

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA
TESE DEFENDIDA POR *Paulo Henrique*
de Araújo Madeira..... E APROVADA
PELA COMISSÃO JULGADORA EM *23.02.2008*


.....
ORIENTADOR

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**

Aplicação do Estudo da Interface Homem Máquina em Cadeiras de Rodas Motorizadas

Autor : **Paulo Henrique de Araújo Madeira**
Orientador: **Prof. Dr. Giuseppe Franco Dedini**

23/2008

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO**

Aplicação do Estudo da Interface Homem Máquina em Cadeiras de Rodas Motorizadas

**Autor : Paulo Henrique de Araújo Madeira
Orientador: Prof. Dr. Franco Giuseppe Dedini**

Curso: Engenharia Mecânica.
Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

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

Campinas, 2008
S.P. – Brasil

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

M264a Madeira, Paulo Henrique de Araújo
 Aplicação do estudo da interface homem máquina em cadeiras de
 rodas motorizadas / Paulo Henrique de Araújo Madeira. --
 Campinas, SP: [s.n.], 2008.

 Orientador: Franco Giuseppe Dedini
 Dissertação (mestrado) - Universidade Estadual de Campinas,
 Faculdade de Engenharia Mecânica.

 1. Deficiente físicos. 2. Cadeiras de rodas. 3. Interfaces de
 usuário (Sistema de computador). 4. Interação homem-máquina. 5.
 Construção modular. 6. Veículos autônomos. I. Dedini, Franco
 Giuseppe. II. Universidade Estadual de Campinas. Faculdade de
 Engenharia Mecânica. III. Título.

Título em Inglês: Application of human machine interface research in powered wheelchair

Palavras-chave em Inglês: Interface, Wheelchair, Integration, Modularization, Navigation support

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Mestre em Engenharia Mecânica

Banca examinadora: Pablo Siqueira Mereilles, Marcelo Becker

Data da defesa: 22/02/2008

Programa de Pós-Graduação: Engenharia Mecânica

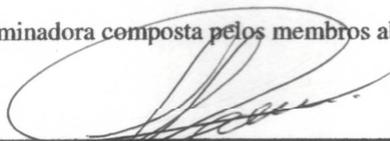
**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO**

DISSERTAÇÃO DE MESTRADO ACADEMICO

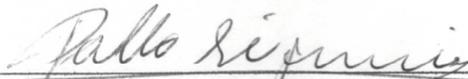
**Aplicação do Estudo da Interface Homem
Máquina em Cadeiras de Rodas Motorizadas**

**Autor : Paulo Henrique de Araújo Madeira
Orientador: Prof. Dr. Franco Giuseppe Dedini**

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:



**Prof. Dr. Franco Giuseppe Dedini, Presidente
FEM/UNICAMP**



**Prof. Dr. Pablo Siqueira Meirelles
FEM/UNICAMP**



**Prof. Dr. Marcelo Becker
PUC/Belo Horizonte**

Campinas, 22 de fevereiro de 2008

Dedicatória:

Dedico este trabalho a minha família, por sempre me apoiar; ao Dr. Torsten Felzer, pelo interesse a apoio; e a comunidade portadora de limitações físicas, a quem espero contribuir com este projeto.

Agradecimentos:

Este trabalho é resultado do apoio de diversas pessoas, a quem devo minha homenagem:

A minha família e amigos Manuel e Giancarlo, por estarem sempre ao meu lado;

Ao Prof. Dr. Franco Giuseppe Dedini pela orientação, incentivo e também por acreditar neste projeto;

Ao Dr. Torsten Felzer pelo apoio e interesse em meu trabalho;

Ao meu pai, que me ajudou a projetar o amplificador para o piezoelétrico;

A toda equipe do Departamento de Projeto Mecânico da Unicamp, Vitor Akira Hanzama, Flávia Bonilha Alvarenga e Geraldo Gonçalves Delgado Neto pela construção do kit de motorização; e inclusive ao Breno Raizer, pela paciência ao me ajudar na revisão da estrutura deste trabalho;

A toda a equipe do Departamento de Inspeção Robótica do CenPRA, inclusive ao Dr. Roberto Tavares por me ensinar a programação em microcontroladores PIC, e ao Sebastião Simões de Lima pelo apoio;

E a CAPES pela ajuda financeira para a realização deste trabalho.

Hey you, don't tell me there's no hope at all

Together we stand, divided we fall

Pink Floyd – Hey You

(1979)

Música composta a respeito do muro de Berlim

Resumo

MADEIRA, Paulo Henrique de Araújo, *Aplicação do Estudo da Interface Homem Máquina em Cadeiras de Rodas Motorizadas*, Campinas,: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2008. 115 p. Dissertação (Mestrado)

Há uma parcela da população que, por possuir severas limitações físicas e motoras, é inapta a utilizar cadeiras de rodas convencionais, manuais ou elétricas, bem como dispositivos de interface que não consideram as suas características particulares. Deste modo, este trabalho possui foco no desenvolvimento de interfaces e sistemas de navegação para cadeiras de rodas, provendo soluções para a inclusão deste grupo de usuários. Para isto, a revisão bibliográfica envolve três aspectos: as interfaces para usuários, os sistemas de navegação e a integração entre ambos. Primeiramente localiza-se a interface como parte da relação entre usuário e tarefa, para posteriormente restringir-se a interfaces para portadores de severas limitações físicas. Reconhecendo que determinados tipos de limitações de usuários demandam sistemas de navegação, estes são descritos na forma de exemplos de implementações. A revisão bibliográfica conclui com a descrição de métodos para a integração entre interface, navegação e controle da cadeira de rodas. Com base nesta pesquisa bibliográfica, desenvolveu-se um protocolo de integração e três tipos interfaces: controle de voz, filtragem do sinal de joystick para usuários com dificuldade em posicionar a mão, e controle por contrações musculares. A interface de reconhecimento de voz permitiu obter dois protótipos: um primeiro que utiliza um software comercial e um segundo em microcontrolador. A interface de filtragem do sinal do joystick foi prototipada virtualmente, demonstrando a resposta deste filtro a diferentes distúrbios. A interface que utiliza controle por contrações musculares foi implementada em protótipo funcional, por meio de um microcontrolador, e modularizada por meio do protocolo de integração desenvolvido. Este protocolo de integração foi testado em um kit de motorização com os módulos joystick, interface por contrações musculares e sonar, permitindo alternar, remover e inserir módulos sem a necessidade de reprogramação.

Palavras chave

- Interface, Cadeiras de Rodas, Integração, Modularidade, Auxílio a Navegação

Abstract

MADEIRA, Paulo Henrique de Araújo, *Aplicação do Estudo da Interface Homem Máquina em Cadeiras de Rodas Motorizadas*, Campinas,: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2008. 115 p. Dissertação (Mestrado)

There is a part of population, with physical impairment, that cannot use conventional manual or powered wheelchair and kinds of interface with do not consider they own characteristics. To solve this problem, this work does a bibliographic review on wheelchair interfaces and navigation support, alloying development on solutions for inclusion of this users group. The bibliographic review embraces three aspects: interfaces and users, navigation support systems, and integration between navigation and interfaces. The first chapter, interfaces and users, localizes the interface as an medium between user and task, allowing to be more focused on next chapter: interfaces for wheelchair uses with severe impairment level. Recognizing about in some kind's of user impairment levels an adequate interface is not enough, navigation support systems are described. The bibliographic review concludes with a description about integration methods for interfaces, navigation support and wheelchair control. Based on this bibliographic review, an integration protocol and three interfaces have developed: voice control, joystick filtering for users with difficult on hand positioning, and muscular contraction control. The voice recognition interface allows creating two prototypes: one with commercial software, and a second using embedded microcontroller on wheelchair. The joystick filtering interface was virtually prototyped allowing response tests for different noisy sources. The muscular contraction interface has implemented as functional prototyping, by a microcontroller, and modularized by the developed modularization protocol. This protocol has tested in a wheelchair motorization kit with joystick, muscular contraction interface and auxiliary sonar modules, allowing inserting or removing then without any reprogramming needs.

Keywords

Interface, Wheelchair, Integration, Modularization, Navigation Support

Índice

Lista de Figuras	xiii
Lista de Tabelas.....	xviii
1. Introdução.....	1
1.1. Formulação da Hipótese e Objetivos da Pesquisa	2
1.2. Histórico de Projetos.....	3
1.3. Organização do Trabalho.....	6
2. Desenvolvimento de Interfaces Homem Máquina.....	8
2.1. Características de uma Interface Adequada ao Usuário.....	10
2.1.1. Interfaces para Crianças	10
2.2. Procedimentos de Desenvolvimento de Interfaces	14
2.3. Avaliação de Tempos de Resposta para Desenvolvimento de Interfaces.....	16
2.4. Dispositivos para Interface com o Usuário.....	21
2.4.1. Dispositivos de Entrada de Dados.....	21
2.4.2. Dispositivos de Saída de Dados	24
3. Interfaces para Cadeiras de Rodas Motorizadas	26
3.1. Utilização de Sensores de Movimento.....	31
3.2. HaWCoS e a Utilização de Sensores Piezoelétricos.....	34
3.3. Interface Remota por Computador Embarcado	42
3.4. Utilização da Informação Facial	44

3.5.	Elétrodos para Leitura de Posição dos Olhos	46
3.6.	Electroencefalografia da Componente Alfa.....	50
3.7.	Implantes Neurais e Comunicação Bidirecional.....	53
3.8.	Aplicações com Implantes Cerebrais.....	56
3.9.	Filtragem e Condicionamento de Sinal.....	58
3.9.1.	Filtrar Eventuais Distorções provenientes do Meio Externo	58
3.9.2.	Condicionar uma Característica do Sensor ou do Sinal Desejado	58
3.9.3.	Ajustar uma Interface às Características do Usuário	59
4.	Sistemas de Navegação e Sensoreamento.....	61
4.1.	Origens da Navegação para Cadeiras de Rodas.....	64
4.1.1.	Roteamento do Caminho.....	65
4.1.2.	Sensoreamento	67
4.1.3.	Controle de Movimento	68
4.1.4.	Orientação	70
4.1.5.	Aceitação e Trabalhos Futuros	71
4.2.	Os projetos TAO – 1 e TAO – 2	71
4.3.	NavChair e a Combinação de Primitivas.....	75
4.3.1.	Algoritmos de Assistência à Navegação do NavChair.....	76
4.3.2.	Modos de Operação.....	79
4.3.3.	Seleção de Modo	80
4.4.	Abordagens VOA e BOA para o Desvio de Obstáculos.....	81
4.4.1.	Abordagem BOA.....	83
4.5.	MAid e a Navegação em Locais Muito Populosos	85
4.5.1.	Detecção de Objetos e Planejamento de Rotas de Desvio	87

