intervalo  $[0, 1]$ , é equivalente à operação “ou” da lógica booleana. A representação formal da operação união é descrita na Equação 4.2. A sua representação gráfica é ilustrada na Figura 4.4.

$$A \cup B \rightarrow \mu_A(x) \cup \mu_B(x) = \max(\mu_A(x), \mu_B(x)) \quad \forall x \in X \quad (4.2)$$

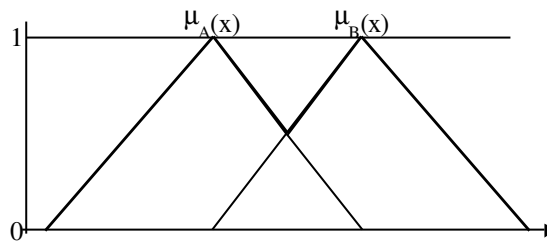


Figura 4.4 - Representação gráfica da operação união.

### 4.3.3 Interseção

A operação interseção é utilizada para definir a região comum entre dois subconjuntos, ou seja, a interseção do subconjunto A com B resulta em um subconjunto abrangendo os pontos que pertencem tanto ao subconjunto A quanto ao subconjunto B. Essa operação, quando tratada nos extremos do intervalo  $[0, 1]$ , é equivalente à operação “e” da lógica booleana. A representação formal da operação interseção é descrita na Equação 4.3. A sua representação gráfica é ilustrada na Figura 4.5.

$$A \cap B \rightarrow \mu_A(x) \cap \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \quad \forall x \in X \quad (4.3)$$

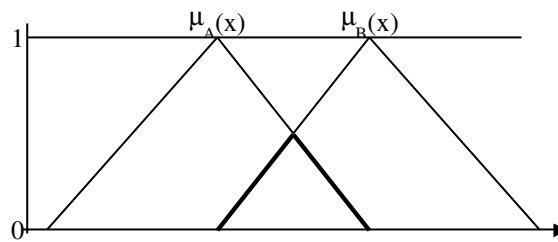


Figura 4.5 - Representação gráfica da operação interseção.

## 4.4 Expressão Fuzzy do Conhecimento

Para expressar conceito é muito comum o uso de elementos qualitativos ao invés de valores quantitativos [GOM 94]. Elementos típicos incluem “mais ou menos”, “alto”, “não muito”, “médio”, entre outros.

Estas idéias são ditas pela definição de variável lingüística. Uma variável lingüística tem por característica assumir valores dentro de um conjunto de termos lingüísticos, ou seja, palavras ou frases [SAL97].

Uma variável lingüística altura poderá assumir um dos membros do conjunto {muito alto, alto, médio, médio alto, baixo, muito baixo}. Para se atribuir um significado aos termos lingüísticos, associa-se a cada um deles um conjunto fuzzy definido sobre um universo de discurso comum.

Uma das formas mais comuns de se expressar o conhecimento é por meio de regras do tipo condição-ação. Exemplificando, um conjunto de condições que descrevem uma parcela observável das saídas do processo, é associado a uma ação de controle que irá manter ou elevar o processo às condições de operações desejadas.

A idéia aqui é representar o conhecimento por meio de um conjunto de regras nas quais as condições são dadas a partir de um conjunto de termos lingüísticos associados a variáveis de entrada/saída do processo (as quais são entradas do controlador).

As ações de controle ou as saídas são expressas de modo similar para cada variável de controle (saída). Regras do tipo *se-então-senão* são freqüentemente chamadas de regras de controle fuzzy.

A lógica fuzzy pode ser descrita na forma das lógicas tradicionais. Por exemplo,

If  $X_1 = A_1$  &  $X_n = A_n$ , then  $Y = B_1$ ,

If  $X_1 = C_1$  &  $X_n = C_n$ , then  $Y = B_2$ .

Tipicamente, uma proposição lingüística sobre o valor das variáveis de entrada é, por exemplo, o erro é grande e positivo. De modo análogo, uma típica ação de controle é uma descrição lingüística, como por exemplo, aumente um pouco a velocidade do carro.

## 4.5 Sistemas de Controle Fuzzy

O controle fuzzy não necessita da modelagem do processo, e sim, da modelagem das ações a partir de um conhecimento de um especialista. Essa é, portanto, uma abordagem diferente dos métodos convencionais de controle de processos [AST89], pois os mesmos são desenvolvidos via modelagem matemática dos processos de modo a derivar as ações de controle como função do estado do processo. A estrutura básica de um controlador fuzzy esta ilustrado na Figura 4.6.

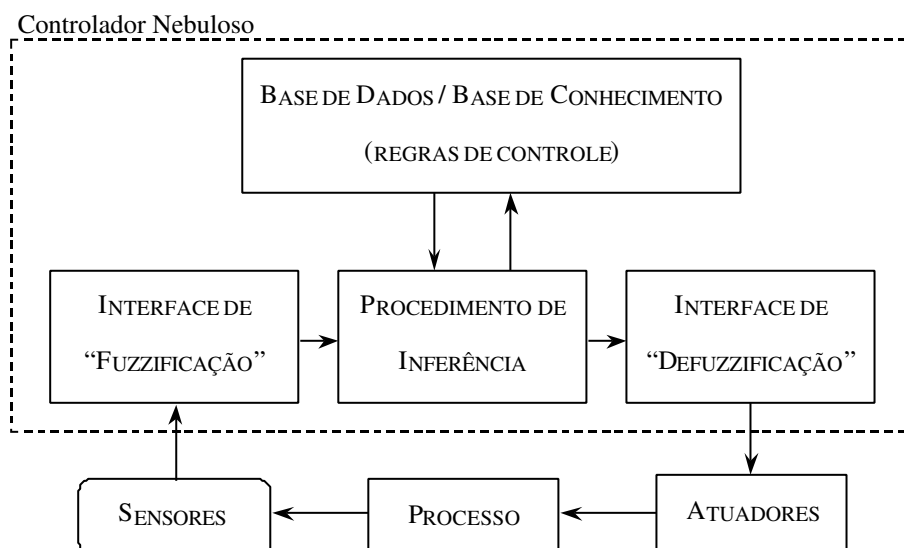


Figura 4.6 - Sistema de Controle Fuzzy.

O núcleo do controlador só analisa variáveis fuzzy. As informações têm que ser transformadas na forma fuzzy ou “fuzzyficadas” (transformadas em conjuntos fuzzy). A interface de “fuzzyficação” recebe os valores das variáveis de entrada (vindo dos

sensores), faz um escalonamento para dimensionar os valores a universos discursos normalizados e “*fuzzyfica*” os valores ( transformando números em conjuntos fuzzy), para torná-los instâncias de variáveis lingüísticas. A base de conhecimento consiste de um conjunto de regras que caracterizam as estratégias de controle e seus objetivos. A base de dados armazena as informações necessárias sobre as discretizações, as normalizações dos universos de discurso, as partições fuzzy dos espaços de entrada e saída e as definições das funções de pertinência.

O procedimento de inferência atua sobre os dados fuzzy de entrada, juntamente com as regras, para inferir as ações de controle fuzzy, usando o operador de implicação fuzzy e as regras de inferência da lógica fuzzy. A inferência de “*defuzzyficação*” atua sobre as ações de controle fuzzy inferidas, transformando-as em ações de controle não fuzzy, efetuando, em seguida, um escalonamento para compatibilizar os valores normalizados vindos do passo anterior com os valores dos universos de discursos reais das variáveis [GUP96].

Deve-se determinar uma ação de controle não-fuzzy para ser enviada ao controlador logo após se inferir a ação de controle fuzzy. A ação de controle não-fuzzy escolhida deve ser a que represente melhor a decisão fuzzy. Não há nenhum procedimento sistemático para escolher a estratégia de “*defuzzyficação*”.

As estratégias mais comuns são: o critério do máximo (MAX), o qual escolhe o ponto onde a função inferida tem o seu máximo; a média dos máximos (MDM), representa o valor médio entre todos os pontos de máximos se existir mais de um máximo; e o método de centro de área (CDA) que retorna as coordenadas do centro da área da função inferida.

## **4.6 Utilização da Lógica Fuzzy**

A primeira aplicação da Lógica Fuzzy bem sucedida foi no desenvolvimento de controladores industriais. Controladores que se baseiam na Lógica Fuzzy são chamados de Controladores Fuzzy. Não é preciso conhecer muita matemática ou em profundidade a teoria de controle para se desenvolver uma aplicação em controle. Controladores fuzzy tratam igualmente sistemas lineares e não lineares, além de não requererem o modelo matemático do processo a ser controlado. Isto tem sido, sem dúvida, o grande atrativo dos

Sistemas Fuzzy.

Sistemas baseados na Lógica Fuzzy têm mostrado grande utilidade em uma variedade de operações de controle industrial e em tarefas de reconhecimento de padrões que se estendem desde reconhecimento de texto manuscrito, até a avaliação de crédito financeiro. Existe também um interesse crescente em se utilizar Lógica Fuzzy em sistemas especialistas para torná-los mais flexíveis.

No Japão, a Lógica Fuzzy já se faz presente no dia a dia do setor industrial e muitos produtos comerciais já se encontram disponíveis.

Inicialmente, sistemas Fuzzy foram ignorados nos Estados Unidos porque foram associados com Inteligência Artificial, um campo que periodicamente se obscurecia, resultando numa falta de credibilidade por parte de alguns segmentos da indústria.

Diversas pesquisas e desenvolvimentos estão em andamento utilizando a lógica Fuzzy. Projetos de software, incluindo sistemas especialistas Fuzzy e integração de lógica Fuzzy com Redes Neurais, os Algoritmos Genéticos Adaptativos que são utilizados no intuito de construir um sistema Fuzzy capaz de aprender.

## **4.7 Perspectivas da Lógica Fuzzy**

Diversas áreas estão sendo beneficiadas pela tecnologia decorrente da Lógica Fuzzy. Dentre essas áreas podem ser citadas algumas que tiveram relevância no avanço tecnológico e que merecem destaque. O Controle de processos industriais foi a área pioneira, sendo as primeiras experiências datadas de 1975 quando foi demonstrado no “Queen College”, em Londres, que um controlador Fuzzy muito simples conseguiu controlar eficientemente uma máquina a vapor.

Nos últimos anos o potencial de manuseio de incertezas e de controle de sistemas complexos tornados possíveis pela Lógica Fuzzy, estão sendo combinados com Redes Neurais artificiais, que por sua vez, possuem características de adaptação e aprendizagem.

A palavra certa para isto é simbiose (cooperação mútua entre pessoas ou grupos em uma sociedade), que vem gerando novas classes de sistemas e de controladores neurofuzzy, combinando desta forma os potenciais e as características individuais em sistemas

adaptativos e inteligentes [GOM92]. Com certeza estes sistemas deverão proporcionar uma significativa contribuição para os sistemas de automação e controle do futuro, principalmente em controle de processos.

## **4.8 Conclusão**

A lógica fuzzy foi descrita neste capítulo de forma resumida para introduzir os principais conceitos. Observou-se que esta lógica possui características fundamentais para a solução de determinados tipos de problemas, principalmente aqueles relacionados com tomada de decisão sobre valores imprecisos.

# *CAPÍTULO 5*

## **5. MAUTOXAD**

### **5.1 Introdução**

Atualmente, os robôs devem perceber o meio em que estão inseridos e modificar suas ações para a realização de tarefas específicas através de um comportamento inteligente. O conjunto de tarefas usadas no posicionamento de um braço robótico para manipulação de objetos exige um sistema inteligente.

Inicialmente o ROBOXAD (Robô para mover peças em um tabuleiro de Xadrez) compunha-se de um manipulador robótico com quatro graus de liberdade [FER99], com uma garra magnética (ímã) na sua extremidade. Em sua versão atual apresenta mais um grau de liberdade, trata-se de um trilho que foi acrescentado à sua arquitetura. O ROBOXAD tem sido utilizado como base de testes para o desenvolvimento e implementação de Sistemas Inteligentes (SI) para gerenciamento e controle de robôs.

Neste capítulo serão apresentados alguns detalhes mecânicos do protótipo do ROBOXAD e uma descrição do sistema inteligente, denominado AUTOXAD (Sistema Inteligente do ROBOXAD) [FER99], desenvolvido para o movimento de peças no tabuleiro de xadrez. A seguir, apresenta-se o projeto do novo sistema de controle baseado em agentes MAUTOXAD (Sistema Inteligente do ROBOXAD utilizando Múltiplos Agentes), sua descrição, arquitetura e funcionamento, e por fim os resultados experimentais obtidos na movimentação de peças no tabuleiro de xadrez.



## 5.2 Descrição do ROBOXAD e do AUTOXAD

Observou-se que o ROBOXAD só conseguia movimentar peças de xadrez dentro de um tabuleiro com três linhas. Para aumentar essa área de trabalho (para seis linhas), foi adicionado um novo dispositivo, denominado trilho, que permite o movimento do tabuleiro de xadrez de forma a aproximá-lo ou afastá-lo da base do ROBOXAD.