5. Integração entre os Sistemas Envolvidos	88
5.1. Protocolo de Comunicação M3S	89
5.2. Utilização de uma Hierarquia de Entradas.....	92
5.3. Abordagem <i>Subsumption</i>	94
5.3.1. O robô RAM-1	100
5.3.2. O Aprendizado de Comportamentos com o Robô Obelix.....	101
5.3.3. Aprendizado por meio de <i>Subsumption</i>	103
6. Construção e Implementação de Protótipos	106
6.1. Reconhecimento de Voz	107
6.2. Filtragem de Joystick por Fuzzy	117
6.3. Implementação do HaWCoS em Microcontroladores PIC	122
6.3.1. Condicionamento de sinal	124
6.3.2. Processamento e Controle	126
6.3.3. Interface de Saída e Controle de Movimentos	127
6.4. Integração entre Interfaces e Sistemas Auxiliares	130
7. Resultados e Discussões.....	137
7.1. Reconhecimento de Voz	137
7.2. Filtragem de <i>Joystick</i> por Fuzzy	139
7.3. Implementação do HaWCoS em Microcontroladores PIC	146
7.4. Integração entre Interfaces e Sistemas Auxiliares	146
8. Conclusões e Planos Futuros.....	150
9. Bibliografia.....	152
Apêndice I Histórico de Cadeiras de Rodas	158
Apêndice II Tabelas com o levantamento das curvas de frequência utilizadas	163

Apêndice III Software Labview	168
Apêndice IV Programação do PIC 16F877A	169
Apêndice V Circuitos Envolvidos na Interface Modular	174
Apêndice VI Algoritmo do Firmware Utilizado na Interface HaWCoS	176

Lista de Figuras

Figura 1.1 - Primeira e segunda versão do kit de motorização	5
Figura 2.1 - Relações entre usuário e tarefa.	9
Figura 2.2 - Exemplo de consistência entre os navegadores Mozilla Firefox 1.0.7, Internet Explorer 6 e Ópera 7.	12
Figura 2.3 - Menu de comandos para unidades removíveis.	14
Figura 2.4 - Comparação da movimentação de um mouse entre usuário com paralisia cerebral e usuário comum.	20
Figura 2.5 - Teclado tipo Chord de cinco teclas.	22
Figura 2.6 - <i>Eyegaze</i> , capaz de ler o posicionamento do olho.	23
Figura 2.7 - Teclado virtual Microsoft®.	24
Figura 2.8 - Display braile.....	25
Figura 3.1 - Interface entre usuário e cadeira de rodas.	29
Figura 3.2 - Sensor piezoelétrico e amplificador.	36
Figura 3.3 - Estágios do sistema HaWCoS.	36
Figura 3.4 - Reconhecimento de evento de clique simples e duplo clique	37
Figura 3.5 - Controle do avanço por cliques.	39
Figura 3.6 - Percursos para teste.	40

Figura 3.7 - Trajetória da cadeira e sistema de controle.....	41
Figura 3.8 - Sinal sem filtragem e com filtragem,	45
Figura 3.9 - Exemplo de utilização do <i>EagleEyes</i>	46
Figura 3.10 - Comparação entre disposição dos sensores pelo método EagleEyes e Law <i>et al.</i> (2002).	49
Figura 3.11 - Diagrama do sistema de controle pela componente alfa.	51
Figura 3.12: Amostragem de componente alfa para dois usuário diferentes.	52
Figura 3.13: Implante Bidirecional e Eletrodos.	54
Figura 4.1 - Exemplo de um mapa de escritório	66
Figura 4.2 - Erros de leitura de sonares.....	68
Figura 4.3 - Utilização de rodapé preto como referencial.....	69
Figura 4.4 - Reconhecimento das paredes de um corredor por sonar.	70
Figura 4.5 - Arquitetura comportamental do projeto TAO-1.....	72
Figura 4.6 - Método MVFH.	77
Figura 4.7 - Método Vector Force Field.	78
Figura 4.8 - Utilização do modo passagem por porta.	79
Figura 4.9 - Obtenção do cone de colisões para um obstáculo fixo.....	81
Figura 4.10 – Determinação de rotas de desvio pelo método VOA.....	83
Figura 4.11 – Utilização da abordagem BOA para o desvio de obstáculos.....	84
Figura 5.1 - Arquitetura M3S para protocolo de integração.	91
Figura 5.2 - Representação de evento em módulo para hierarquia de sonar.....	93
Figura 5.3 - Abordagem tradicional para sistemas de navegação.	94
Figura 5.4 - Abordagem subsumption para sistemas de navegação.....	96
Figura 5.5 - Camada 0, destinada ao controle.	97

Figura 5.6 - Integração entre Camadas 0 e 1.....	98
Figura 5.7 - Integração entre camadas 0, 1 e 2.....	99
Figura 5.8 - Robô RAM1.	100
Figura 5.9 - Módulos do robô Obelix.....	102
Figura 5.10 - Robô Genghis, MIT.....	103
Figura 5.11 - Sensores de feedback para o robô Genghis.	104
Figura 6.1 - Arquitetura do primeiro protótipo de reconhecimento de voz.	108
Figura 6.2 - Arquitetura do segundo protótipo de reconhecimento de voz.....	110
Figura 6.3 – Comparação entre fonemas ‘a’, ‘e’, ‘i’, ‘o’, ‘u’	112
Figura 6.4 – Aquisição das bordas de subida, para os fonemas ‘a’ e ‘i’.....	112
Figura 6.5 - Curvas de frequência para diferentes palavras.	114
Figura 6.6 - Comparando uma amostragem com duas curvas de frequências da biblioteca.....	115
Figura 6.7 – Aplicação direta de lógica fuzzy para um eixo do joystick.	118
Figura 6.8 - Software para obtenção da posição desejada pelo usuário.	121
Figura 6.9 - Arquiteturas do Sistema HaWCoS.	122
Figura 6.10 - Arquitetura utilizada para interface	124
Figura 6.11 - Piezoelétrico.	125
Figura 6.12 - Interface de saída.	127
Figura 6.13 - Navegação na interface por cliques.	128
Figura 6.14 - Interface simplificada modelo 1.	129
Figura 6.15 - Interface simplificada modelo 2.	130
Figura 6.16 - Configuração de pinos no protocolo utilizado.....	133
Figura 6.17 - Regulador de tensão.	134
Figura 6.18 - Exemplo de integração hierárquica entre módulos.	135

Figura 6.19 - Arquitetura modular utilizada.....	136
Figura 7.1 - Fontes de ruído senoidal e randômico adicionada a uma entrada.	139
Figura 7.2 - Acomodação de 4 segundos para 40 amostras.	140
Figura 7.3 - Supressão do ruído senoidal por filtragem.	141
Figura 7.4 - Resposta em degrau para ruído senoidal de 1Hz.	142
Figura 7.5 - Filtragem passa baixa de um ruído randômico.....	143
Figura 7.6 - Filtragem fuzzy.....	143
Figura 7.7 - Curvas relacionadas a filtragem.	144
Figura 7.8 - Testes com dois <i>joysticks</i> reais.	145
Figura 7.9 - Módulos utilizados no kit de motorização.....	147
Figura 7.10 - Conectividade entre módulos.	148
Figura I.1 - Maca grega e cadeira com rodas chinesa.	159
Figura I.2 - Cadeira de rodas utilizada pelo rei Philip II.....	159
Figura I.3 - Cadeira de Stephan Farfler.....	160
Figura I.4 - Cadeira de John Dawson.....	160
Figura I.5 - Cadeira modelo <i>seating wheelchair</i>	161
Figura I.6 - Cadeira com foco em redução de peso.....	161
Figura I.7 - Cadeira de Everest & Jennings.	162
Figura II.1 – Comparação entre curvas médias obtidas	167
Figura III.1 – Software em Labview	168
Figura IV.1 – Pinagens de microcontroladores da família 16F.....	170
Figura IV.2 – Placa de testes para programas	171
Figura IV.3 – Fonte de alimentação	172
Figura. IV.4 – Progranador	173

Figura V.1 – Amplificador Piezoelétrico	174
Figura V.2 – Ajuste de Níveis de Tensão para o driver (MAANT).....	174
Figura V.3 – Inibidor para adicionar módulo de Sonar.....	175
Figura V.4 – Aplicação do microcontrolador 16F877A na Interface HaWCoS	175
Figura VI.1 – Subrotinas do menu 1	176
Figura VI.2 – Rorina principal, responsável por organizar menus	177
Figura VI.3 – Subrotinas do menu 2	178
Figura VI.4 – Subrotinas do menu 3	179
Figura VI.5 – Subrotinas para obter interações do usuário por interrupção.....	180
Figura VI.6 – Atribuição de movimentos.....	181

Lista de Tabelas

Tabela 3.1 - Interface por movimentos dos ombros.....	32
Tabela 3.2 - Comparação de tempos (segundos) entre percursos.	41
Tabela 3.3 - Movimentação pelo método de Law et al. (2002).....	50
Tabela 4.1 - Métodos de controle em função do grau de autonomia.	62
Tabela 5.1 - Exemplo de hierarquia para interface.	92
Tabela 6.1 - Comparação entre recursos de microcontrolador e computador.....	116
Tabela 6.2 - Lógica de controle para o driver.	131
Tabela 6.3 - Relação entre comandos de direção e movimentos	134
Tabela 7.1 - Estimativa de custos da segunda arquitetura.....	138
Tabela II.1 – Amostragem da palavra Frente.....	163
Tabela II.2 – Amostragem da palavra Testando.....	164
Tabela II.3 – Amostragem da palavra Esquerda	164
Tabela II.4 – Amostragem da palavra Direita.....	165
Tabela II.5 – Amostragem da palavra Trás.....	165
Tabela II.6 – Amostragem da palavra Liga.....	166
Tabela II.7 – Amostragem da palavra Desliga.....	166

Capítulo 1

Introdução

Graças aos progressos nas características construtivas de cadeiras de rodas, como as reduções no custo e peso, ao longo dos anos foi possível permitir que um número cada vez maior de portadores de limitações físicas tenham acesso a este veículo. Este progresso inicialmente ocorreu na melhoria dos materiais (da madeira para os metais de ligas leves) e em características construtivas, que juntos garantiram maior conforto e menor fadiga ao usuário. Com os avanços atuais da engenharia mecatrônica, observa-se que há um novo segmento de pesquisa em cadeiras de rodas: interfaces e sistemas de auxílio à navegação.

Assim, do mesmo modo que os avanços tecnológicos iniciais permitiram incluir um maior grupo de usuários por meio de uma redução de peso e ajustes ergonômicos nas cadeiras, os avanços atuais permitem incluir um grupo ainda maior de usuários, como resultado de uma significativa redução no esforço ao utilizar a cadeira de rodas. Esta redução de esforços é consequência direta das inovações tecnológicas posteriores à motorização de cadeiras de rodas, entre as quais é possível citar a detecção de contrações musculares para interfaces, e o compartilhamento de níveis de autonomia entre usuário e o sistema de navegação da cadeira de rodas.

Deste modo, a realização de um levantamento bibliográfico e o desenvolvimento de novas interfaces e sistemas de auxílio a navegação permitem encontrar soluções que se adaptem a diversas características de usuários. A escolha ou o desenvolvimento de uma interface adequada permitirá, em última instância, um significativo ganho na mobilidade e qualidade de vida de portadores de limitações físicas: é possível observar que alguns usuários consideram que a cadeira de rodas é uma extensão de seu corpo.

1.1. Formulação da Hipótese e Objetivos da Pesquisa

Reconhecendo que portadores de limitações severas são incapazes de utilizar cadeiras de rodas motorizadas convencionais, que utilizam interface por joystick, formulou-se a seguinte hipótese: “A utilização de um sistema de interface e auxílio à navegação adequado permite que usuários com severas limitações físicas controlem uma cadeira de rodas.” Esta hipótese permitiu definir o objetivo principal para este trabalho: “pesquisar e desenvolver interfaces e sistemas de auxílio a navegação, que possam ser acessíveis a usuários com limitações físicas graves”.

Iniciando-se com a pesquisa de interfaces foi possível obter uma relação entre usuário e tarefa, a qual permite definir as características necessárias para uma interface, como um produto desta relação. Ou seja, a interface é um módulo que permite relacionar o usuário, e suas formas de interação, com a tarefa e os comandos necessários para a sua execução. Esta abordagem permitiu restringir-se às características de interfaces para usuários de cadeiras de rodas com limitações motoras graves e, posteriormente, a sistemas de auxílio à navegação.

Após a pesquisa em soluções de interface e sistemas de auxílio a navegação, observou-se a necessidade de integrar estes sistemas em uma arquitetura modular, que permita o intercâmbio entre soluções de diferentes autores. Para isto, delimitou-se um objetivo secundário que não é dirigido a desenvolver uma interface específica, mas um protocolo para permitir utilizar diferentes formas de interfaces e sistemas de auxílio a navegação.

Por meio de uma revisão bibliográfica, foi possível reunir três áreas de conhecimento: interfaces para usuários com limitações físicas, sistemas de navegação e métodos para a integração entre ambos com a cadeira de rodas. A base teórica nestes três campos permitiu determinar um terceiro e último objetivo: desenvolver e, se possível, implementar sistemas de interface para a sua utilização em um kit de motorização de baixo custo. Observa-se que este kit foi desenvolvido pelo Departamento de Projeto Mecânico, que possui um protótipo funcional.

1.2. Histórico de Projetos

Os projetos desenvolvidos e descritos neste trabalho são parte das pesquisas sobre motorização de cadeiras de rodas realizadas no laboratório de projeto mecânico da Unicamp, sob orientação do Prof. Dr. Franco Giuseppe Dedini. Desta forma é possível apresentar um breve histórico à respeito destes trabalhos realizados, permitindo localizar a contribuição deste trabalho.

A dissertação de mestrado de Becker (1997) avaliou as características construtivas relacionadas à locomoção de robôs com o objetivo da construção de veículos destinados ao transporte de usuários com limitações físicas. Desta forma, realizou-se um modelamento das rodas avaliando deslocamentos com e sem escorregamento e as características de rodas elásticas ou rígidas. Este modelamento das rodas permitiu avaliar diferentes formas construtivas de robôs, combinando o número e a disposição das rodas de tração e direção, para avaliar a estabilidade ao tombamento e dirigibilidade de cada construção. A dissertação conclui com uma discussão a respeito de formas de controle nebuloso, e o desenvolvimento e simulação de uma técnica de controle para o desvio de obstáculos.

Em seu doutorado, Becker (2000) estuda e propõe técnicas assistivas para cadeiras de rodas inteligentes. Para isto inicia-se com uma descrição à respeito dos portadores de deficiência usuários de cadeiras de rodas no Brasil: quem são onde vivem e tipos de deficiência apresentados. Após esta descrição ampla, restringe-se ao aspectos ergonômicos da relação entre usuário e cadeiras de rodas: dimensões do usuário, tipos de estofamentos adequados, exemplos de interfaces tipo joystick e demais fatores associados. A delimitação das características ergonômicas permite estudar as formas de acionamento manual de cadeiras de rodas em função do tipo de lesão e posicionamento do usuário na cadeiras de rodas

Após concluir uma análise das características do cadeirante, Becker (2000) apresenta o modelamento e a simulação estática e dinâmica da estabilidade lateral e longitudinal para cadeira de rodas, relacionando os fatores posição do centro de gravidade, cambagem e acelerações decorrentes do movimento. Por meio destas características estáticas e dinâmicas obtidas, e levantamento bibliográfico, desenvolvem-se técnicas de navegação e sensoamento por lógica fuzzy para cadeiras de rodas. Estas técnicas de navegação e sensoamento permitem reconhecer e mapear paredes, e demais obstáculos, para desviar-se de rodas de colisão e alcançar um local desejado.

A dissertação de mestrado de Lombardi Júnior (2002) demonstra um mecanismo servo-assistido para reduzir os esforços realizados por cadeirantes, permitindo transpor obstáculos e evitar lesões por esforços repetitivos. Esta dissertação inicialmente define e caracteriza os portadores de deficiência no Brasil e relaciona a cadeira de rodas com o portador de limitações. Este relação envolve aspectos como as características das cadeiras e a influência da cadeira de rodas nas relações interpessoais. Após descrever o usuário, e suas relações com as cadeiras de rodas, descrevem-se os aspectos estáticos e dinâmicos das cadeiras de rodas, como: modelamento da roda, estabilidade, tombamento e dirigibilidade.

Desta forma, Lombardi Júnior (2002) realiza um estudo das características biomecânicas, ou seja, como é a relação de trabalho entre homem e cadeira de rodas avaliando os esforços musculares. A análise biomecânica considerou os esforços, e diferentes formas de acionamento de cadeiras de rodas manuais, para propor um sistema servo-assistido que proporcione uma redução de esforços no controle de uma cadeira de rodas.

No mesmo ano, a dissertação de mestrado de Alvarenga (2002) reconhece que aproximadamente 75% das cadeiras de rodas utilizadas no Brasil são manuais, e analisa as possibilidades para a motorização destas cadeiras com uma solução modular e de custo acessível. Para isto, inicialmente é realizada um estudo sobre portadores de limitações físicas sobre os aspectos de: tipos de deficiência, trabalho, lazer e acessibilidade. De forma semelhante também estudaram-se tipos e características para cadeiras de rodas e sistemas de motorização já existentes. Por meio de uma metodologia de projeto, que inclusive considerou as características ergonômicas dos cadeirantes e modelamentos dinâmicos das cadeiras de rodas, desenvolveu-se

um módulo tipo reboque para motorização de cadeiras de rodas. A dissertação concluiu com as simulações dinâmicas do produto desenvolvido.

Continuando a linha de pesquisas, a tese de doutorado de Alvarenga (2006) propõe uma abordagem para o projeto de produtos inclusivos por meio do *Universal Design*. Desta forma inicia-se com uma revisão bibliográfica a respeito das características do projeto inclusivo para posterior análise e discussão de metodologias de projeto. Por meio desta revisão Alvarenga (2006) desenvolve uma metodologia de projeto e a aplica no desenvolvimento de um produto inclusivo: um kit de motorização para cadeiras de rodas. No decorrer da dissertação de mestrado foram desenvolvidos e construídos duas versões de kits de motorização para cadeiras de rodas, ilustrados na figura 1.1.



Fig. 1.1 - Primeira e segunda versão do kit de motorização, respectivamente esquerda e direita, (fonte: Alvarenga, 2006).

Assim, neste contexto, o presente trabalho realizará uma revisão bibliográfica sobre soluções em interface para usuários tetraplégicos para o posterior desenvolvimento e implementação destas interfaces no kit de motorização desenvolvido na tese de Alvarenga (2006). É necessário ressaltar que, seguindo o conceito apresentado por Alvarenga (2006), é mantido o compromisso de desenvolver soluções de custo acessível a todos os usuários portadores de limitações físicas. Para isto, é apresentada uma abordagem na relação usuário e tarefa para encontrar as necessidades principais do usuário e, desta forma, encontrar quais são os mecanismos essenciais na execução de uma determinada tarefa.

1.3. Organização do Trabalho

Há uma parcela da população que, por possuir severas limitações físicas e motoras, é inapta a utilizar cadeiras de rodas convencionais, manuais ou elétricas, bem como dispositivos de interface que não consideram as suas características.

Assim, este trabalho descreverá a pesquisa e desenvolvimento de interfaces, sistemas de auxílio a navegação, e a integração entre ambos para o controle de cadeiras de rodas motorizadas, provendo a inclusão de usuários com limitações motoras graves. Estas soluções, já anteriormente citadas, envolvem três aspectos: as interfaces para usuários, os sistemas de navegação e a integração entre ambos. Para facilitar a delimitação do tema, o apêndice I apresenta um breve histórico sobre o desenvolvimento das cadeiras de rodas que permite observar a sua evolução até os dias atuais.