No ROBOXAD, apresentado na fotografia da Figura 5.1, cada componente tem uma função: o braço mecânico (que é formado por três elos) se posiciona sobre o tabuleiro, o ímã (garra) se constitui do componente que tem a capacidade de pegar e soltar as peças de xadrez, e o tabuleiro móvel cujo objetivo é de expandir a capacidade de alcance do braço. Um desenho esquemático do ROBOXAD é apresentado na Figura 5.2 onde são apontados cada um de seus componentes.

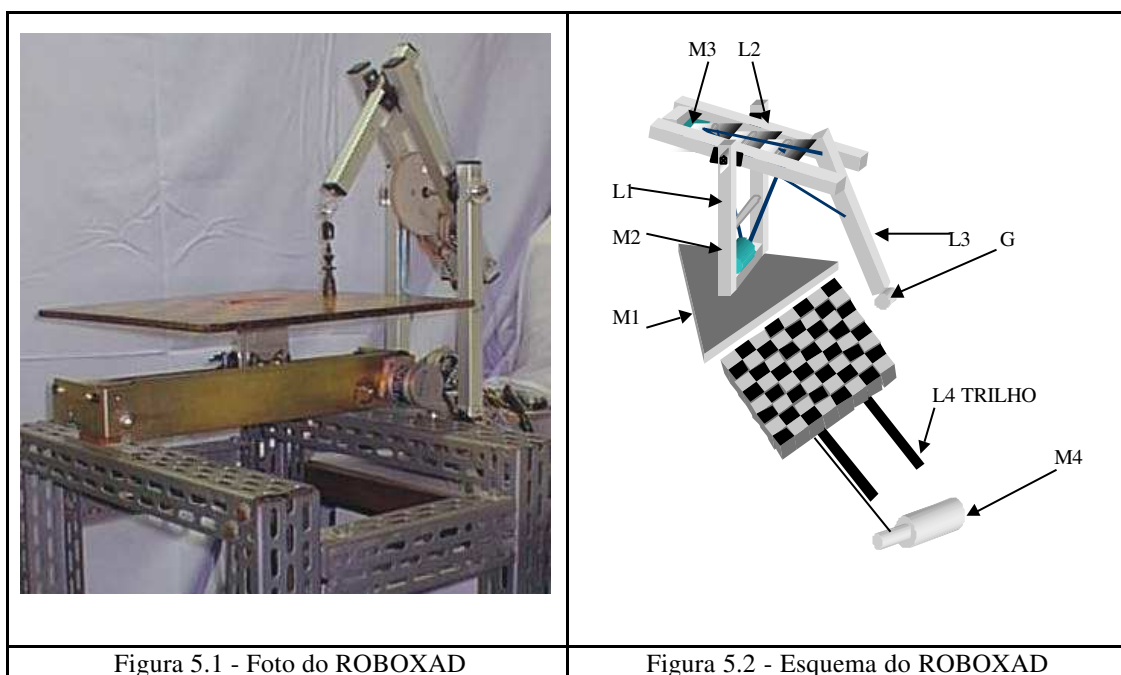


Figura 5.1 - Foto do ROBOXAD

Figura 5.2 - Esquema do ROBOXAD

L1 – Elo Base

L4 - Elo Trilho

M – Motor de Passo

L2 – Elo Antebraço

G - Garra ímã

J - Junta (M + L)

L3 – Elo Braço

Em cada junta (J) do robô está acoplado um motor de passo (M), sendo que a junta J1 (motor M1 e elo L1) gira em torno da normal ao chão, e seus movimentos são

para a direita e a esquerda, ela gira em torno da normal ao Eixo Base (Figura 5.3). As juntas J2 e J3 (motores M2 e M3 e elos L2 e L3 respectivamente) movem-se para cima e para baixo. A garra (G) foi implementada como um ímã. A junta J4 (motor M4 e elo L4 ou trilho) representa o tabuleiro de xadrez que é aproximado ou afastado do Eixo Base.

Convencionou-se que a frente do tabuleiro deve ser paralela ao lado frontal do triângulo equilátero da base do robô (Figura 5.4). Além disso, a reta perpendicular a esse lado deve passar pelo centro do tabuleiro dividindo-o em dois lados (lado da dama e lado do rei). Esses posicionamentos são necessários devido a análise de simetria que foi feita para especificar as posições das peças no tabuleiro de xadrez [FER99a].

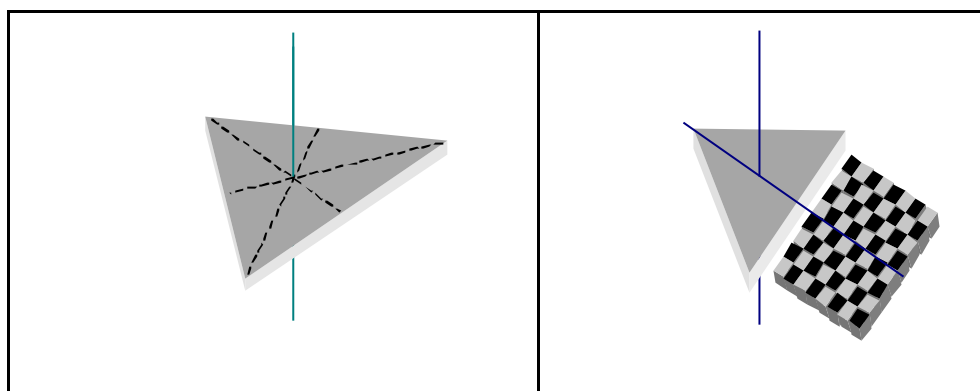


Figura 5.3 - Eixo base

Figura 5.4 - Posição das estruturas

O AUTOXAD [FER99] é um Sistema Inteligente que foi desenvolvido para o posicionamento das peças de xadrez sobre o tabuleiro. Compõe-se de uma BASE DE CONHECIMENTO, onde estão armazenadas as informações, em forma de regras, sobre a localização de todas as casas do tabuleiro de xadrez; um módulo Fuzzy [ZAD88] que permite a inferência sobre as regras; e um módulo ATUADOR encarregado do movimento dos braços do robô (Figura 5.5).

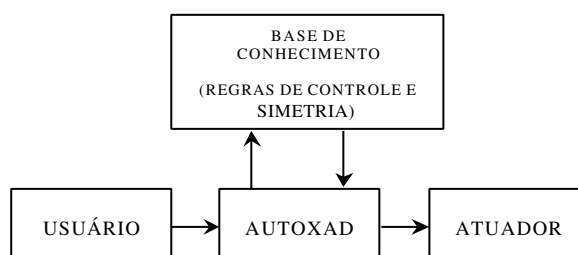


Figura 5.5 - Esquema básico do SI

Analisando-se o tabuleiro de xadrez, pode-se observar a existência de simetria entre os lados da dama e do rei do tabuleiro. Por exemplo, se a garra do AUTOXAD estiver posicionada sobre o ponto X (encontro das linhas tracejadas), como vemos na Figura 5.6, o movimento para a coluna D é simétrico ao movimento para a coluna R, pois o manipulador utilizará apenas o motor M1 (da base) para deslocar a garra e posicioná-la sobre uma das duas colunas. Observa-se que todas colunas de um lado do tabuleiro possuem colunas simétricas do outro lado.

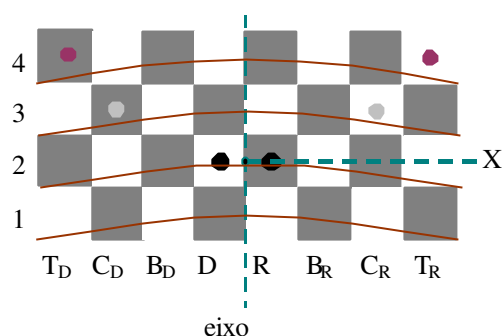


Figura 5.6 - Casas simétricas na mesma linha

A cada movimento das peças no tabuleiro de xadrez foi associado uma casa fonte e uma casa destino. Por exemplo, 3CD e 3CR , indicando o movimento de uma peça vinda da terceira linha da coluna do cavalo da dama (fonte) para a terceira linha da coluna do bispo do rei (destino).

O SI necessita de um conhecimento prévio de seu ambiente de trabalho, ele deve conhecer o tabuleiro e suas casas, as peças utilizadas no jogo, a posição inicial da garra no tabuleiro (centro), e outras informações. Isto significa que o sistema necessita de uma base de conhecimento composta por dados e regras para que o manipulador consiga alcançar sua meta com êxito (Ver Figura 5.5).

A cada ação do AUTOXAD foi associada uma meta ou objetivo. A meta do sistema foi planejada a partir da analogia de uma situação do mundo real, isto é, baseou-se na observação dos movimentos realizados por um jogador de xadrez. Com isso, foi possível predefinir uma seqüência de movimentos necessários para a realização da meta desejada. Definiram-se duas estratégias: IDENTIFICA e MOVE. A estratégia IDENTIFICA localiza a posição da peça e da garra (fonte), pega a peça (acionando o ímã), e a seguir aciona a

estratégia MOVE. A estratégia MOVE movimenta o robô para a posição da peça e a garra solta a peça (desligando o ímã) no lugar determinado (destino).

As estratégias foram implementadas em forma de tarefas para que fosse possível executá-las através de um escalonamento em tempo real [FER98]. A cada tarefa foi associado um descritor. Os descritores são representados por descr(id, st, temp, freq) onde, id é a identificação da tarefa; st, indica o estado da tarefa (executando=-1, bloqueada=0, e pronta=1); tempo, o tempo de ativação da tarefa em ms; e freq, intervalo de tempo em que a tarefa será novamente ativada.

No posicionamento da garra do ROBOXAD utilizando SIMETRIA, inicialmente deve-se especificar o posicionamento das casas fonte e destino. Por exemplo, sabendo-se que a peça\_fonte está em 3CD e a casa\_destino é 3CR, o AUTOXAD detecta que: o lado\_fonte=D, o lado\_destino=R, a linha\_fonte=3, a linha\_destino=3, a coluna\_fonte=C e a coluna\_destino=B. O módulo que analisa a SIMETRIA no movimento da peça da casa fonte para a casa destino é apresentado no quadro das regras 5.1. No quadro 5.1 os sinais == indicam igualdade e != indicam diferença.

```
SIMETRIA=0 //Variável booleana que indica a simetria
if (posicionado) //Garra posicionada sobre uma casa
then
  if(linha_fonte == linha_destino) //Casas fonte e destino na mesma linha
  then
    if(lado_fonte != lado_destino) //Lados da dama e do rei
    then
      if(coluna_fonte==coluna_destino || coluna_fonte==(R | D) || coluna_fonte==(D | R))
      then SIMETRIA = 1
```

Quadro 5.1 - Quadro de regras de simetria

Quando não existe simetria os elos L2 e L3 do ROBOXAD devem ser movidos. Na Figura 5.7 apresenta-se um desenho no plano cartesiano dos elos L2 e L3. O plano cartesiano é perpendicular ao eixo base do AUTOXAD. Observe-se que J2 está na origem

dos eixos X e Y. Considere-se que o braço L2 faz um ângulo  $\theta_2$  em relação ao eixo X e L3 faz um ângulo  $\theta_3$  em relação ao eixo X, e que a garra do braço L3 está sobre o eixo X. Na Figura 5.7 os elos L2 e L3 possuem o mesmo comprimento.

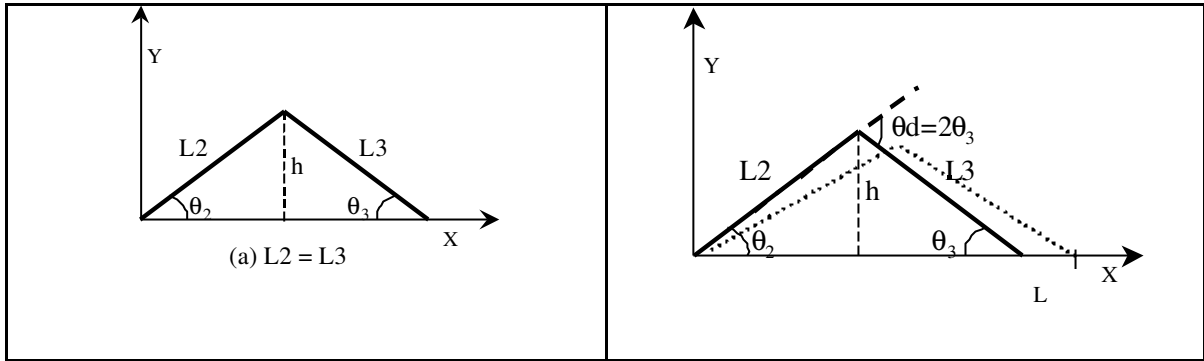


Figura 5.7 - Posicionamento dos elos L2 e L3

Figura 5.8 - Movimento da garra sobre o plano

Para que a garra seja movida sobre o eixo X, sobre o plano em que está o tabuleiro de xadrez, os valores dos ângulos  $\theta_2$  e  $\theta_3$  devem ser iguais. Nas Figuras 5.7 e 5.8 apresentam-se os desenhos no plano cartesiano de L2 e L3 quando a garra é movimentada sobre o plano do tabuleiro de xadrez. Utilizando-se geometria básica consegue-se calcular os ângulos  $\theta_2$  e  $\theta_3$  sendo conhecido o ponto em que a garra encosta no eixo X. O ROBOXAD possui braços com comprimentos iguais ( $L_2=L_3$ ).

$$\text{sen}\theta_2 = \frac{h}{L_2} \quad (5.1)$$

$$\text{sen}\theta_3 = \frac{h}{L_3} \quad (5.2)$$

$$\frac{\text{sen}\theta_2}{\text{sen}\theta_3} = \frac{L_3}{L_2} \quad (5.3)$$

$$\text{sen}\theta_3 = \frac{L_2}{L_3} \text{sen}\theta_2 \quad (5.4)$$

Se os elos L2 e L3 possuírem o mesmo comprimento, a partir da equação 5.4 obtém-se  $\theta_3 = \theta_2$ . Se os comprimentos dos elos L2 e L3 forem diferentes, a partir da equação 5.4, obtém-se a equação 5.5. A regra fuzzy mostrada na equação 5.6, utilizada no Sistema Inteligente do AUTOXAD, ilustra a movimentação da garra sobre o eixo x quando  $\theta_3 = \theta_2$ . A variável booleana `garra_sobre_eixo_X` indica que a garra está

posicionada sobre o eixo X das coordenadas cartesianas. O AUTOXAD possui braços com comprimentos iguais ( $L_2=L_3$ ).

$$\theta_3 = \arcsen\left(\frac{L_2}{L_3} \text{sen}\theta_2\right) \quad (5.5)$$

$$\text{if } L_2 == L_3 \text{ and } \theta_2 == \theta_3 \quad \text{then garra\_sobre\_eixo\_X} \quad (5.6)$$

$$\text{if lado\_fonte} == \text{lado\_destino} \quad \text{then } \theta = \theta\_destino - \theta\_fonte \quad (5.7)$$

$$\text{if lado\_fonte} != \text{lado\_destino} \quad \text{then } \theta = \theta\_destino + \theta\_fonte \quad (5.8)$$

A Figura 5.9 ilustra a representação do posicionamento da garra do manipulador sobre as posições fonte (2R) e destino (2CR). O primeiro passo do AUTOXAD é verificar se a garra está sobre a casa fonte e que a casa destino é diferente da casa fonte. O segundo passo do AUTOXAD é verificar, utilizando as regras do quadro 5.1, se a posição\_fonte=2R é simétrica em relação a posição\_destino=2CR. Após a confirmação de que as posições “fonte” e “destino” não são simétricas, calcula-se a distância a ser percorrida pelos elos L2 e L3 e o ângulo  $\theta$  de rotação de L1 na direção da posição destino.

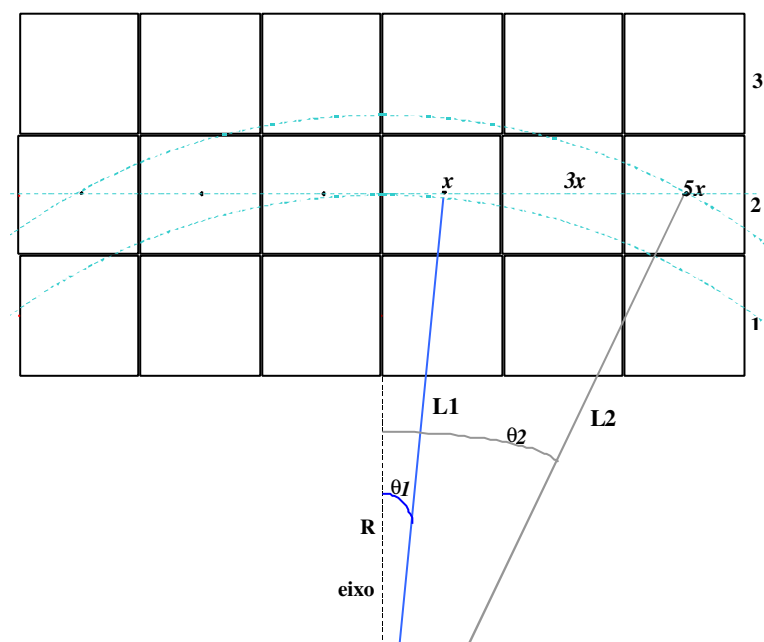


Figura 5.9 - Representação do posicionamento da garra nas posições 2R→2CR.

Na tabela 5.1 apresentam-se os dados de posicionamento da casa fonte e os dados de posicionamento da casa destino. Com esses dados pode-se calcular o deslocamento que deve ser feito pelos elos L1, L2 e L3.

Posição	R	x	Y
Fonte	15	2	6
Destino	15	10	6

Tabela 5.1 - Coordenadas das posições 2R e 2CR.

O cálculo dos deslocamentos que devem ser feito pelos elos L2 e L3 é feito com a fórmula  $\sqrt{(R+y)^2 + x^2}$ . Obtém-se como resultado: L1=21,09 cm e L2=23,26 cm. A distância a ser percorrida pelos elos L2 e L3 é dada por  $l=L2-L1$ . Obtém-se  $l=2,17$ cm. O motor de passo do elo L2 necessita de dois passos para deslocar 1 cm. Sabendo-se que o deslocamento em passos de L3 é duas vezes o deslocamento de L2, conclui-se que a quantidade de passos a serem percorridos pelos motores de L2 e L3 são 4 e 8 respectivamente.

O ângulo do plano formado pelos braços L2 e L3 em relação ao eixo base é dado pela Equação (5.9). Observe-se que se deve conhecer R e x para se encontrar o ângulo  $\theta_1$ . O motor de passo da base do AUTOXAD é conectado. Por correia, a um disco rígido que está acoplado ao eixo da base. A relação dos raios dos círculos do motor de passo e do disco é igual a 9. Sabendo-se que o motor de passo possui um passo de 7.5 graus, necessitando de 48 passos para uma rotação, calculam-se  $D=9*48=432$  passos para uma rotação do disco (ou 360°). Calculou-se 1,2 passos/grau. Por exemplo, para  $\theta=15^\circ$ , o número de passos é  $np=1,2*15=18$ passos.

$$\theta_1 = \arctg\left(\frac{x}{R+3x}\right) \quad (5.9)$$

O ângulo rotacional  $\theta_1$  é calculado pela regra fuzzy mostrada na Equação 5.2, então  $\theta = 20,02^\circ$ , que corresponde a 24 passos a ser percorrido pelo elo da base (ou L1).

$$(\theta = \theta_{\text{DESTINO}} - \theta_{\text{FONTE}}) \quad (5.10)$$

Para posicionar a garra em L2 deve-se movimentar os elos L2 e L3, observe-se que  $L2 > L1$ , o que pode ser visto na projeção de L1 sobre L2. Define-se a projeção de L1 sobre L2 como o ponto em que o círculo de raio L1 cruza a reta que passa sobre L2. A diferença entre os dois raios ( $l = L2 - L1$ ) representa a distância que os dois elos (L2 e L3) devem percorrer sobre o eixo X (ver Figura 5.8).

De acordo com a regra da equação 5.6 para que a garra seja movida sobre o eixo X os valores dos ângulos devem ser iguais. Na Figura 5.8 a posição fonte (ângulos  $\theta_2$  e  $\theta_3$ , ou  $\theta$ ) dos elos L2 e L3 estão representadas com linhas cheias e a posição destino (ângulos  $\theta_2'$  e  $\theta_3'$  ou  $\theta'$ ) destes elos estão representadas com linhas tracejadas. Observe-se que  $l = 2 * L * (\cos(\theta') - \cos(\theta))$ . Conhecendo-se  $l$  calcula-se  $\theta'$  utilizando a equação 5.12.

$$\cos(\theta') = \cos(\theta) + l/2 \quad (5.11)$$

$$\theta' = \arccos[\cos(\theta) + l/2] \quad (5.12)$$

A comunicação entre o software (AUTOXAD) e o hardware (ROBOXAD) é feita através de uma das portas paralelas do computador com a utilização de uma placa de interface controladora (Apêndice B) que permite converter os níveis de corrente extremamente baixos da porta paralela em sinais adequados para acionar os dispositivos de potência que alimentam os atuadores; em nosso caso os motores de passo (Apêndice A).

A seguir será apresentado o novo Sistema Inteligente (MAUTOXAD) desenvolvido para o controle do braço robótico. O MAUTOXAD foi desenvolvido utilizando o conceito de Agentes Inteligentes.

### **5.3 Sistema de Controle utilizando Agentes Inteligentes**

A concepção de sistemas inteligentes baseados no conceito de agentes pode ser orientada tanto para um agente, quanto para uma sociedade de agentes. Quando em sociedade, os agentes lidam com conceitos como cooperação, competição e comunicação para a realização de uma tarefa. Um agente robótico utiliza, por exemplo, câmaras e raios



infravermelhos, como sensores e vários motores como efetadores.

Neste trabalho considerou-se o ambiente como sendo o tabuleiro de xadrez e os efetadores como os elos do robô (Figura 5.10).

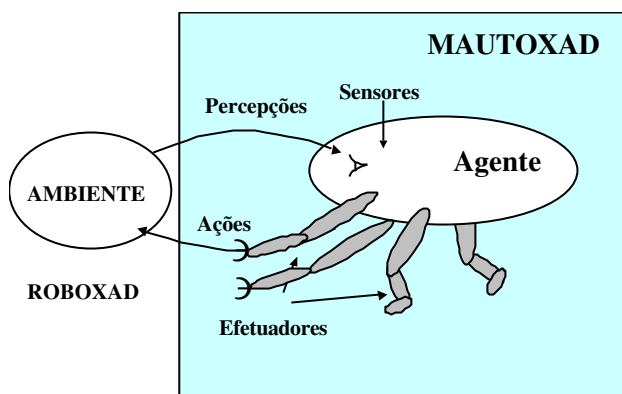


Figura 5.10 - Representação de Agentes

O novo Sistema Inteligente implementado para o ROBOXAD, utiliza uma Abordagem de Agentes, e denomina-se MAUTOXAD. O MAUTOXAD compõe-se de um Agente Mestre e diversos Agentes escravos.

O MAUTOXAD é composto ao todo por seis agentes inteligentes, sendo um deles. O Agente Mestre que controla os Agentes escravos, que são: Agente1 (base), Agente2 (braço), Agente3 (antebraço), Agente Tabuleiro (trilho) e Agente Ímã (garra). Cada Agente Escravo é responsável por um componente do sistema. Na Figura 5.11 apresenta-se um esquema que ilustra as relações entre os agentes, o ambiente ROBOXAD e o usuário.

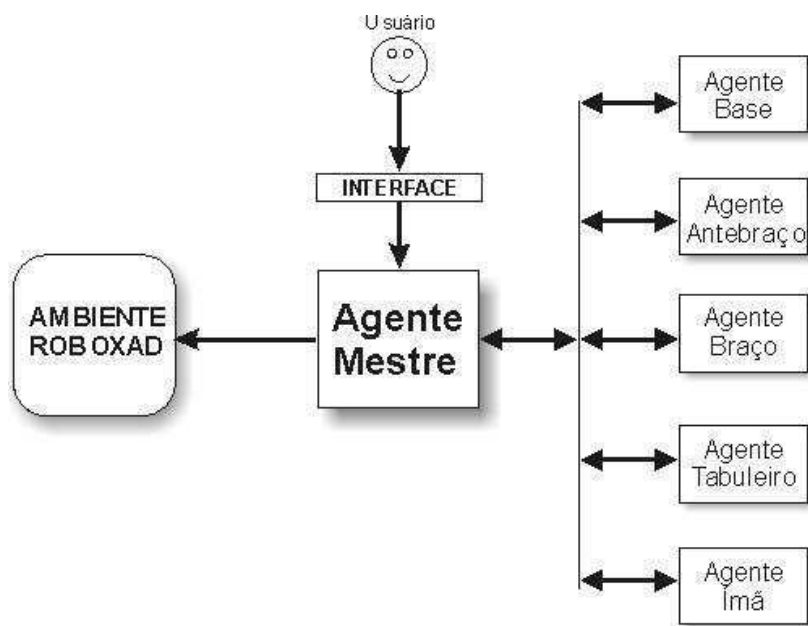


Figura 5.11 - Relações entre os componentes do MAUTOXAD

A seguir, será feita uma descrição do Sistema MAUTOXAD: sua arquitetura, seus agentes (funções e características) e as características do ambiente.