Prioritariamente ao desenvolvimento de interfaces para portadores de limitações físicas, é necessário reconhecer quais características serão necessárias em uma interface. Desta forma, o capítulo 2: Desenvolvimento de Interfaces Homem Máquina localiza a interface como meio para relação entre usuário e tarefa e descreve as características presentes em uma boas interfaces e procedimentos para o seu desenvolvimento. O conhecimento no desenvolvimento de interfaces permite restringir esta área de conhecimento a portadores de severas limitações físicas no capítulo 3: Interfaces para Cadeiras de Rodas Motorizadas

Contudo, ainda que estas interfaces permitam interagir com o usuário, para determinadas limitações são necessários sistemas de auxílio à navegação. Estes sistemas permitem uma navegação facilitada (semi-autônoma) ou mesmo autônoma, e serão apresentados no capítulo 4: Sistemas de Navegação e Sensoreamento. O levantamento bibliográfico realizado permitiu observar que, em grande parte dos casos, as interfaces são desenvolvidas e embarcadas na forma de soluções específicas para um determinado problema. Esta abordagem dificulta o compartilhamento entre interfaces e sistemas de navegação desenvolvidos por autores diferentes.

Para uma integração entre diferentes soluções, para diferentes características de usuários, no capítulo 5: Integração entre os Sistemas Envolvidos são apresentados protocolos e meios de integração. Com base na teoria apresentada, foi possível desenvolver três módulos de interface e um protocolo de integração. Estes sistemas são descritos no capítulo 6: Construção e Implementação de Protótipos.

Estas três interfaces e o protocolo de integração desenvolvidos foram testados no capítulo 7: Resultados e Discussões, que demonstra algumas dificuldades encontradas no seu desenvolvimento e os resultados obtidos nos experimentos realizados. Uma conclusão que engloba os quatro sistemas desenvolvidos é apresentada no capítulo 8: Conclusões e Planos Futuros.

Capítulo 2

Desenvolvimento de Interfaces Homem Máquina

A interface está presente em qualquer ferramenta, na forma de intermédio entre o usuário e a tarefa realizada. Contudo, antes de desenvolver uma determinada interface, é necessário um estudo prévio dos objetivos do usuário em função da tarefa para definir qual a ferramenta será a mais adequada.

Segundo Donald A. Norman, em Laurel (1994), uma análise adequada deve sugerir um conjunto de interface e ferramenta que melhor se adaptem a relação de usuário e tarefa, de forma a evitar a falta de dispositivos em locais necessários ou o seu emprego em locais desnecessários. Desta forma, este capítulo descreve os elementos envolvidos no desenvolvimento de interfaces, favorecendo a delimitação das características necessárias em função da relação usuário e tarefa.

Refletindo a respeito das características do usuário e da atividade a ser executada pode ser observado que, em função da experiência do usuário, a ferramenta ideal pode ser a solução que apresente um menor número de recursos. A medida em que o usuário possui maior aptidão a lidar com recursos mais sofisticados podem ser utilizadas interfaces mais complexas, e que possuem um número maior de recursos e funções. Esta aptidão envolve aspectos culturais, sociais, idade ou características como limitações físicas (determinadas limitações necessitam interfaces adaptadas).

Desta forma, é possível observar que o tipo de ferramenta e interface associada deve ser encarada como um produto da relação entre usuário e aplicação. Estas situações podem ser ilustradas na figura 2.1, que representa a diversidade nas relações entre usuário e tarefa, em função da sofisticação da ferramenta utilizada:

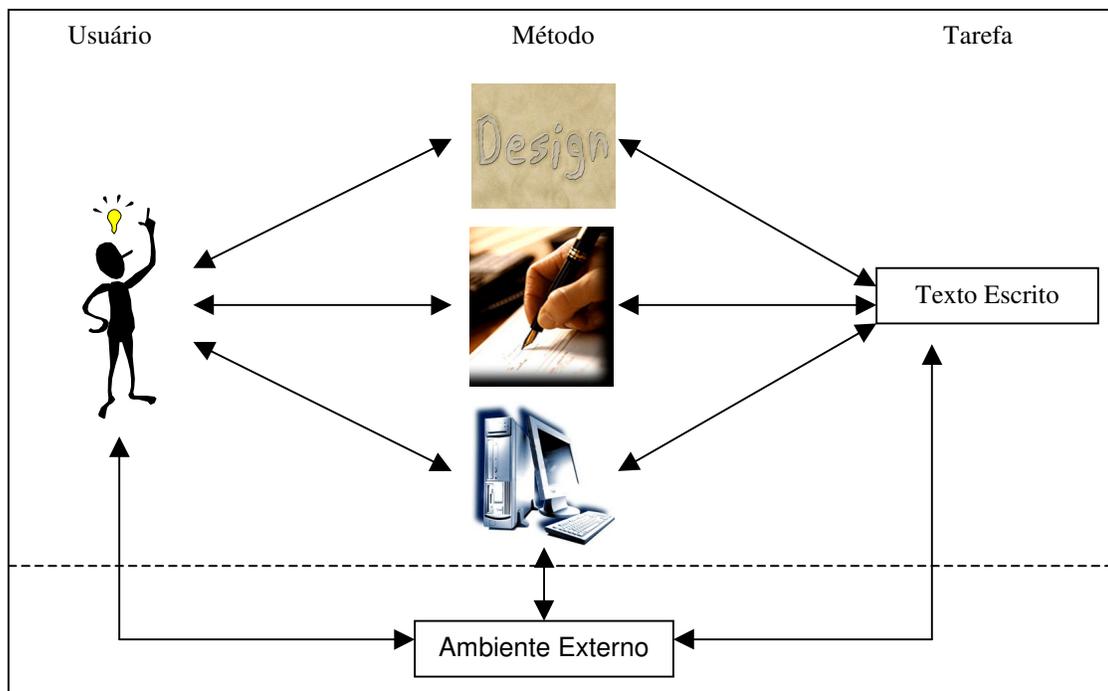


Fig. 2.1 - Relações entre usuário e tarefa.

É observada também a presença do ambiente externo neste diagrama, pois não é possível a existência de um sistema completamente isolado. Como exemplos, este ambiente pode permitir retorno por informação visual das interações do usuário; ou pode ocasionar distúrbios na ferramenta ou método utilizado (a poluição sonora pode inviabilizar o reconhecimento de voz).

Uma avaliação correta das características do usuário, e posterior escolha da interface adequada, permitirá que este se relacione do modo mais direto possível com a tarefa realizada. Esta proximidade favorece a cognição e posterior aprendizado, bem como a facilidade na execução da tarefa pois, a medida em que uma interface torna-se mais complexa, há uma menor transparência entre usuário e tarefa.

Os tópicos seguintes descrevem as características destes elementos que compõe o desenvolvimento de interfaces, abrangendo o usuário, características da interfaces adequadas, métodos para o seu desenvolvimento e exemplos de interfaces e suas aplicações.

2.1. Características de uma Interface Adequada ao Usuário

No processo de desenvolvimento de uma interface é necessário estudar as características comportamentais e as limitações do usuário, pois estas variam em função de aspectos culturais, sociais, idade e características próprias como alguma limitação física. Como exemplo são citadas características particulares a relação entre crianças e interfaces.

2.1.1. Interfaces para Crianças

A idade influencia de forma significativa na forma com a qual uma pessoa interage com o seu meio. Esta influência sugere que, para desenvolver uma interface adequada, esta interação deve ser facilitada de forma a considerar as características que cada idade apresenta. Segundo Joyce Hakansson, em Laurel (1994), que por possuir experiência no desenvolvimento de produtos associados ao aprendizado para crianças, cita algumas lições aprendidas em respeito ao comportamento de crianças:

- Há interesse por jogos eletrônicos, principalmente quando este apresenta níveis de dificuldade graduais, pois permite que elas possam se testar e comparar-se a outras.

- Crianças apresentam certa dificuldade com textos e grande facilidade com imagens, pois estas permitem associação direta à interface. É interessante observar que imagens também facilitam uma comunicação mais direta e menos cansativa, inclusive para adultos.

- Quando há menos computadores que crianças, é possível a realização de trabalhos em grupo por meio do aprendizado de novos recursos.

- Pode ser observado um maior interesse por computadores em garotos que em meninas após uma determinada idade, como resultado de motivos culturais e das características dos softwares desenvolvidos.

- Crianças podem aprender a utilizar uma grande variedade de interfaces, embora encontrem dificuldades com o teclado, pois precisam dividir as palavras e procurar cada letra, ainda que estas estejam em ordem alfabética. Desta forma interfaces como um joystick ou mouse são mais indicadas.

- O aprendizado é facilitado por recursos multimídia (com gráficos, sons e movimentos) se comparado a uma linguagem apenas textual.

- Desde que estimuladas, as crianças são capazes de aprender por associações únicas e inesperadas. Como exemplo, nos anos 60, crianças com idade entre 5 e 7 anos aprenderam as letras do alfabeto assistindo programas educativos como a Vila Sésamo (“Sesame Street”).

Segundo Donald A. Norman, em Laurel (1994), “Devemos nos perguntar a respeito de quais tarefas as pessoas precisam cumprir, e quais seriam as ferramentas mais apropriadas”, de forma a sugerir um design centrado na relação entre a tarefa a ser realizada e o usuário. Esta abordagem sugere que uma interface adequada é aquela que se comporta de forma transparente com a relação usuário e tarefa, sugerindo que esta deve ser simples e possuir aspectos semelhantes a outras interfaces já conhecidas pelo usuário (consistência).

Uma abordagem mais restrita é realizada por Preece *et al.* (2002), ao delimitar que o desenvolvimento de interfaces compreende duas áreas de objetivos: a relação entre usuário e interface, e a prática de uso.

A relação entre usuário e interface, definida como “*user experience goals*” por Preece *et al.* (2002), é composta por aspectos subjetivos. Desta forma uma interface adequada ao usuário deve ser: agradável, divertida, interessante, fácil de aprender, motivante, possuir uma estética

harmoniosa, facilitar a criatividade, recompensadora e agregar emoções. Sob o ponto de vista da prática de uso (*usability*) são definidas metas de como melhorar a interface para o usuário, associadas às características de: efetividade, eficiência, segurança, utilidade e facilidade para aprender e memorizar para uso posterior.