### 5.3.1 Descrição do MAUTOXAD

Para a construção dos agentes do sistema MAUTOXAD, foi adotada a arquitetura de agentes reativa. De acordo com Oliveira [OLI96], as Arquiteturas Reativas são próprias de agentes que possuem estrutura interna simples e interação de forma limitada; geralmente não possuem representação dos estados mentais; o desempenho da sociedade é resultado do número de agentes e da rapidez nas interações (ver seção 3.2.4.2).

#### 5.3.1.1 Os Agentes do MAUTOXAD

O Agente Mestre é o agente principal do funcionamento do sistema (Figura 5.12). Ele possui uma arquitetura reativa e tem a função de receber do usuário a jogada (deslocamento da peça de uma posição fonte para uma posição destino), comunicar-se com os agentes informando a jogada, e receber deles as respectivas tarefas para a execução da mesma.

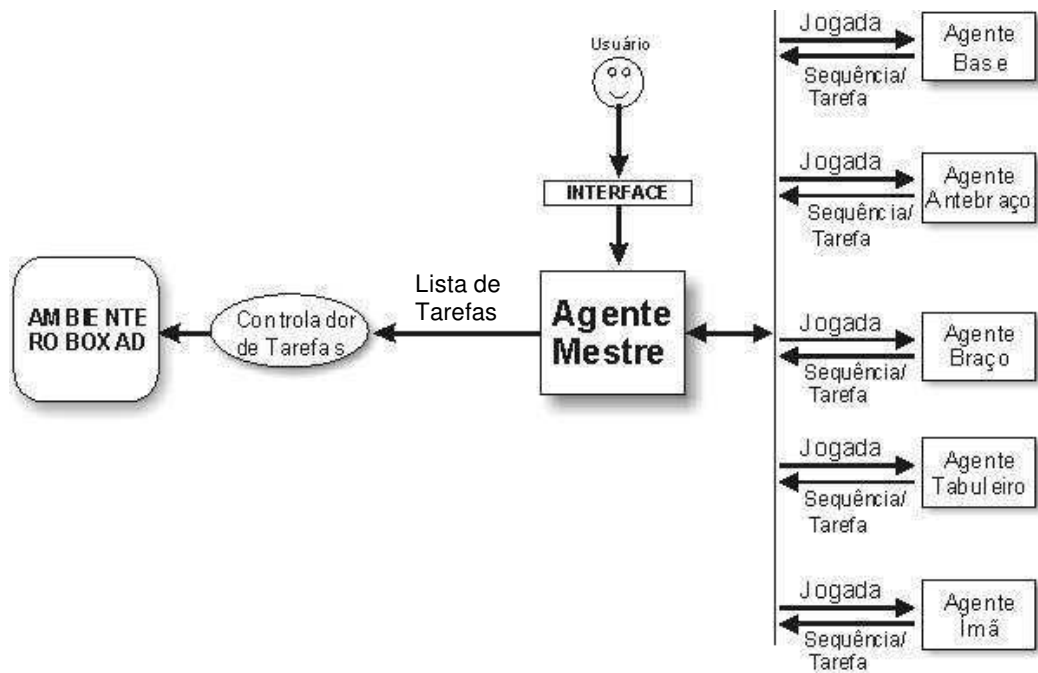


Figura 5.12 - Esquema Geral do MAUTOXAD

Inicialmente ele informa aos agentes a jogada que deve ser realizada e os consulta com o objetivo de saber qual será a seqüência de movimentos, ou seja, em que ordem os Agentes escravos irão se mover. Em seguida, o Agente Mestre reenvia a jogada para os Agentes escravos calcularem os movimentos (ângulos e passos) necessários para a execução da jogada. Após o cálculo dos movimentos, estes são retornados ao Agente Mestre que de forma ordenada (de acordo com a seqüência ditada pelo conjunto de Agentes escravos) as envia para o controlador de tarefas.

Na Figura 5.12 apresenta-se o esquema geral do MAUTOXAD acrescentado do controlador de tarefas e da comunicação entre os agentes. O controlador de tarefas tem a função de executar as tarefas que recebe do Agente Mestre. Ao terminar a execução das tarefas, ele retorna um sinal para o Agente Mestre informando que a execução foi concluída, permanecendo pronto para receber uma outra lista de tarefas (jogada).

No MAUTOXAD, o agente Base (L1 na Figura 5.2) se encarregará de posicionar o plano formado pelo braço, antebraço e a garra do robô na direção do alvo. O plano deve se posicionar sobre a reta que atravessa a base do alvo. Basicamente, o agente base deseja apontar o alvo (casa destino).

O Agente Base é um agente escravo que recebe a consulta do Agente Mestre para

saber qual a seqüência da jogada e o ângulo do movimento. O Agente Base, na sua regra fuzzy (1), utiliza uma variável booleana que indica a ocorrência (SIMETRIA=SIM) ou não (SIMETRIA=NÃO) de simetria, e uma variável real (Equação 5.9) que indica o ângulo formado com a posição destino. Na primeira consulta, de acordo com a existência de simetria ou não (ver Regra fuzzy (1) que é igual ao quadro 5.1), ele informa, via a variável booleana SIMETRIA, se será o primeiro a se mover ou não. Se existir simetria, ele informará ao Agente Mestre com SIMETRIA=SIM que será o primeiro a se mover. Se não existir simetria, a seqüência do seu movimento dependerá da decisão dos demais agentes. Em seguida, o Agente Mestre reenvia a jogada para o Agente Base calcular o número de passos necessários para que o motor faça o giro da base (esquerda ou direita).

```

Regra fuzzy (1)
SIMETRIA=0 //Variável booleana que indica a simetria
if (posicionado) //Garra posicionada sobre uma casa
  then
    if (linha_fonte == linha_destino) //Casas fonte e destino na mesma linha
      then
        if (lado_fonte != lado_destino) //Lados da dama e do rei
          then
            if (coluna_fonte==coluna_destino || coluna_fonte==(R | D) || coluna_fonte==(D | R))
              then SIMETRIA = 1

```

O Agente Antebraço (L2 na Figura 5.2) deve se posicionar para otimizar a posição do antebraço; ele se preocupa com o momento (força vezes distância) criado por ele e o Agente Braço. O Agente Antebraço (L2) é um agente escravo. Ele recebe a consulta da jogada do Agente Mestre, em seguida envia de volta a seqüência de execução dos movimentos dos braços. Se o movimento é na direção que diminui  $\theta_2$ , o antebraço deve se movimentar antes do braço. Definido-se  $\theta_{2f}$  e  $\theta_{2d}$  como os valores de  $\theta_2$  nas posições fonte e destino, a regra fuzzy (2) ilustra a determinação da seqüência de movimento do antebraço e braço.

Regra (2): **if**  $\theta_{2d} > \theta_{2f}$  **then** M23=SIM  
**else** M23=NÃO

A seguir o Agente Mestre envia novamente a jogada para que o Agente Antebraço faça o cálculo do número de passos do motor necessários para movimentar o antebraço para cima ou para baixo.

O Agente Braço (L3 na Figura 5.2) é um agente escravo. Ele recebe a jogada do Agente Mestre e decide o movimento do Agente Tabuleiro (trilho). O Agente Braço deve se posicionar para otimizar a posição do braço, posicionando a garra sobre a peça. Se a posição não for satisfatória ele aciona o Agente Tabuleiro que aproxima ou afasta o tabuleiro da estrutura do robô. Definindo-se  $\theta_{3d}$  como o valor calculado para o ângulo  $\theta_3$  no destino, o valor qualitativo fuzzy de  $\theta_{3d}$  é calculado usando a operação união do conjunto fuzzy. As variáveis lingüísticas de  $\theta_{3d}$  estão apresentadas na Figura 5.13, com a abscissa representada em graus. Definindo-se  $X=\theta_{3d}$ , a função de pertinência ( $\mu_P(x)$ ,  $\cup\mu_M(x)$  e  $\cup\mu_G(x)$ ) vencedora é a que tem maior valor, calculada pela operação fuzzy união  $\theta_{3fuzzy} = \mu_P(x) \cup \mu_M(x) \cup \mu_G(x)$ .  $\theta_{3fuzzy}$  representa uma das funções de pertinência  $\mu_P(x)$ ,  $\cup\mu_M(x)$  ou  $\cup\mu_G(x)$ . A seguir, serão avaliadas as regras fuzzy (3) e (4) que indicarão o movimento do Agente Tabuleiro utilizando as variáveis booleanas aproxima e afasta. As variáveis booleanas devem ser inicializadas como aproxima=NÃO e afasta=NÃO

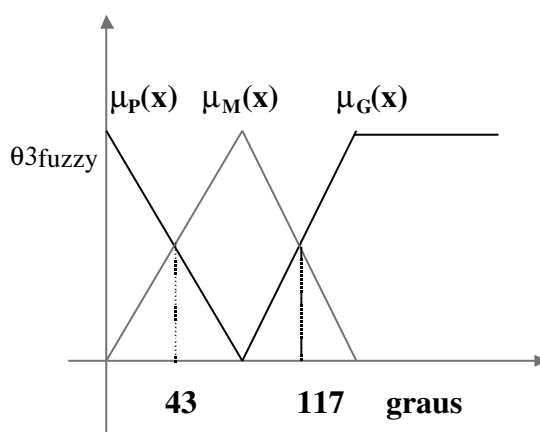


Figura 5.13 - Funções de pertinência

Regra (3): **if**  $\theta_{3fuzzy} == \mu_G(x)$  **then** aproxima=**SIM**

Regra (4): **if**  $\theta_{3fuzzy} == \mu_P(x)$  **then** afasta=**SIM**

O Agente Mestre recebe a seqüência determinada pelo agente e envia a jogada

novamente para o Agente Braço que faz o cálculo do número de passos do motor necessários para movimentar o braço e retorna a tarefa para o Agente Mestre.

O Agente Tabuleiro deve aproximar ou afastar o tabuleiro de xadrez do ROBOXAD. Ele deve ser acionado quando não for possível que a garra se posicione sobre a peça destino. Isto ocorre porque o alcance do braço e do antebraço é restringido a 4 linhas do tabuleiro. O Agente Tabuleiro (L4 na Figura 5.2) é um agente escravo. Ele tem a função de receber a jogada do Agente Mestre e calcular o número de passos necessários para que o motor faça o movimento do tabuleiro de acordo com a jogada (para frente ou para trás).

O Agente Ímã (Garra) é um agente escravo. Ele tem a função de receber a jogada do Agente Mestre, e de acionar e desligar o ímã depois que o braço está posicionado sobre a peça, ele a pega na posição fonte e a solta na posição de destino. As regras fuzzy (5) e (6) ilustram o Agente Ímã.

Regra (5): **if fonte==SIM then agarra=SIM**

Regra (6): **if destino==SIM then agarra=NÃO**

Os agentes descritos acima apresentam uma série de características próprias dos agentes, como foi visto no Capítulo 3. Entre elas destacam-se as seguintes:

a) autonomia, os agentes possuem a capacidade de decidir a seqüência de movimentos de cada um deles e implicitamente têm consciência de qual a ordem de realização dos movimento dos demais agentes; cada um dos agentes escravos são responsáveis pelo cálculo dos passos necessários no motor para executar os movimentos;

b) comunicabilidade, todos os agentes comunicam-se com o Agente Mestre e vice-versa;

c) inteligência, utilização da lógica fuzzy, que deixa o braço mais otimizado;

d) reatividade, os agentes percebem seu ambiente e reage sobre ele;

e) flexibilidade, reside na habilidade dos agentes em escolher dinamicamente as ações e a seqüência de execução das mesmas de acordo com o ambiente, por exemplo a

presença ou não de um obstáculo);

f) cooperatividade, os agentes cooperam com o Agente Mestre para realizar uma determinada jogada;

g) coerência, os agente possui a capacidade realizar seu objetivo ou movimentos mesmo com a presença de um obstáculo;

h) planejamento, para cada jogada são definidas e planejadas as tarefas a serem executadas.

O ambiente no qual estão inseridos os agentes é o tabuleiro de xadrez (representado pelo espaço ocupado por um tabuleiro de 6 linhas x 8 colunas) e pode ser caracterizado da seguinte forma:

a) acessível tendo em vista que os sensores (ainda não existem, mas no caso atual, trata-se do operador) tem acesso a todo o ambiente, verificando o posicionamento das peças a cada jogada;

b) determinístico e não episódico pois cada nova jogada e novo estado dependerá da jogada e estado anterior;

c) estático porque a área do ambiente não se modifica;

d) discreto porque o número de movimentos é fixo.

### **5.3.2 Implementação do MAUTOXAD**

A arquitetura do software do MAUTOXAD, apresenta no seu mais baixo nível um escalonador de tarefas em tempo real e a cada agente é associada uma tarefa. O sistema multi tarefas se encarrega da organização das comunicações entre os agentes além da ativação deles na forma de tarefas. A comunicação entre os agentes, caracterizada pela troca de mensagens entre agentes, é representada por regras, axiomas e variáveis oriundas da lógica fuzzy.

Na implementação na linguagem C++, no nível mais alto, os agentes são representados por funções que são chamadas pelo Agente Mestre. Além disso, como será

visto nas próximas seções, o Agente Mestre se comporta como uma função recursiva durante a ativação dos demais agentes. Isto ocorre quando houver obstáculo entre as casas fonte e destino, enquanto não for encontrado o caminho final o Agente Mestre se auto aciona.

O MAUTOXAD, assim como o AUTOXAD, possui duas estratégias IDENTIFICA e MOVE. A estratégia IDENTIFICA do MAUTOXAD apresentada na Figura 5.14, localiza a posição da peça e da garra (fonte), pega a peça (acionando o ímã). A estratégia MOVE (ver Figura 5.15) movimenta o robô para a posição da peça e a garra solta a peça (desligando o ímã) no lugar determinado (destino).

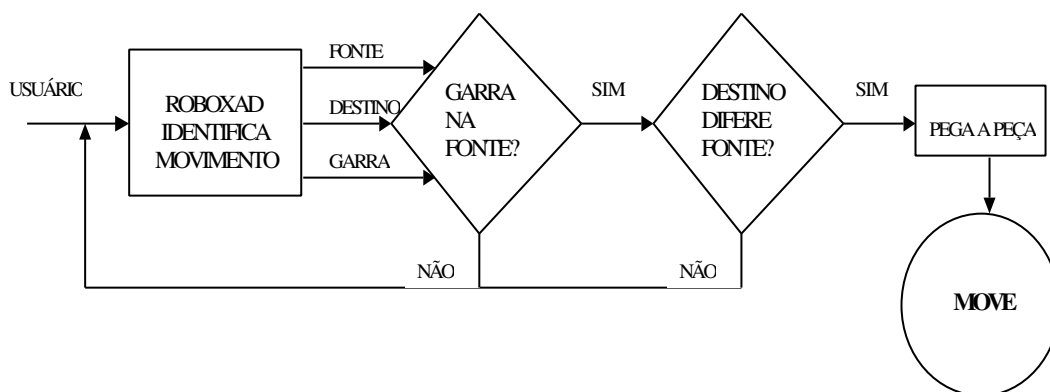


Figura 5.14 - Estratégia IDENTIFICA

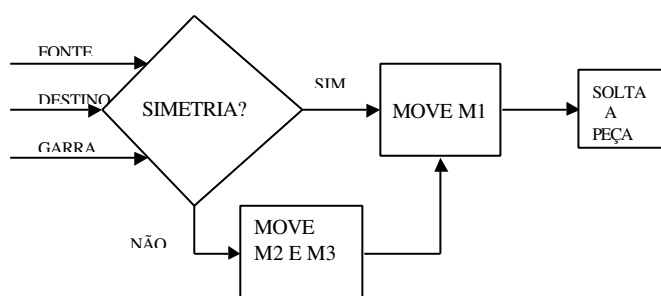


Figura 5.15 - Estratégia MOVE

O MAUTOXAD pode ser considerado como um sistema orientado a tarefas, pois esta é a unidade básica do sistema. Uma tarefa é constituída por uma identificação, pelo número de passos, por uma descrição e pelo nível de satisfação. Os identificadores são os seguintes:

TAREFA\_AG1\_DIREITA



TAREFA\_AG1\_ESQUERDA

TAREFA\_AG2\_SUBIR

TAREFA\_AG2\_DESCER

TAREFA\_AG3\_SUBIR

TAREFA\_AG3\_DESCER

TAREFA\_AGTABULEIRO\_APROXIMA

TAREFA\_AGTABULEIRO\_AFASTA

TAREFA\_AGIMA\_PEGAR

TAREFA\_AGIMA\_SOLTAR

TAREFA\_PARADO

Como se pode ver, estes indicadores informam a semântica da tarefa, isto é, o que ela deverá executar ou fazer. O número de passos de uma tarefa informa a quantidade de passos necessários para execução da tarefa, uma vez que todos os agentes, exceto o Agente Ímã, possuem motores de passos que efetuam os movimentos mecânicos. O terceiro item de uma tarefa, isto é, a descrição, recebe os comentários do agente a respeito dela. Estas mensagens são impressas no componente de saída do sistema. O quarto e último item da tarefa, o grau de satisfação, armazena o nível de satisfação da tarefa.

Para ilustrar o funcionamento do MAUTOXAD, principalmente a tomada de decisão entre diferentes estratégias de movimento, inseriu-se mais uma peça no tabuleiro de xadrez. Quando esta peça se encontra entre a posição fonte e destino, funciona como um obstáculo. Para realizar o desvio desse obstáculo apresenta-se abaixo a descrição das estratégias das jogadas.

No sistema são possíveis jogadas sem obstáculos (S) e com obstáculo (C), podendo a jogada (L) ser com ou sem obstáculo. A seguir, apresenta-se como devem ser feitas essas jogadas e como se deve realizar o desvio do obstáculo. Na Figura 5.16 apresenta-se um tabuleiro com 12 casas utilizado para descrever os possíveis movimentos das peças

num tabuleiro. Nesse tabuleiro são indicadas as posições fonte (F) e de destino (D) das peças e a posição da peça extra (X).

Jogada Simples (S): este tipo de jogada acontece quando o Agente Mestre verifica que não existe obstáculo (X) em um quadro do tabuleiro onde se encontram as posições Fonte (F) e destino (D), então a jogada é realizada de forma simples, sem desvio do obstáculo (ver Figura 5.22).

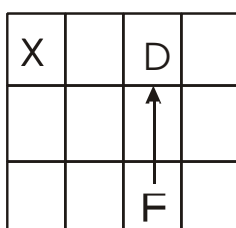


Figura 5.16 - Jogada Simples (S)

Jogada em L (L): o nosso sistema não trabalha com deslocamentos na diagonal, portanto este tipo da jogada acontece quando a posição destino (D) não se encontra alinhada à posição fonte (F) nem em X, nem em Y. Neste caso o movimento executado será em “L”, ocorre um deslocamento de uma casa ou mais, dependendo da posição destino e finaliza executando uma jogada do tipo simples (S) (Figura 5.17).

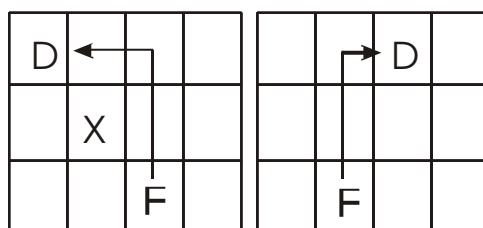


Figura 5.17 - Jogada em L (L)

Jogada com obstáculo (C), é o caso em que o Agente Mestre verifica um obstáculo alinhado ou em X ou em Y em relação às posições fonte e destino. Neste caso ocorre inicialmente o deslocamento de uma casa, o que permite uma jogada em L (L), que por sua vez, leva a uma jogada simples (S) (Figura 5.18 e 5.19).

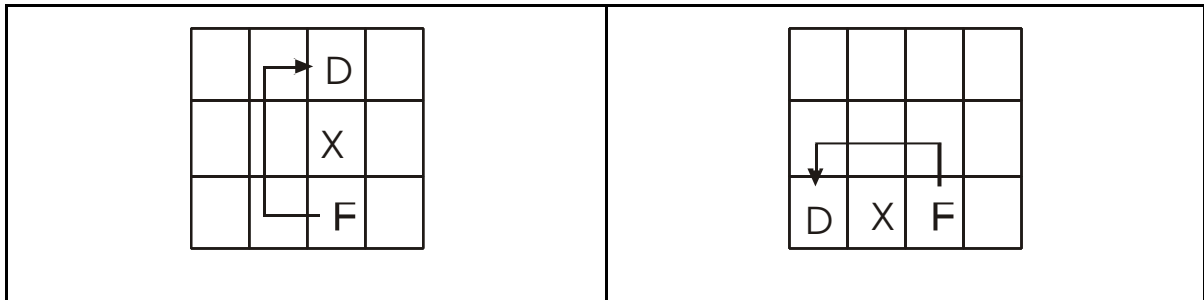


Figura 5.18 - Jogada em C (C) em Y

Figura 5.19 - Jogada em C (C) em X

A verificação da ocorrência de obstáculo é feita com as regras fuzzy (7)(8)(9) e (10). Define-se linha fonte como Lf, coluna fonte como Cf, linha destino como Ld, coluna destino Cd, coluna da peça extra Cx, linha da peça extra Lx.

Regra (7): **if** Cf ==Cd = Cx & Lx > Lf & Ld > Lx **then** obst=SIM

Regra (8): **if** Cf ==Cd = Cx & Lx > Ld & Lf > Lx **then** obst=SIM

Regra (9): **if** Lf ==Ld = Lx & Cx > Cf & Cd > Cx **then** obst=SIM

Regra (10): **if** Lf ==Ld = Lx & Cx > Cd & Cf > Cx **then** obst=SIM

Na Figura 5.20 apresenta-se o esquema geral de acionamento dos agentes movimentando o braço e desviando dos obstáculos. Pode-se observar que a função mover() é recursiva, se houver obstáculo ele será desviado.

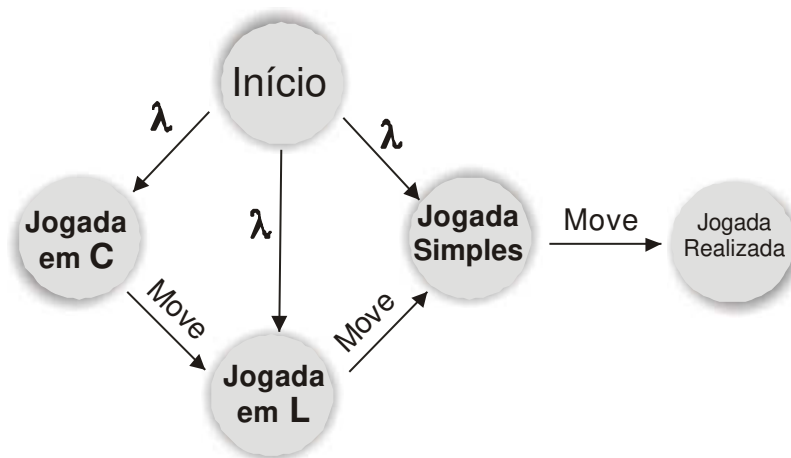


Figura 5.20 - Esquema de Jogadas do MAUTOXAD

### 5.3.1.2 Interface entre o usuário e o MAUTOXAD

Na Figura 5.21 apresenta-se a tela de trabalho do MAUTOXAD. Nesta tela o usuário se comunica com o MAUTOXAD e observa as informações a respeito dos movimentos dos braços do robô.



Figura 5.21 - Tela principal

A tela principal do MAUTOXAD é dividida em diversas partes. Na parte central da Tela encontram-se:

- *Área do tabuleiro:* representada pelo tabuleiro com 64 casas, é a área onde o usuário observa o posicionamento das peças e escolhe a jogada que deseja fazer. Observe-se que as peças sobre o tabuleiro são diferenciadas pela cor de acordo com a legenda no canto esquerdo inferior da tela (Extra, Fonte e Destino).
- *Área de movimentos:* representada pelo quadrado em branco, é a área onde o MAUTOXAD mostra os movimentos e as ações executadas pelos motores de passo.
- *Peça extra:* na parte inferior da região central, é a área onde é apresentada a localização da peça extra no tabuleiro. No exemplo da Figura 5.21, a peça extra

encontra-se no lado da dama, especificamente localizada na quarta linha e na primeira coluna do lado A do tabuleiro. O botão Mudar possibilita modificar essa localização.

No lado esquerdo da tela têm-se as seguintes áreas:

- **Posição Fonte:** na parte superior, a área fornece dados sobre o posicionamento da peça no início da jogada, onde:
  - *Casa:* representa a casa fonte da peça (Lado Rei ou Dama, Linha e Coluna);
  - *R:* representa a distância entre o eixo base e a primeira linha do tabuleiro (casa fonte).
  - *X:* representa a metade do comprimento da linha ou da coluna de cada casa do tabuleiro.
  - *Ângulo Fonte:* representa a ângulo da posição fonte da peça;
  - *Deslocamento:* representa a deslocamento da posição fonte da peça..
  
- **Posição Destino:** na parte central, fornece dados sobre o posicionamento da peça no fim da jogada, onde:
  - *Casa:* representa a casa de destino da peça (Lado Rei ou Dama, Linha e Coluna);
  - *R:* representa a distância entre o eixo base e a casa de destino.
  - *X:* representa a metade do comprimento da linha ou da coluna de cada casa do tabuleiro.
  - *Ângulo Destino:* representa a ângulo da posição de destino da peça.
  - *Deslocamento:* representa a deslocamento da posição de destino da peça.