A combinação das metas de prática de uso com a relação entre usuário e interface, permitem obter as necessidades de grupo de usuários para o qual a interface se destina em foco. Esta delimitação do grupo de usuários e levantamento de suas necessidades permite reconhecer em quais pontos uma interface pode ser melhorada. Como exemplo do emprego das metas de prática de uso (*usability*), a programação de um vídeo cassete para gravação de um determinado programa é uma tarefa difícil para a maioria dos usuários. Isto se deve ao fato de que a maioria das interfaces não favorece aspectos como facilidade de aprendizado e memorização.

O aprendizado e facilidade de memorização podem ser favorecidos quando se utiliza recursos consistentes, ou seja, associar sempre um comando a uma determinada função, facilitando a sua memorização. Pode ser citado um exemplo na figura 2.2, que demonstra as ferramentas de navegação em internet para retornar e avançar páginas em três programas diferentes:

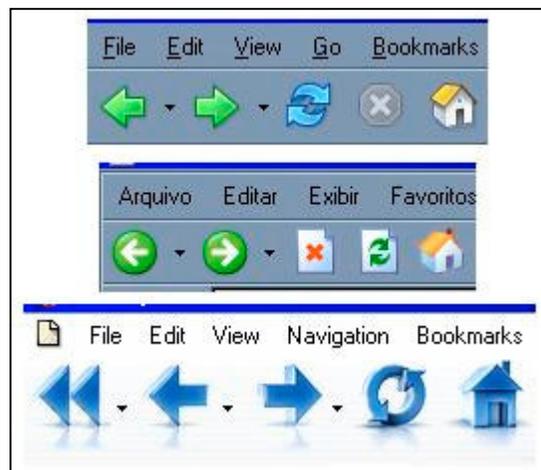


Fig. 2.2 - Exemplo de consistência entre os navegadores Mozilla Firefox 1.0.7 (superior), Internet Explorer 6 (centro) e Ópera 7 (inferior).

Segundo Bruce Tognazzini, em Laurel (1994), “Uma interpretação consistente do comportamento do usuário pelo sistema é mais importante que o comportamento consistente de sistemas ou objetos”, ou seja, o sistema deve realizar a operação desejada pelo usuário, aquela que ele está acostumado a realizar em sistemas semelhantes.

Pode ser citado como exemplo, os gravadores de fita: quando o usuário insere a fita e pressiona o comando play, este reproduz o lado da fita que é aparente para o usuário, independente do sentido de rotação do motor (horário ou anti-horário), pois a fita pode ser colocada por cima, por baixo ou horizontalmente em função do tipo de equipamento. Desta forma, conclui-se que o comando de play, para o usuário, equivale a reproduzir o lado que lhe é visível, e não o sentido de rotação da fita.

Uma interface com comandos simplificados e uma utilização de imagens e cores associadas aos comandos mais utilizados também facilitam o aprendizado e memorização. Esta facilidade se demonstra na redução do tempo gasto para a leitura da função de um determinado comando, e pode ser observada no exemplo dos navegadores de Internet já citados na figura 2.2, onde se utiliza setas para indicar funções comuns.

É observado consenso entre Bruce Tognazzini, em Laurel (1994), e Preece *et al.* (2002) que os aspectos associados à segurança podem ser obtidos de diversas formas, como organizar os comandos em grupos próprios, utilizar mensagens e meios de confirmar ações, prevenir ações que podem colocar em risco o usuário, fazer uso de funções consistentes e utilizar mecanismos redundantes para corrigir erros (como confirmar ações ou fazer cópias de segurança). Comandos organizados em grupos, em acordo com a sua função, permitem evitar erros ao pressionar a tecla indesejada. Exemplos de organização inadequada incluem colocar próximo funções de salvar e fechar, ou desligar próximo de controles direcionais.

Como exemplo, a figura 2.3 representa uma disposição de comandos para unidades de armazenamento removíveis. Nesta disposição a função ejetar, que é muito utilizada, encontra-se ao lado de uma função formatar, favorecendo um erro que resulta na perda de todos os dados.

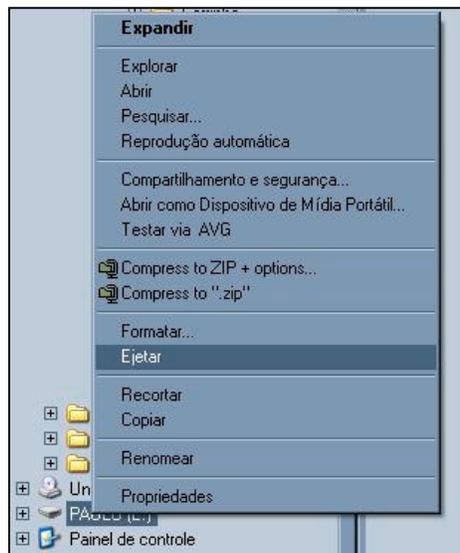


Fig. 2.3 - Menu de comandos para unidades removíveis (fonte: MS WindowsXP).

Em mecanismos que apresentam riscos a usuários inexperientes como crianças, segundo Preece *et al.* (2002), podem ser utilizados sistemas de proteção: a confirmação de ações pode ser realizada por meio de mensagens de confirmação tipo sim, não e cancelar ou sistemas mecânicos, como proteções sobre botões que associam-se a ações arriscadas. Como exemplo, a caixa de pilhas de brinquedos que só pode ser aberta por adultos, pois é necessária uma força maior que a utilizada por uma criança para brincar, outra restrição de acesso é a tampa sobre a caixa de disjuntores.

2.2. Procedimentos de Desenvolvimento de Interfaces

De forma análoga às ferramentas para o desenvolvimento de um determinado produto, também é possível utilizar métodos para o desenvolvimento de interfaces. Segundo Preece *et al.* (2002), é possível citar as principais fases que compõe o desenvolvimento de uma interface: identificação das necessidades requeridas, desenvolvimento de novos projetos, desenvolvimento de projetos interativos e fase de testes.

Para o desenvolvimento de uma interface centrada no usuário, é necessário compreender quais as suas expectativas em relação ao produto. Esta compreensão favorece o reconhecimento de aspectos da relação entre usuário e interface, permitindo atribuir maior ou menor peso a aspectos como: diversão, interesse, estética harmoniosa, favorecer aprendizado e criatividade. Observa-se que a atribuição de pesos a estes aspectos é resultado do tipo de interface, usuário e tarefa.

Após definir os interesses do usuário em um produto, são desenvolvidos os projetos com as novas características para incorporar estas necessidades. Este projeto pode ser subdividido em duas fases: conceitual e físico. Na fase conceitual o produto, é descrito como este que deve ser, se comportar e como pode ser a sua aparência. A fase seguinte é o projeto físico, que inclui uma quantidade maior de detalhes, como cores sons e imagens caso seja um projeto de interface gráfica.

Definidos os projetos conceitual e físico inicia-se o seu refinamento, antes de construir protótipos de alta fidelidade. Este refinamento é realizado por meio de protótipos de baixa fidelidade, pois agregam pouco custo e podem ser construídos e alterados em pouco tempo. Estes protótipos de baixa fidelidade permitem um certo grau de interação com a interface que é suficiente para que os projetistas e, caso necessário os usuários, avaliem os pontos em que o projeto pode melhorar.

Os protótipos de baixa fidelidade não necessitam apresentar funcionalidade, permitindo a sua construção em papel ou cartão, representando uma interface simples com botões e *displays*. Para a construção destes protótipos podem ser utilizados outros materiais, como caixas pintadas, desde que seja possível fazer o protótipo em pouco tempo a um baixo custo sem se preocupar com muitos detalhes. Deve ser ressaltado que o objetivo do protótipo de baixa fidelidade é transmitir os conceitos básicos da interface a ser desenvolvida.

Segundo Preece *et al.* (2002), após a avaliação e otimização dos protótipos de baixa fidelidade, inicia-se a construção de protótipos de alta fidelidade e decorrente fase de avaliação dos mesmos. Esta avaliação é realizada com o auxílio dos usuários que testam os protótipos, permitindo obter informações para aproximar o produto as suas necessidades. O protótipo

também pode ser dividido em dois tipos sob o ponto de vista do seu objetivo: protótipo horizontal (com várias funções embora poucos detalhes) ou vertical (poucas funções e muitos detalhes).

Durante o desenvolvimento destas quatro fases é necessário manter contato direto com o usuário, de forma a melhorar continuamente a interface. Este conceito é reafirmado por Donald A. Norman, no livro de Laurel (1994), que sugere um acompanhamento constante do início ao fim do projeto por usuários na forma de protótipos rápidos. Observa-se que Preece *et al.* (2002) também compartilha deste ponto de vista por meio do conceito projeto interativo e da necessidade em desenvolver protótipos, ainda que estes sejam de baixa fidelidade.

Os testes realizados com os usuários permitem obter novos pontos de vista, de modo a evitar a análise realizada apenas por projetistas e colegas de trabalho. Estes novos pontos de vista permitem novas correções no projeto de forma a garantir uma interface mais próxima ao grupo de usuários ao qual se destina.

2.3. Avaliação de Tempos de Resposta para Desenvolvimento de Interfaces

Segundo Keates *et al.* (1999) a utilização de computadores para portadores de limitações físicas trouxe muitos benefícios, como a leitura e utilização de processadores de texto, que permitiram inclusive a integração no mercado de trabalho deste grupo de pessoas. Contudo, segundo Karshmer (1997), as HCI (*Human Computer Interface*) desenvolvidas para portadores de limitações são adaptações de interfaces para usuários que não possuem limitações. Entre os principais motivos para a falta de interfaces adequadas, segundo Keates *et al.* (1999), pode ser citado que a maioria dos usuários não possui limitações, aumentando a dificuldade e custos para encontrar e realizar testes com grupos representativos de usuários.

Deste modo, são necessárias mudanças na filosofia de projeto para incluir usuários com limitações. No desenvolvimento de interfaces confortáveis e úteis para usuários com limitações (idosos ou deficientes), é necessário reconhecer as características do usuário. Para isto, é possível utilizar uma abordagem que permita obter um modelo do usuário em relação a sua percepção, cognição e reação no manuseio de uma determinada interface.