- **Movimento:** na parte inferior, fornece dados sobre o movimento que vai ser realizado

- *R:* representa a distância entre a casa do destino e a casa fonte da peça;
- *Teta:* representa a diferença entre os ângulos de destino e fonte da peça;
- *Deslocamento do movimento:* representa a diferença entre o deslocamento da posição de destino e da posição fonte da peça;

No lado direito da tela têm-se as seguintes áreas:

- *Agente Mestre:* na parte superior, apresentam-se dois botões que podem ser clicados pelo usuário. Se for clicado o botão Mover será executada a jogada planejada. O botão Parar interrompe a jogada.
- Os demais campos abaixo do Agente Mestre apresentam os números dos passos a ser movidos de acordo com a jogada para cada um dos Agentes escravos. Além disso, os botões de direção (< >) permitem mover, caso seja necessário, manualmente passo a passo para esquerda e para direita (agente Base), para cima e para baixo (agentes Braço e Antebraço) ou para frente e para trás (agente Tabuleiro) No caso do Agente Garra permite ativar ou desativar o ímã.

### **5.3.3 Exemplos dos movimentos do MAUTOXAD**

A área de trabalho total do ROBOXAD, limitada pelas características mecânicas do robô, é de 6 linhas por 8 colunas. Para simplificação na programação do MAUTOXAD definiram-se duas áreas no tabuleiro de xadrez. As áreas foram denominadas de tabuleiro A e Tabuleiro B. O Tabuleiro A representa 4 linhas por 8 colunas. O Tabuleiro B representa 2 linhas por 8 colunas (ver Figura 5.22).

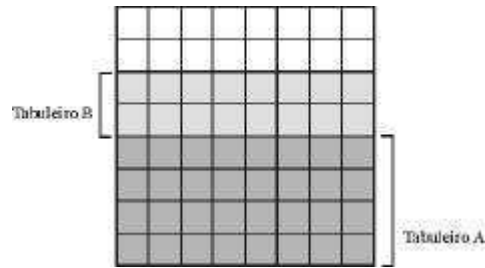


Figura 5.22 - Divisão do tabuleiro e área de alcance

### 5.3.3.1 Jogada sem obstáculo

A jogada consiste em mover a peça da posição fonte (REI 1 1 A, lado do rei, primeira linha e primeira coluna ou R1R em xadrez) do Tabuleiro para a posição destino (REI 3 1 A, ou R3R em xadrez) do Tabuleiro A. Observa-se na Figura 5.23 abaixo que a peça extra (obstáculo) está fora da área de movimentos, portanto este movimento é considerado como uma Jogada Livre (S).

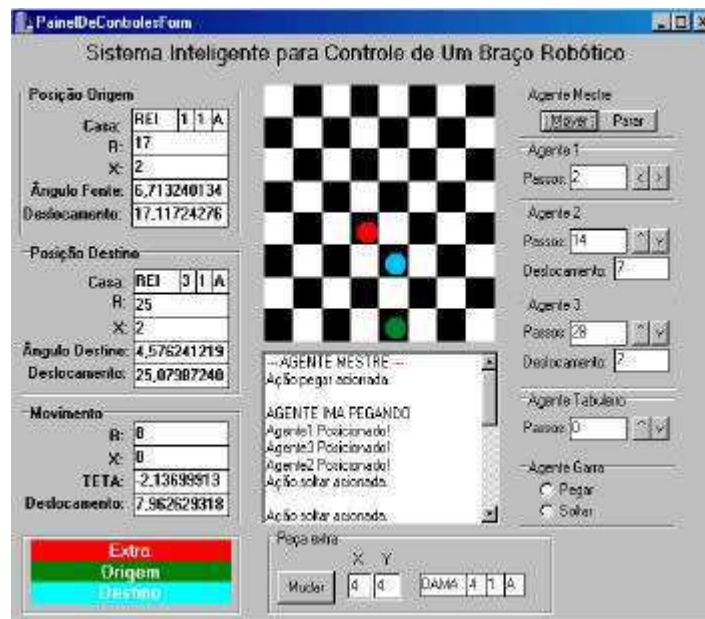


Figura 5.23 - Jogada sem obstáculo

As etapas da jogada são as seguintes:

- O Agente Mestre recebe a jogada do usuário;

- O Agente Mestre verifica, de acordo com o fluxograma apresentado na Figura 5.14, que a casa destino é diferente da casa fonte e posiciona o ímã sobre a peça;
- O Agente Mestre consulta os Agentes escravos sobre a ordem em que eles efetuarão os movimentos;
- O Agente Base verifica que não existe simetria, para isso utiliza a regra fuzzy (1),
- Os demais agentes, incluindo o Agente Base definem a sua ordem de execução de movimentos;
- Os agentes enviam a ordem de execução de movimentos para o Agente Mestre; (Agente Ímã, Agente Base, Agente Braço, Agente Antebraço, Agente Ímã);
- O Agente Mestre envia a jogada para os agentes, para que efetuem os cálculos necessários na realização dos movimentos;
- Em seguida, cada agente envia de volta para o Agente Mestre a tarefa (cálculos necessários para efetuar o movimento: Agente Base (passos=2); agente Antebraço (passos=14 e deslocamento=7); Agente Braço (passos=28 e deslocamento=7); Agente Tabuleiro (passos=0). O Agente Ímã pegar e depois soltar a peça; Neste caso, A agente Tabuleiro não trabalha. Os agentes braço e antebraço utilizam a Equação 5.2 para calcular o deslocamento.
- O Agente Mestre recebe as tarefas dos Agentes escravos, as ordena de acordo com a seqüência dos movimentos e envia a lista de tarefas para o controlador de tarefas;
- O Controlador de tarefas tem a função de processar o movimento da jogada, ao terminar a execução ele retorna um sinal informando o fim da execução.

### **5.3.3.2 Jogada com Obstáculo**

A jogada consiste em mover a peça da posição fonte (R 1 1 A, R1R em xadrez) do Tabuleiro A para a posição destino (R 2 1 B, R2R em xadrez) do Tabuleiro B. Observa-se



na Figura 5.24 abaixo que a peça extra (obstáculo) está dentro da área de movimentos e alinhados em Y em relação às posições fonte e destino (R 4 1, R4R em xadrez) do Tabuleiro A, portanto é considerada como uma Jogada com Obstáculo. A Figura abaixo apresenta o resultado do movimento.



Figura 5.24 - Início da Jogada com obstáculo

As etapas da jogada são as seguintes:

- O Agente Mestre recebe a jogada do usuário apresenta a tela da Figura 5.24. Após o usuário clicar sobre o botão Mover é iniciada a ação da jogada cujo tela resultante é apresentada na Figura 5.25. O MAUTOXAD apresenta a mensagem \_\_ AGENTE MESTRE -- na área de movimentos. Após alguns instantes é apresentada a mensagem Ação pegar acionada.
- O Agente Mestre verifica que a casa destino é diferente da casa fonte e posiciona o ímã sobre a peça, e apresenta a mensagem AGENTE ÍMÃ PEGANDO.
- O Agente Mestre consulta os Agentes escravos sobre a ordem em que eles efetuarão os movimentos;

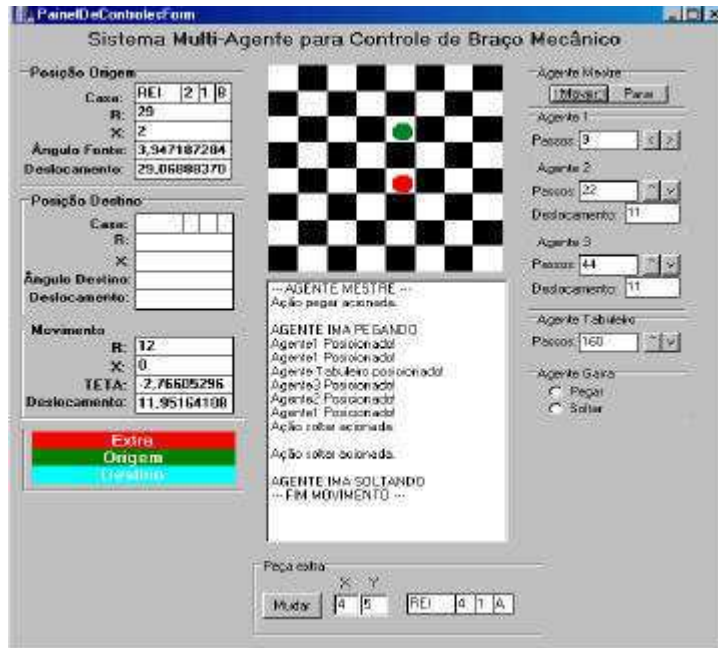


Figura 5.25 - Fim da Jogada com obstáculo

- O Agente Base verifica que não existe simetria;
- Os demais agentes, incluindo o agente base definem a sua ordem de execução de movimentos;
- Os agentes enviam a ordem de execução de movimentos para o Agente Mestre; (Agente Ímã, Agente Base, Agente Tabuleiro, Agente Braço, Agente Antebraço, Agente Base, Agente Ímã);
- O Agente Mestre envia a jogada para os agentes, para que efetuem cálculos necessários à realização dos movimentos;
- Em seguida, cada agente envia de volta para o Agente Mestre a tarefa (cálculos necessários para efetuar o movimento: Agente Base (passos=9); agente Antebraço (passos=22 e deslocamento=11); Agente Braço (passos=44 e deslocamento=11); Agente Tabuleiro (passos=160);
- O Agente Mestre recebe as tarefas dos Agentes escravos, as ordena de acordo com a

seqüência dos movimentos e envia a lista de tarefas para o controlador de tarefas;

- O Controlador de tarefas tem a função de processar o movimento da jogada, ao terminar a execução ele retorna um sinal informando o fim da execução. As seguintes são apresentadas na área de movimento:

- Agente 1 Posicionado!, indica que houve o movimento da base do robô durante o movimento em L para escapar do obstáculo, passos = 9. O movimento é feito na seqüência apresentada na Figura 5.26.

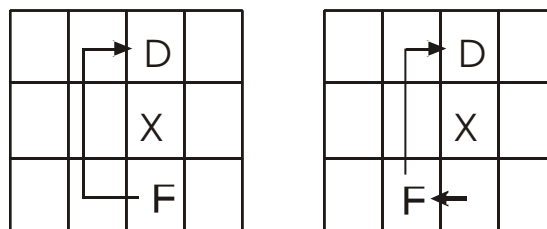


Figura 5.26 - Início da seqüência de movimento

- Agente Tabuleiro posicionado. Como o ângulo do braço teve um valor fuzzy grande ( $\mu_G(X)=SIM$ ) o agente tabuleiro foi acionado para aproximar o tabuleiro duas linhas do corpo do robô. Duas linhas do tabuleiro passos= 160.
- Agente 3 Posicionado. O agente antebraço indicou que para este movimento o antebraço deve se mover antes do braço. (ver regra fuzzy (2)) Deslocamento=11, passos=44.
- Agente 2 Posicionado. Anteriormente o agente braço havia indicado o movimento do tabuleiro. Deslocamento=11, passos= 22.
- Agente 1 Posicionado.

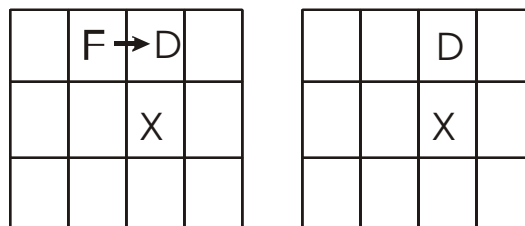


Figura 5.27 - Fim da seqüência de movimento.

- Ação soltar acionada! O ímã é desenergizado e a peça é solta sobre a casa destino.
- AGENTE ÍMÃ SOLTANDO
- -- FIM MOVIMENTO --

O MAUTOXAD permanece esperando a próxima jogada.

## 5.4 Conclusão

Neste capítulo apresentou-se o sistema inteligente de controle de um protótipo de um robô com cinco graus de liberdade (ROBOXAD). A partir do Sistema Inteligente (AUTOXAD) já desenvolvido com a Lógica Fuzzy propôs-se uma arquitetura baseada em agentes inteligentes (MAUTOXAD) que é capaz de movimentar peças de xadrez sobre um tabuleiro desviando de obstáculos.

O uso da Lógica Fuzzy, além de utilizar variáveis lingüísticas semelhantes às empregadas pelos operadores de robôs, permitiu a aplicação do robô em sistemas tridimensionais sem o conhecimento formal da sua posição e das peças do tabuleiro de xadrez no espaço.

# CAPÍTULO 6

## 6. CONCLUSÃO

Esta dissertação teve como ênfase a construção de um sistema de controle inteligente de um braço robótico utilizando uma abordagem de agentes inteligentes baseado em regras fuzzy. Foram apresentados os detalhes da implementação do MAUTOXAD e alguns resultados experimentais no controle do manipulador robótico para o posicionamento de uma garra magnética que move uma peça e desvia de obstáculos sobre um tabuleiro de xadrez.

O protótipo ROBOXAD possui uma estrutura relativamente simples e foi desenvolvido para ser utilizado como base de testes dos sistemas desenvolvidos no NEUROLAB. Com este protótipo foi possível implementar e testar o MAUTOXAD, sistema desenvolvido com o paradigma de agentes inteligentes.

### 6.1 Contribuições

A utilização da abordagem de agente inteligente proporcionou uma maior organização ao sistema principalmente no que diz respeito à distribuição de tarefas, de forma que cada agente do sistema é responsável por um determinado grupo de tarefas e conseqüentemente por cada uma das partes do ROBOXAD. A extensão do sistema com a utilização de agentes inteligentes permitiu que o braço robótico desviasse de obstáculos, ou seja de outras peças no tabuleiro, o que será de fundamental importância para o projeto de visão que está sendo desenvolvido paralelamente.

A utilização de regras fuzzy permite, através de uma análise lingüística dos ângulos, otimizar o posicionamento dos braços, tabuleiro, ímã e determinar quais agentes e em que momento deverão ser acionados. Portanto, a utilização da técnica Lógica Fuzzy proporcionou ao sistema características de inteligência e autonomia.

O SI foi implementado utilizando a linguagem de programação C++ e a ferramenta C++ Builder da Borland. Procurou-se simplificar o código de forma que seja possível ser utilizado em aplicações futuras, como por exemplo na junção com o sistema de visão que está sendo desenvolvido paralelamente para o ROBOXAD.

A nova interface do sistema proporciona uma maior facilidade de interação. É possível definir a jogada através de manipulação direta com as peças no tabuleiro (na interface), no entanto, percebe-se ainda alguns pontos em que poderia ser melhorada, principalmente no que diz respeito à realimentação para o usuário.

A implementação de um trilho no tabuleiro do ROBOXAD, permitiu que a área de trabalho do MAUTOXAD fosse ampliada de 3 linhas por 6 colunas para 6 linhas por 8 colunas. Quando o braço alcança seu ângulo máximo de trabalho o motor de passo que movimenta o tabuleiro sobre um trilho é acionado movendo-se para frente/trás, permitindo que a jogada possa ser realizada numa área maior.

## **6.2 Dificuldades Encontradas**

No decorrer deste trabalho, foram encontradas algumas dificuldades:

- Houve uma certa dificuldade em compreender o programa fonte do AUTOXAD principalmente pela complexidade do código.
- A placa de interface controladora queimou diversas vezes devido a problemas de tensão. Foi necessário uma adaptação na quantidade de energia recebida. Este problema atrasou os testes e consequentemente o andamento do trabalho.
- Durante os testes, percebeu-se dificuldades relacionadas à montagem do ROBOXAD, que por ser construído em sua maior parte de material de sucata e num processo incremental, apresenta uma série de limitações. A estrutura do robô

é muito leve, o que impede de sustentar pesos maiores que 10g, o que faz com que muitas vezes durante a jogada a peça caia. Portanto, não é possível se obter precisão nos movimentos e conseqüentemente nas jogadas realizadas. No caso de várias jogadas seqüenciais, ocorre acumulação de erro, o que resulta no erro das jogadas, o que limita a confiabilidade do sistema.

### **6.3 Sugestões para Trabalhos Futuros**

Após a finalização deste trabalho, é possível sugerir alguns trabalhos que podem complementar e melhorar o sistema MAUTOXAD:

- Um novo protótipo do ROBOXAD, projetado especificamente para o Sistema desenvolvido. Seria interessante que o braço robótico pudesse alcançar todo o tabuleiro sem a necessidade que este se movesse.
- Um sistema de visão que fosse capaz de resolver o problema de erros acumulativos.
- Desenvolver uma interface mais amigável.

# REFERÊNCIAS BIBLIOGRÁFICAS

- [ALV97] ÁLVARES, L.O.; SICHMAN, J.S. **Introdução aos Sistemas Multiagentes**, EINE'97 - I Escola de Informática SBC - Edição Nordeste, página143.
- [AMI93] AMIROUCHE, F. M. L. **Computer – Aided Design and Manufacturing**. Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [ÄST89] ÄSTROM, K. J.; HÄGGLUND, T. **Automatic Tuning of PID Controllers**. Instrument Society of America, USA, 1989.
- [AUE95] AUER, K. **Agentes**. [doc WWW – 01/12/97] URL: <http://www.pcug.org.au/~kauer/project/main.htm>, 1995.
- [BEL95] BELGRAVE, M. **The unified agent architecture**. A white paper. [doc WWW – 25/11/97] URL: [http://www.ee.mcgill.ca/~belmarc/uaa\\_paper.html](http://www.ee.mcgill.ca/~belmarc/uaa_paper.html). 1995.
- [BEZ94] BEZDEK, J. **Fuzzy models – What are they, and why?**. In: Marks II, Robert J. Fuzzy logic technology and applications. New York: IEEE Technical activities board, 1994.
- [BRO91] BROOKS, R. A. **Intelligence without reason**. In Computers and Thought lecture: Proceedings of IJCAI-91, Sydney, 1991.
- [CAV95] CAVALCANTI, J. H. F. , FERNEDA, E. **Intelligent control of an inverted pendulum – training and evolution**. In: SEMISH'95 – Conferência Latino-americana de Informática, 21ª, 1995. Anais. Canelas RS, 1995. 2v., v.1, p.627-637, 1995.
- [CAV99] CAVALCANTI, J. H. F., ALSINA, P. J., FERNEDA, E. **Posicionamento de um pêndulo invertido usando algoritmos genéticos**. SBA Controle & Automação, São Paulo, v.10, n.1. p.31-38, 1999.
- [CRA86] CRAIG, J. J. **Introduction to Robotics Mechanics & Control**. Addison -



Wesley Publishing Company, USA, 303p. 1986.

- [DEM95] DEMAZEAU, Y. **From interactions to collective behaviour in Agent-Based System**. Saint Malo: European Conference on Cognitive Science, 1995.
- [FER91] FERBER, J. , ERCEAU J. "**L'Intelligence Artificielle Distribuée**". Revista La Recherche 233 - Vol. 22. Juin , pp 750-758. 1991.
- [FER98] FERREIRA, J. R. S., ALSINA, P. J., CAVALCANTI, J. H. F. **Escalonador inteligente de tarefas aplicado à robótica**. In: INDUSCON'98 – CONFERÊNCIA DE APLICAÇÕES INDUSTRIAIS, 3º, 1998. Anais. São Paulo: Escola Politécnica da USP, 1v., p.117-120. 1998.
- [FER99] FERREIRA, C. N. M. A. **Posicionamento Inteligente de um Manipulador Robótico**, Dissertação de Mestrado em Ciências da Computação, Departamento de Sistemas e Computação do CCT/UFPB, 23/08/1999
- [FER99a] FERREIRA, C. N. M. A., CAVALCANTI, J. H. F., ALSINA, P. J., FERREIRA, J. R. S. **Posicionamento Inteligente de um Manipulador Robótico**., COBEM'99. Águas de Lindóia – SP 1999.
- [FER99b] FERREIRA, J. R. S., CAVALCANTI, J. H. F. **Interfaces controladoras de motores**. 25p. Relatório técnico N.º DSC/001/99 (Biblioteca setorial do Departamento de Sistemas e Computação) – Universidade Federal da Paraíba. Campina Grande: 1999
- [FER99c] FERREIRA, J. R. S., CAVALCANTI, J. H. F, ALSINA, P. J. **Intelligent tasks scheduler**. In: IJCNN'99 – INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 1999. Proceedings. Washington, D.C. USA: Renascence Hotel, . 1999.
- [FER99d] FERREIRA, J. R. S., CAVALCANTI, J. H. F. **Interface controladora de motores de corrente contínua**. Revista Interação (Centro de Ciências e Tecnologia da UFPB), Campina Grande, 1999.
- [FER99e] FERREIRA, J. R. S., CAVALCANTI, J. H. F., ALSINA, P. J. **An**

**application using intelligent tasks scheduling and knowledge acquisition.**  
In: CARS&FPF'99 – INTERNATIONAL CONFERENCE ON CAD/CAM,  
ROBOTICS AND FACTORIES OF THE FUTURE, 15°, 1999.

- [FRA96] FRANKLIN, S., GRAESSER, A. **Is it an agent, or just a program?: A taxonomy for autonomous agents.** [doc WWW – 28/11/97] URL: <http://www.msci.memphis.edu/~franklin/AgentProg.html>, 1996.
- [GIL96] GILBERT, D., MANNY, A. **Intelligent agent strategic.** [doc WWW] URL: <http://activist.gpl.ibm.com:81/WhitePaper/ptc2.html>, 1996.
- [GOM92] GOMIDE, F., ROCHA, A., ALBERTOS, P. **Neurofuzzy Controllers-IFAC- LCA '92**, Viena, Austria.1992.
- [GOM94] GOMES, S. C. P. **Modelagem e Controle de Manipuladores Flexíveis.** 10° Congresso Brasileiro de Automática & 6° Congresso Latino Americano de Controle Automático. Rio de Janeiro, pp. 1010-1015. 19 a 23 de Setembro de 1994.
- [GUP96] GUPTA, M. M., SINHA, N. K. **Intelligent control systems.** New Jersey: IEEE Press, 820p. 1996.
- [KLI88] KLIR, G. J., FOLGER, T. A. **Fuzzy sets, uncertainty, and information.** New Jersey: Prentice Hall, 355p. 1988.
- [LEM96] LEMON, B., PYNADATH, D., TAYLOR, G., et al. **Cognitive architectures.** [doc WWW – 28/11/97] URL: <http://krusty.eecs.umich.edu/cogarch4/toc.html>, 1996
- [MAE91] MAES P. "Situated Agents Can Have Goals." **Designing Autonomous Agents.** Ed: P. Maes. MIT-Bradford Press, 1991.
- [MAE94] MAES, P. **Agents that Reduce Work and Information Overload.** [on line] Disponível na Internet via WWW. URL: <http://pattie.www.media.mit.edu/people/pattie/CACM-94/CACM-4.P1.html>, 1994.

- [MAM74] MAMDAMI, E. H. "**Application of fuzzy algorithms for control of simple dynamic plant**", Proceedings of the Institution of Electrical Engineers, vol. 121, no. 12, pp. 1585-8, 1974.
- [NIS95] NISSEN, M. **Intelligent agents: A technology and business application** [doc WWW] URL: <http://haas.berkeley.edu/~heilmann/agents>, 1995.
- [OLI96] OLIVEIRA, F. M. **Inteligência Artificial Distribuída**. In: IV Escola Regional de Informática, SBC, SC, 1996.
- [PAR96] PARAISO, E. C. **Sistema Multi-Agente para Monitoração e Controle de Processo**. [doc WWW – 18/11/97] URL: <http://dainf.cefetpr.br/~paraiso>, 1996.
- [POR89] PORTER, B. **Issues in the Design of Ontelligent Control Systems**, IEEE Control Systems Magazine, January pp. 97-99.1989.
- [RUS95] RUSSEL, S., NORVIG, P. **Artificial Intelligence: A Modern Approach** Prentice Hall, 1995.
- [SAL90] SALANT, M. A. **Introdução à Robótica**. São Paulo: McGraw-Hill, 1990.
- [SAL97] SALES, E. F. Jr. **Sistema de controle inteligente para um elo robótico**. Campina Grande: Dissertação (Mestrado em Informática; Inteligência Artificial) – Universidade Federal da Paraíba.1997
- [SPO89] SPONG, M. W., VIDYASAGAR, M. **Robot Dynamics and Control**. John Wiley & Sons, Inc. 1989.
- [TAF95] TAFNER, M.A., et al. **Redes Neurais Artificiais: Introdução e Princípios de Neurocomputação**. Blumenau: Ed. FURB, 1995.
- [WIJ97] WIJFFEIS, T. **Os Motores de Passo**. Revista Elector, 39-45, 1997.
- [WOO94] WOOLDRIDGE, M., JENNINGS, N. **Intelligent Agents: Theory and Practice**,1994.
- [WOO95] WOOLDRIDGE, M., JENNINGS, N. **Theories, Architectures, and**

**Languages:** A survey. [on line] Disponível na Internet via WWW. URL:  
<http://www.cerc.wvu.edu/~juggy/ia/notes/Mar16.notes>,1995.