Segundo Blessing *et al.* (1995), o desenvolvimento de uma metodologia para a criação de interface ocorre em três estágios:

- *Estágio 1, Definir o problema:* aprender como usuários com limitações de movimentos interagem com computadores e as diferenças entre usuário com ou sem limitações. Neste estágio um modelo de interação entre usuário e máquina, como o desenvolvido por Card *et al.* (1983), pode ser utilizado.
- *Estágio 2: Sugerir uma solução:* desenvolver uma metodologia que considere o conhecimento do usuário
- *Estágio 3: Testar a solução:* garantir-se que a solução é efetiva

Ignorar o primeiro estágio resulta em produtos não usáveis (Buhler, 1997) ou mal sucedidos comercialmente (Mahoney, 1997). Segundo Keates *et al.* (1999) o modelo do usuário pode ser obtido pelos métodos os modelos MHP (Modelo do Processador Humano) e KLM (*Keystroke Level Model*), desenvolvidos por Card *et al.* (1983).

A origem destas modelagens para o comportamento humano provém da neuropsicologia, e utilizam modelagem empírica. Ao utilizar dispositivos como teclado e mouse, esta metodologia adaptou-se também para o campo de conhecimento da HCI (*Human Computer Interface*). Estas metodologias, propostas por Card *et al.* (1983), subdividem e permitem quantificar os tempos relacionados a execução de uma determinada tarefa, bem como prever os tempos gastos em tarefas semelhantes. É ressaltado que cada usuário possui suas características individuais e, deste modo, necessitam de uma calibração própria para seus tempos.

Quando não há dados disponíveis a respeito de uma determinada característica do usuário é comum assumir estimativas, como utilizar a resposta motora baseando-se em dados de outros usuários. No Modelo do Processador Humano (MHP) o processo de interação é subdividido em três tipos de tempos:

- *Percepção* (τ_p): Tempo para percepção do evento,
- *Cognição* (τ_c): Tempo para processar a situação e encontrar uma resposta adequada,
- *Motora* (τ_m): Tempo para realizar a resposta apropriada.

Cada tempo é multiplicado por um coeficiente representado por um número inteiro, representando os ciclos de cada tipo de tempo (x, y e z, respectivamente), como ilustrado na equação 2.1:

$$TempoTotal = x.\tau_p + y.\tau_c + z.\tau_m \quad \text{Eq. 2.1}$$

O coeficiente (x, y, z) é sempre um número inteiro, pois não se considera meia percepção, meia cognição ou meio movimento. Caso um evento ocorra mais de uma vez, este o coeficiente é dado em função do número de repetições da tarefa, como levantar e descer um membro resulta em duas ações motoras.

Neste método, assume-se que o tempo total da tarefa e os coeficientes x, y e z de cada tarefa são conhecidos. Estas informações permitem obter os tempos relativos a percepção, cognição e movimentação do usuário por meio de comparações entre os tempo utilizados para realizar tarefas que apresentam constantes x, y, z distintas, resultando em um sistema de equações.

Assim, uma tarefa de apertar um botão em função de um *flash* de luz permite definir todos os coeficientes como um, pois esta percepção luminosa é um estímulo simples que consome um único tempo de percepção, e a cognição e movimentação (abaixar o dedo) também consomem um único período de tempo. Caso se utilize um botão que necessita ser pressionado e solto, dois eventos motores são considerados: apertar e soltar o botão. Desta forma é possível obter diretamente este tempo ao realizar uma tarefa de pressionar e soltar o botão repetidamente.

Ainda que Harter (1967) observe que a percepção seja diferente em função da natureza de cada sentido humano, bem como de indivíduo para indivíduo, Card *et al.* (1983) restringem que no contexto de interface homem-máquina apenas estímulos visuais e audíveis são considerados. Esta condição permite que o tempo de percepção visual também seja obtido diretamente por meio de um teste de percepção a estímulos visuais. Neste teste utilizam-se pequenas interrupções de tempo em uma animação: a menor interrupção de tempo perceptível equivale ao tempo de percepção.

O tempo cognitivo não pode ser obtido diretamente, mas por meio de tipos particulares de estímulos: a medida em que se aumenta o nível de abstração da atividade, este tempo cognitivo também aumenta. Desta forma o menor nível de abstração é reconhecer uma cor e uma forma que ocupam um único ciclo de cognição. Caso seja adicionada uma escolha a ser realizada, como entre um triângulo vermelho e um círculo azul, totalizam-se dois ciclos cognitivos (um para reconhecimento e outro para a decisão).

Um nível de maior abstração (três ciclos cognitivos) envolve formas já conhecidas ou que possuem nome, como a seleção de letras, pois adiciona-se um ciclo cognitivo adicional. Desta forma, obtém-se primeiramente os tempos visual e motor. Utilizando-se estes tempos, realizam-se testes para a obtenção dos tempos cognitivos, variando-se os graus de abstração e mantendo-se estímulos visuais e motores com o mínimo possível de alterações.

Para casos mais complexos, se comparado aos utilizados em modelagem MHP, Card *et al.* (1983) descreve a metodologia *Keystroke Level Model* (KLM), direcionada a teclado. O método KLM não utiliza coeficientes para os tempos de cada tarefa, mas aumenta o nível de subdivisões da tarefa total, como demonstrado na equação 2.2.

$$TempoTotal = W_k + W_p + W_h + W_d + W_m + W_r \quad \text{Eq. 2.2}$$

W_k : Pressionar tecla

W_d : Arrastar

W_p : Apontar

W_m : Operações Mentais

W_h : Direcionar

W_r : Resposta respectiva

Ambos os modelos MHP e KLM segmentam as ações em partes quantificáveis que podem descrever as ações realizadas pelo usuário em uma determinada tarefa. Estes métodos apresentam como limitação não ser considerada uma interdependência entre estas partes subdivididas, o que impede que o tempo gasto em um segmento altere o tempo gasto em outro segmento desta mesma tarefa.

Para o estudo da interação de usuários com limitações com dispositivos apontadores, como o mouse, é necessário uma metodologia diferenciada, que permita avaliar o percurso realizado durante a movimentação do usuário. Deste modo, Graham e MacKenzie (1996) observam que a interação de usuários sem limitações com dispositivos apontadores, combinam um movimento inicial balístico (rápido) com um movimento lento em torno do ponto a ser selecionado.

Uma comparação realizada por Keates *et al.* (1999) entre usuários com e sem limitações motoras demonstra a existência de movimentos não controlados e conseqüente dificuldade de posicionamento como ilustrado na figura 2.4:

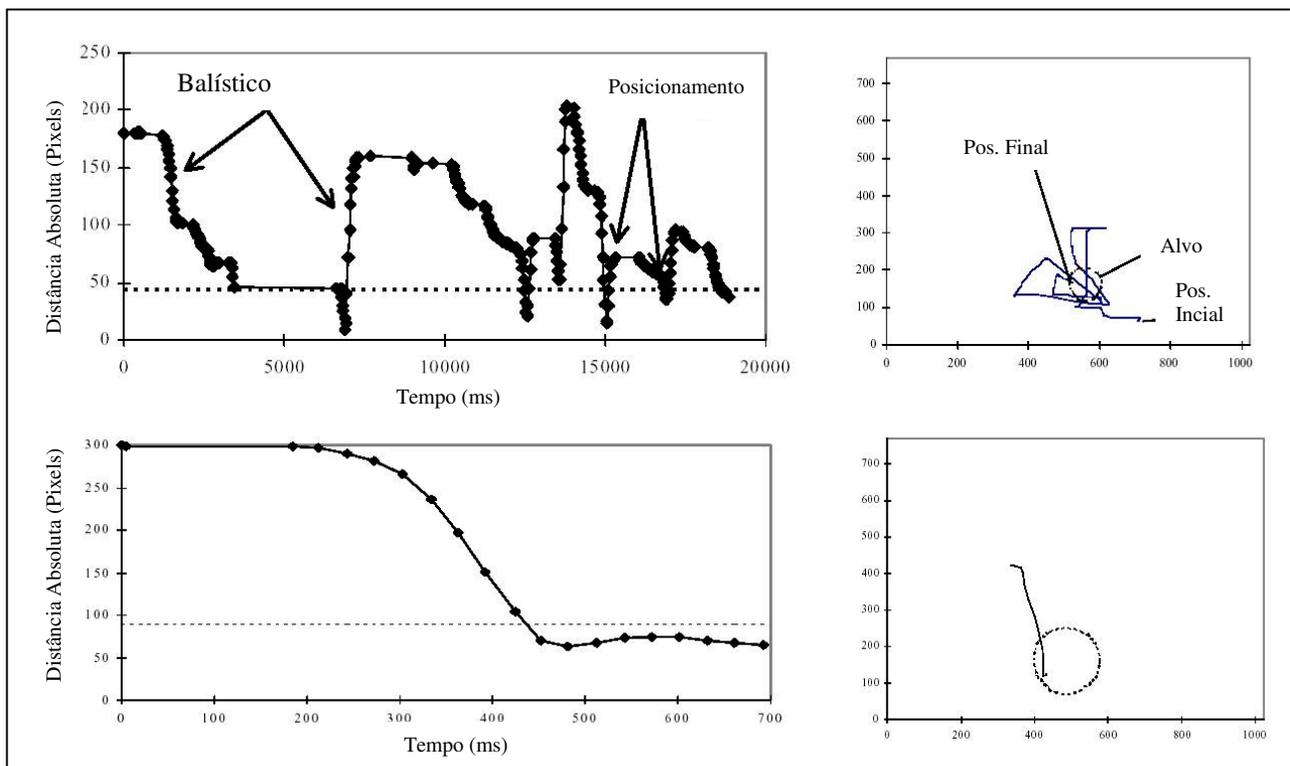


Fig. 2.4 - Comparação da movimentação de um mouse entre usuário com paralisia cerebral (acima), e usuário comum (abaixo) (fonte: Keates *et al.*, 1999).