- [WOO96] WOOLDRIDGE, M., JENNINGS, N. **Software agents**. IEE Review, 1996.
- [ZAD65] ZADEH, L. A. **Fuzzy Sets** Inf. Control, Vol. 8, pp. 338-353. 1965
- [ZAD88] ZADEH, L. A. **Fuzzy Logic**, IEEE Computer Magazine, pp.83-93.  
Publicado também In: Anderson, J.A.; Rosenfeld, E. (Ed.) Neurocomputing  
Foundations of Researchs, pp.177-194, The MIT Press, Cambridge, USA.  
1988
- [ZAD94] ZADEH. L. A. **The Role of Fuzzy Logic in Modeling, Identification and Control**, 15 (3):191-203, 1994.

# APÊNDICE A: Motor de Passo

## A.1 Introdução

O motor de passo é um mecanismo que converte energia elétrica em movimento. Ele é controlado através de pulsos digitais, o que possibilita o seu deslocamento por passo. Com o passar dos anos houve um aumento na popularidade deste motor, principalmente pelo seu tamanho e custo reduzidos e também a total adaptação para ser controlado digitalmente. Outra vantagem do motor de passos em relação aos outros motores é a estabilidade. Quando se quer obter uma rotação de uma fração de uma rotação representada em graus ( $1 \text{ grau} = 2\pi \text{ rad}$ ), calcula-se o número pulsos referente à rotação, o que possibilita uma boa precisão no movimento do eixo do motor.

Há alguns anos atrás os motores utilizados eram motores de corrente contínua. Eles tinham pouca precisão, geralmente passavam do ponto (rotação do eixo do motor maior do que a desejada) e para retornar à posição desejada necessitavam de uma realimentação negativa. Os motores de passos possuem pouca inércia diminuindo a sua possibilidade de passarem do ponto desejado. Mesmo assim, se eles passarem da posição desejada eles podem facilmente reposicionados utilizando uma lógica digital.



Figura A.1 - Motor de passo

## A.2 Funcionamento dos motores de passo

Normalmente os motores de passo são projetados com enrolamento de estator polifásico. O número de pólos é determinado pelo passo angular desejado por pulsos de entrada. Eles possuem três estados de operação: parados, ativados com rotor parado e girando em etapas. Na Figura A.2 esta ilustrada uma representação esquemática de um tipo de motor de passo. O estator é composto de quatro pólos eletromagnéticos e o rotor é um ímã permanente de dois pólos.

Se os pólos eletromagnéticos do estator forem ativados de modo que o pólo 3 seja N (Norte magnético) e o pólo 1 seja S, então o alinhamento do rotor será como ilustrado na Figura A.2 (a). Se o estator for excitado de modo que o pólo 4 seja N e o pólo 2 seja S, o rotor fará um giro de noventa graus no sentido horário como apresentado na Figura A.2 (b).

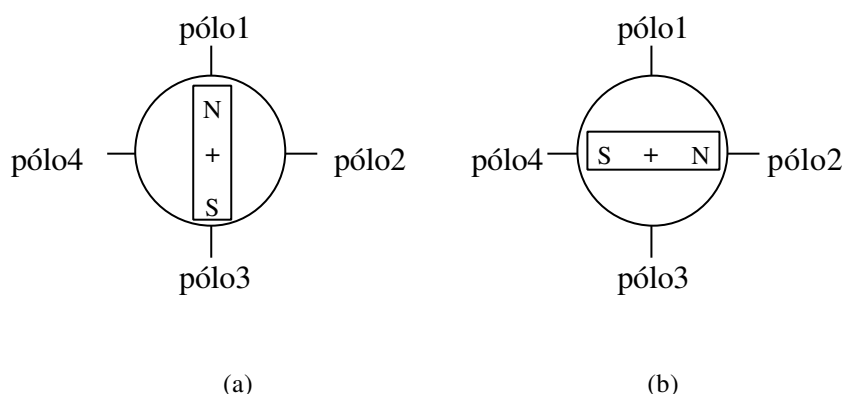


Figura A.2 - Motor genérico de quatro fases

No modo de acionamento normal, o rotor se posiciona exatamente sobre o local designado para cada passo, percorrendo quatro passos por ciclo. A seqüência de acionamento pode ser obtida excitando-se uma fase por vez [FER99].

Os motores de passo têm alimentação externa. Os pulsos na entrada do circuito de alimentação fornecem correntes aos enrolamentos certos para fornecer o deslocamento desejado.

## A.3 Características dos motores de passo

O torque do motor de passo depende da frequência aplicada ao circuito de alimentação. Os motores de passo possuem algumas características básicas:

Quanto maior a frequência, menor o torque, porque o rotor tem menos tempo para mover-se de um ângulo para outro;

A faixa de partida deste motor é aquela na qual a posição da carga segue os pulsos sem perder passos;

A faixa de giro é aquela na qual a velocidade da carga também segue a frequência dos pulsos mas com uma diferença: não pode partir, parar ou inverter, independente do comando utilizado.

## A.4 Tipos dos Motores de Passo

**RELUTÂNCIA VARIÁVEL:** Apresenta um rotor com muitas polaridades construído a partir de ferro doce, possui também em estator laminado. Por não possuir imã, quando energizado apresenta torque estático nulo. Tendo assim baixa inércia de rotor não pode ser utilizado com carga inercial grande.

**IMÃ PERMANENTE:** Apresenta um rotor de material alnico ou ferrite e é magnetizado radialmente. O seu torque estático não é nulo.

**HÍBRIDO:** É uma mistura dos dois anteriores e apresenta rotor e estator multidentados. O rotor é de imã permanente e magnetizado axialmente. Apresenta grande precisão ( 3% ), boa relação torque/tamanho e ângulos pequenos ( 0,9 e 1,8 graus ).

Para que o rotor avance um passo é necessário que a polaridade magnética de um dente do estator se alinhe com a polaridade magnética oposta de um dente do rotor.

## **A.5 Aplicações dos motores de passo**

Como os motores de passos têm movimentos precisos, qualquer equipamento que necessite de precisão no movimento podem utilizar esses motores.

Pode-se citar, pôr exemplo, o controle de micro câmeras num circuito interno de vigilância, em clínicas radiológicas no auxílio de operadores para os mesmos orientarem o posicionamento das pessoas submetidas a uma radiografia, posicionamento de uma mesa de trabalho em duas dimensões, furação automática de acordo com instruções em fita sobre as posições dos furos.

A utilização dos motores de passo é muito ampla, pode-se dizer que vai desde o controle de máquinas industriais (robôs) até pequenas demonstrações num curso de robótica .

## **A.6 Vantagens e Desvantagens dos motores de passo**

Em relação aos motores de corrente contínua os motores de passos apresentam evidentes vantagens, como tamanho e custo reduzidos, total adaptação a lógica digital (o que permite o controle preciso da velocidade direção e distância), características de bloqueio, pouco desgaste e dispensa realimentação.

São poucas as desvantagens mais elas existem: má relação potência volume e principalmente controle relativamente complexo.



# APÊNDICE B: Interface controladora de motores de passo

## B.1 Introdução

Os motores de passo são considerados atuadores de fundamental importância devido os mesmos terem total adaptação à lógica digital [WIJ97]. Este tipo de motor é muito utilizado quando se deseja o posicionamento de determinados dispositivos com precisão. O NEUROLAB utiliza uma interface controladora de motores de passos, que permite o controle de até 4 motores de passo. A base do circuito dessa interface foi projetada no Departamento de Engenharia Elétrica da Universidade Federal da Paraíba para o NEUROLAB, composta basicamente de dois módulos: Um módulo de CIs para demultiplexação dos sinais e um módulo analógico para o fornecimento da tensão de 12V necessária para o acionamento dos motores de passo [FER99b].

## B.2 A interface controladora completa.

A interface controladora de motores de passo, semelhante a interface de motores cc, possui além dos módulos descritos anteriormente, outros componentes listados na tabela B.1 abaixo.

Componente	Valor	Qtde
Resistor	1K	16
Circuito integrado	SN74L S273N	2
CI regulador de tensão	7805	1
Capacitor eletrolítico	220mF	1
Transistor de potência	TIP 122	16
Resistor	280R	16
Diodo	1N4007	16
Conector centronic	Fêmea	1

Conector DB25	Fêmea	1
Fusível	3A	1
LED		1
Resistor	1K	1
Conector para energia		1

Tabela B.1 - Tabela de relação dos componentes utilizados na interface.

A Figura B.1 ilustra o circuito impresso desta interface.

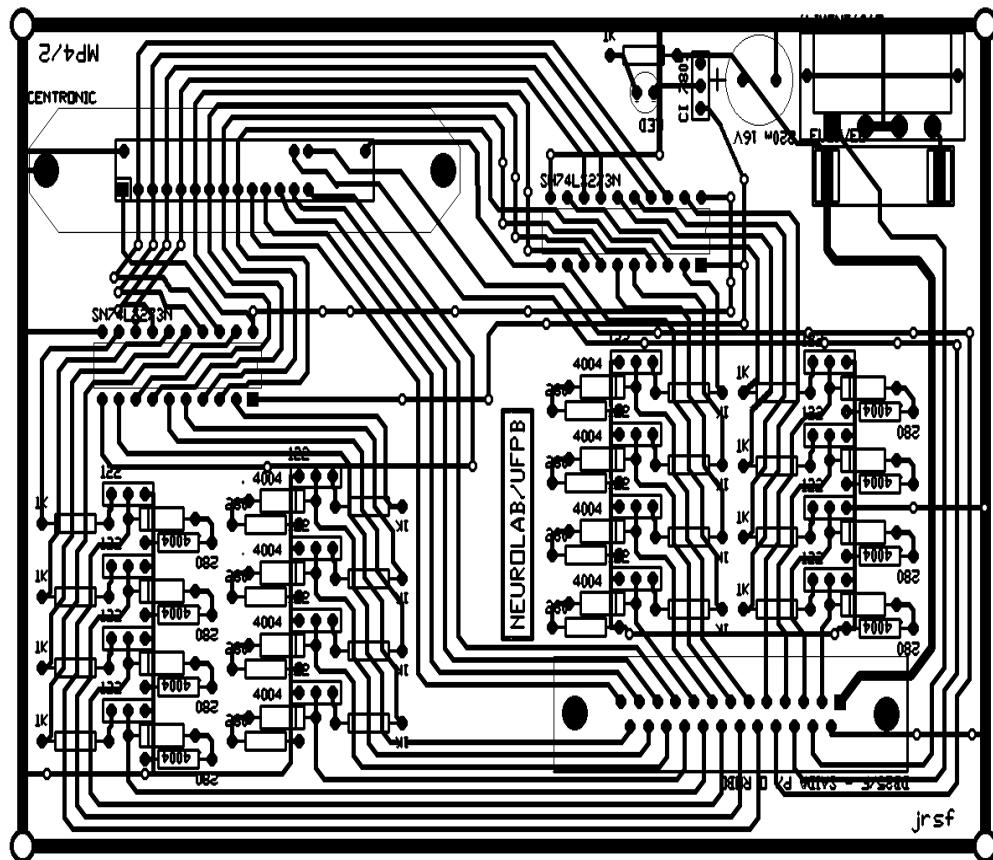


Figura B.1 - Circuito impresso da interface de motor de passo

### B.3 Interface painel para motores de passo

A interface painel utilizada juntamente com a interface controladora de motores de passo é um circuito de distribuição de sinal muito simples. A sua única finalidade é permitir mais liberdade na instalação do robô dentro do NEUROLAB. Os únicos componentes desta interface são os 4 plugs de 5 pinos para a conexão dos motores e 2 plugs de 4 pinos para a

conexão dos detectores de posição, não havendo necessidade de ser apresentada a lista de componentes e nem o diagrama esquemático. É apresentado somente o circuito impresso, ilustrado na Figura B.2.

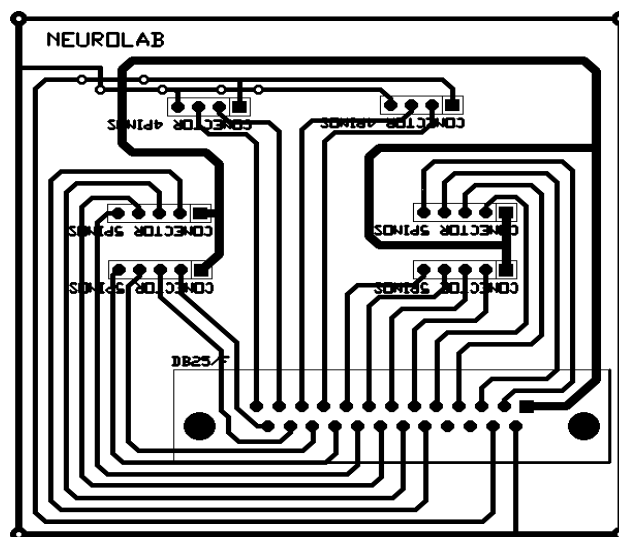


Figura B.2 - Circuito impresso do painel p/ motor de passo