Esta dificuldade pode ser compensada pelo uso de filtros passa baixa, curvas de velocidade e aceleração, interfaces auto-adaptativas e agentes inteligentes de interfaces. Outro aspecto a ser avaliado é a técnica de seleção, em que estudos iniciais demonstram que a utilização de botões

tipo *switch* facilita a interação e a compreensão, se comparados a movimentos gestuais, ou no original *gestures*.

2.4. Dispositivos para Interface com o Usuário

Mesmo que um sistema possua um elevado grau de autonomia, como veículos inteligentes, ainda é necessária a existência de uma interface entre homem e máquina. Esta interface permite uma realimentação do sistema pelo usuário, seja na forma de corrigir erros ou prover novas tarefas.

No seu desenvolvimento, a interface deve ser pensada como função das variáveis dos sistemas a serem controlados (ligar/desligar, controle de temperatura, vazão, etc.), e características do usuário (força, altura, presença de limitações físicas, etc.). Neste tópico será realizada uma revisão dos dispositivos de interfaces homem-máquina mais comuns, listados por Dix *et al.* (1998) e subdivididos em: entrada de texto, posicionamento, e dispositivos de saída de dados. Também foram pesquisados dispositivos menos conhecidos para usuários portadores de algum tipo de limitação, de forma a organizá-los seguindo os critérios já descritos.

2.4.1. Dispositivos de Entrada de Dados

Segundo Dix *et al.* (1998), o teclado tipo QWERTY é um dos dispositivos de entrada de texto mais utilizados em computadores, embora também existam os modelos alfabético e DVORAC, nos quais apenas a disposição das teclas de letras é alterada. Diversamente do teclado QWERTY, que possui uma letra por tecla, o teclado tipo *chord* pode possuir quatro ou cinco teclas, de modo que as letras são uma combinação entre as teclas pressionadas.

Esta abordagem permite a digitação com apenas uma mão, este teclado é ilustrado na figura 2.5:

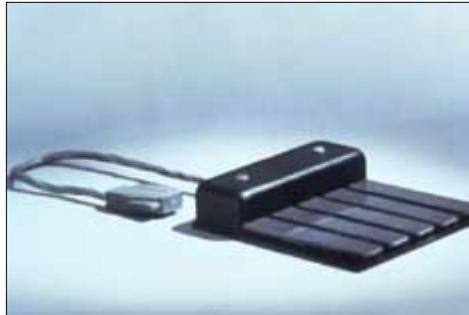


Fig. 2.5 - Teclado tipo Chord de cinco teclas (fonte: Engelbart, 2007).

Além do uso de teclados, informação textual também pode ser adquirida por meio de reconhecimento de escrita em uma área dedicada, como em palmtop, ou por reconhecimento de voz. Contudo, é necessário relevar que cada usuário possui características próprias (tipo de letra ou sotaque na fala) que, associadas a fatores emotivos, influenciam na confiabilidade da interpretação textual ou vocalizada.

Outro tipo de dispositivo para entrada é o que permite apontar ou arrastar objetos. Dentre os mais conhecidos pode se citar o mouse (convencional ou óptico), *touchpad* utilizado em laptops, o *touchscreen* ou tela de toque (utilizada em sistemas de informação para o público) e o *joystick* que pode ser acionado com a mão ou pé (em função do grau das suas limitações).

Para aplicações associadas à escrita ou desenho à mão, é possível utilizar o *tablet* de digitalização, pois este comporta de forma semelhante à uma *touchscreen*, ainda que não possua nenhum dispositivo para geração de imagens abaixo da área na qual se escreve ou desenha.

Quando é necessária a leitura de vários pontos da posição do usuário, como em ambientes de realidade virtual, dispositivos como o *dataglove* (luva com sensores para determinar posição da mão e dedos) ou técnicas para leitura do corpo inteiro (como reconhecimento de imagem) podem ser utilizadas.

Segundo Poole e Ball (2007), em casos onde há um grau de perda muito severa nos movimentos é possível utilizar o *eyegaze*. Esta técnica consiste em emitir luz infravermelha de baixa potência sobre o olho. Por meio da medição da posição de reflexão desta luz na córnea, é possível obter a localização para a qual o usuário está observando.

Atualmente este dispositivo, demonstrado na figura 2.6, apresenta um preço que pode torná-lo inacessível para os usuários finais, em torno de 14900 dólares, como observado em uma pesquisa de preços no site LC Technologies (2007).



Fig. 2.6 - *Eyegaze*, capaz de ler o posicionamento do olho (fonte: LC Technologies, 2007).

Segundo o site de referência à portadores de limitações físicas Microsoft® (2007), há opções de baixo custo para usuários que perdem movimentos dos membros superiores e inferiores, e necessitam realizar tarefas como digitar pequenos textos ou utilizar funções ativadas por botões. Entre estas alternativas é citada a utilização de um bastão que, preso a cabeça, permite utilizar interfaces pouco adaptadas ao usuário, como teclados de computador.

Outra solução são os teclados virtuais, que permitem a digitação por meio de apenas um entrada (único botão do teclado, *joystick*, contrações musculares ou sistema de sopro e sucção), como demonstrado na figura 2.7, por meio do teclado Microsoft® (2007):



Fig. 2.7 - Teclado virtual Microsoft® (fonte: Microsoft®, 2007).

É também possível a utilização de dispositivos de sopro e sucção de forma a permitir controle de cadeiras de rodas, segundo Speropoulos e Townson (2001), ou o controle de dispositivos eletrônicos como o desenvolvido por Cunningham (2007).

2.4.2. Dispositivos de Saída de Dados

Para a saída de dados, inicialmente utilizavam-se apenas sistemas mecânicos, como galvanômetros e indicadores baseados na leitura direta de transdutores (como termômetros a base de mercúrio e sensores de pressão). Com o desenvolvimento da eletrônica, as interfaces para saída de dados também se modernizaram, de forma a combinar componentes mecânicos e eletrônicos, como impressoras, ou tornando-se puramente eletrônicas.

Segundo Siegart e Nourbakhsh (2004), dentre os sistemas eletrônicos mais comuns para exibir dados, podem ser citados *displays* (CRT ou LCD) utilizados em computadores e diversas aplicações. Outro dispositivo de saída de dados utilizado são os diodos emissores de luz, também conhecidos por LED's. Os LED's possuem como vantagem não precisar do uso de uma luz externa para exibir dados, como o LCD, e um custo que o torna acessível a diversas aplicações. Em ambientes de realidade virtual pode ser utilizada a projeção holográfica, ou o uso de pares de imagens estereográficas associadas a óculos que exibem uma imagem para cada olho.

Para portadores de limitações físicas torna-se necessário um estudo de qual interface se adapta melhor a cada tipo de limitação. Usuários de computador com perda total de visão podem utilizar softwares capazes de vocalizar palavras, como o IBM ViaVoice®, que permite a leitura de textos sem a necessidade direta de um monitor.

Outra alternativa, também destinada a usuários com perda de visão são os *displays* em braile. Estes displays, como o representado pela figura 2.8, utilizam células de 6 a 8 pinos capazes de subir ou descer por controle eletrônico, exibindo mensagens de texto codificadas em braile.



Fig. 2.8 - Display braile (fonte: James, 2007)

A escolha da interface para portadores de limitações auditivas deriva do seu grau de perda de sensibilidade, assim as soluções podem variar de amplificadores de áudio até a cirurgia de implante coclear demonstrada pela USP (2007) e utilizada em casos de surdez profunda.

Capítulo 3

Interfaces para Cadeiras de Rodas Motorizadas

Usuários com dificuldades cognitivas ou com incapacidades físicas (força insuficiente ou problemas de coordenação) podem encontrar dificuldades ou apresentar incapacidade ao operar cadeiras de rodas manuais ou elétricas. Um sistema adequado de interface deve se adaptar às características dos usuários (diferenças cognitivas e físicas) e aos locais onde este será utilizado, (residência, centro de readaptação, hospital e outros). A revisão bibliográfica, a respeito de interfaces para cadeiras de rodas, permitiu delimitar objetivos considerados relevantes no desenvolvimento destas interfaces.

Para incluir uma maior diversidade em tipos de limitações, segundo Agostini e Bourhis (1995), uma interface deve apresentar uma modularidade com maior foco em software que em hardware. Segundo os autores, este foco facilita os aspectos de implementação, segurança e confiança.

No desenvolvimento das cadeiras TAO-1 e TAO-2, Gomi e Griffith (1998) definiram quatro características:

- *Suavidade e Flexibilidade*: não significa apenas utilizar materiais macios nos acabamentos, mas também prover uma arquitetura flexível para acomodar os aspectos físicos e psicológicos do usuário.

Em muitas circunstâncias, os aspectos comportamentais humanos são muito diversificados e não podem ser modelados por algoritmos. Deste modo observa-se a necessidade de um sistema flexível capaz de evitar modos rígidos de operação, desenvolvidos para cenários específicos. Como exemplo, utilizar apenas reconhecimento de voz pode impossibilitar o controle de uma cadeira de rodas em locais com poluição sonora acentuada. É necessária a presença de uma segunda interface para usuário, como um *joystick* ou mecanismo de sopro e sucção.

- *Falha segura e robustez*: o conceito de falha segura refere-se a existência de sistemas adicionais em espera. Estes sistemas garantem a segurança ao tornarem-se ativos em relação a uma possível falha.

- *Degradação esperada*: as interfaces e sistemas que se relacionam com o usuário devem evitar falhas inesperadas. Para isto, observa-se que um sistema composto de um maior número de pequenos módulos permite uma degradação mais lenta, se comparado a sistemas que possuem módulos maiores que envolvem muitas funções.

- *Evolução*: segundo Gomi e Griffith (1998), um dos motivos pelo qual as teorias de aprendizado em inteligência artificial falham, é o curto prazo de tempo no qual um sistema é testado até se obter um determinado resultado. Em sistemas incorporados e situados em um ambiente real, como organismos naturais ou robôs autônomos, a evolução é resultado de uma grande diversidade de combinação de características reais. Desta forma, uma adaptação que considere estas características pode necessitar um prazo maior que o esperado.

Segundo Harvey (1992), robôs evolutivos (ER) são uma campo emergente da ciência e tecnologia, no qual estruturas autônomas reais e virtuais evoluem para adquirir características coletivas de aprendizado em função de novas gerações. Os princípios da robótica evolutiva podem ser implementados na relação homem-máquina para, por meio de um aprendizado em longo prazo, prover uma adaptação a mudanças de operação.

A pesquisa de Min *et al.* (2002) cita quatro metas para o desenvolvimento de uma interface para usuários de cadeiras de rodas tetraplégicas:

- *Design amigável ao usuário*: focando uma aparência agradável.
- *Facilidade para vestir*: há interfaces, como o uso de eletrodos ou sensores de movimento tipo botão que são presos ao corpo, o que pode dificultar o ato de vestir ou retirar. Um exemplo é apresentado por Law *et al.* (2002), ao otimizar o conceito *EagleEyes* de Gips (1998);
- *Modo de dirigir intuitivo*: relacionado à facilidade do usuário em aprender a utilizar os comandos necessários e lembrar-se quando necessário. Esta característica assemelha-se ao conceito de consistência apresentado por Preece *et al.* (2002);
- *Baixo custo*.

É também necessário observar o fenômeno “toque de Midas”, descrito por Jacob (1991), como “Em um primeiro momento, é muito gratificante saber que é possível apenas olhar e fazer acontecer, contudo ao fim de algum tempo tudo que o usuário quer olhar executa um comando. Desta forma o desafio é construir uma interface que evite este problema”.

Seguindo a abordagem de interfaces já demonstrada, é possível relacionar um usuário, e suas respectivas formas de interação, com uma tarefa de controlar o deslocamento de uma cadeira de rodas elétrica. Deste modo é possível estruturar uma arquitetura para a relação entre usuário, interface, navegação e tarefa, como descrito na figura 3.1:

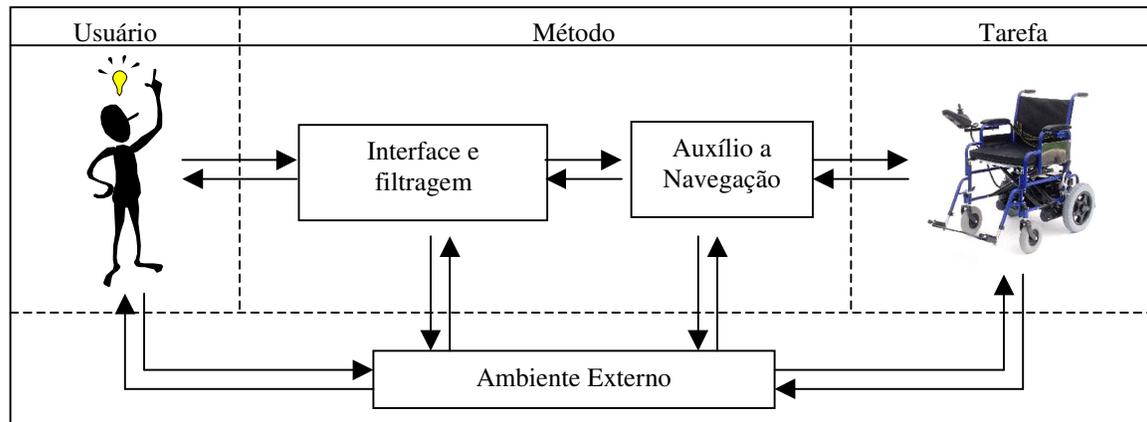


Fig. 3.1 - Interface entre usuário e cadeira de rodas.

É necessário observar novamente a presença do ambiente externo, que fecha o sistema provendo *feedback* visual para o usuário e informações para os sensores. Observa-se que a interface, filtragem, auxílio à navegação e cadeira de rodas também podem ser influenciados por distúrbios externos, alterando a resposta às interações do usuário.

Para um projeto centrado nas capacidades de interface do usuário, é necessário primeiramente relacionar as suas formas de interação com os comandos necessários à execução da tarefa, neste caso controlar a cadeira motorizada. Esta associação permite escolher qual é o melhor tipo de interface para cada usuário, como movimentos da cabeça, de fala, de dispositivos de sopro e sucção e de *joysticks* montados no queixo.

Observa-se também que a interface escolhida influenciará diretamente na relação do usuário com o método de navegação e no controle da cadeira de rodas. Esta influência associa-se a fatores como complexidade para aquisição de dados, graus de liberdade para movimentos, custo, confiabilidade e facilidade de uso.

Como exemplos, pode-se citar a pesquisa de Shaw *et al.* (1990) que descreve ser possível controlar uma cadeira de rodas com dois comandos (virar para a esquerda e andar para frente) e Law *et al.* (2002) que descrevem um controle com três comandos (virar para a esquerda, virar para a direita e andar para frente), ambos resumindo-se ao menor número de funções necessárias. À medida que o usuário apresenta uma maior diversidade de modos de interação é também possível aumentar o número de funções controladas.

Conhecendo-se as formas de interação que o usuário pode realizar e as funções da cadeira de rodas a serem controladas, o método de interface, a navegação e o controle são desenvolvidos como uma combinação de dois fatores:

- *Tipo de interface a ser utilizada: joystick, controle de voz, respiração (sistemas de sopro), movimentos da cabeça e sistemas tipo eye-gaze.* Esta avaliação deve ser focada nos graus de liberdade que o usuário possui.

- *Grau de autonomia do sistema de navegação:* definir se e quais funções de suporte para navegação (primitivas) serão utilizadas, como exemplo estão as funções evitar colisões e seguir parede. Este estudo deve ser focado nas dificuldades que o usuário possui ao utilizar uma determinada interface previamente escolhida.

Deve ser ressaltado que, na abordagem apresentada neste trabalho e ilustrada na figura 3.1, o filtro e o sistema de aquisição de sinal são considerados partes integrantes da mesma interface, favorecendo a sua modularização. Este conceito permite padronizar entradas e saídas, e alternar entre módulos de interface sem alterações no sistema de navegação e controle. Assim serão descritas soluções em interface destinadas a usuários tetraplégicos, para posteriormente exemplificar algumas das características de filtragem e condicionamento de sinal presentes nestas interfaces.

3.1. Utilização de Sensores de Movimento

Interfaces por sensores de movimento relacionam as soluções que utilizam estes sensores posicionados nos locais onde o usuário ainda possui movimentos. Desta forma, as interações de controle normalmente são resultado da combinação de dois ou mais graus de liberdade e respectivos movimentos do usuário nestes graus. É também possível combinar o movimento de um grau de liberdade em função do tempo, como em dispositivos de scan de opções (teclados virtuais).

A escolha e implementação deste tipo de interface inicia-se pela identificação de quais graus de liberdade o usuário possui. Posteriormente estes graus são utilizados para acionar sensores presos ao seu corpo ou a estrutura de uma cadeira de rodas motorizada.

Caso o usuário possua apenas um grau de liberdade disponível, em última instância, podem ser utilizados sistemas de scan de botões semelhantes ao teclado virtual Microsoft. Nestas interfaces apenas uma entrada é necessária, como em sistemas de sopro e sucção ou botões tipo interruptor.

Segundo Yanco (1998a), interfaces de scan apresentam problemas relativos à lentidão, pois é necessário esperar o tempo de scan. É citado como exemplo que, caso um pneu esteja escorregando, é preciso compensar parando a cadeira, realizando-se novas interações para a curva de correção do percurso e retornar ao modo de avanço. Sistemas de navegação automática podem melhorar esta interface acrescentando funções como evitar obstáculos e o controle de derrapagem.

Outro método de interface é apresentado por Min *et al.* (2002), ao descrever duas interfaces que utilizam como sensores resistores sensíveis à força (FSR). Estes FSR se encontram associados a um mecanismo preso ao ombro na primeira interface, e na cabeça na segunda opção. Em ambos os casos, antes de utilizar os sensores de movimentos, foram realizados levantamentos do percurso que o usuário pode realizar em cada grau de liberdade. Neste levantamento

utilizaram-se sensores magnéticos de posição (FOB), demonstrando quais graus podem ser utilizados para cada interface.

Desta forma, analisaram seis pessoas que possuem movimentos da cabeça e ombros mas, devido à coluna vertebral comprometida, são impossibilitadas de movimentar braços e pernas. Os testes de graus de liberdade de um primeiro grupo de usuários demonstrou ser possível utilizar o movimento de subir e descer dos ombros. Esta interface permitiu duas entradas independentes e quatro movimentos em função dos ombros esquerdos e direito, como demonstrado na tabela 3.1:

Tabela 3.1 - Interface por movimentos dos ombros (fonte: Min *et al.*, 2002).

Movimentação da cadeira	Ombro esquerdo	Ombro direito
Virar a direita	Cima	Repouso
Virar a esquerda	Repouso	Cima
Para frente	Cima	Cima
Para trás	Cima e Cima	Cima e Cima

O segundo método, desenvolvido por Min *et al.* (2002), utiliza movimentos da cabeça. Este método também se iniciou por um levantamento dos graus de liberdade para cada usuário: subir e descer (*pitch*) e inclinar lateralmente (*roll*). Para este levantamento utilizaram-se sensores magnéticos de posição (FOB).

Os testes de ambas as interfaces, com usuários portadores de limitações, demonstraram que há certa dificuldade em coordenar os movimentos do ombro ou da cabeça de forma a associá-los ao movimento desejado. Em comparação com pessoas normais é possível observar que pessoas com limitações demoram mais para realizar o trajeto, bem como um maior número de colisões.

Nos testes realizados, com os usuários que apresentam limitações, houve uma satisfação por poder controlar a cadeira de rodas sem a necessidade de ajuda de outra pessoa. Os usuários também aprovaram a interface que possui sensores nos ombros no quesito possuir um design agradável.

Taylor e Nguyen (2003) combinaram redes neurais ou ANN (*Artificial Neural Network*) e acelerômetros para reconhecer movimentos da cabeça, testando esta interface com seis pessoas, das quais 2 são incapazes de utilizar *joystick*.

Esta interface também utilizou um *display* para retornar ao usuário a posição lida da cabeça de modo gráfico. Este retorno permite observar o quanto o usuário se movimentou a partir de um referencial neutro. Os movimentos reconhecidos são: frente, trás, neutro, esquerda e direita, nas variações:

- *Aceno*: avanço para a posição de movimento e retorno imediato para a posição neutra;
- *Retenção*: avanço para a posição de movimento e pequena pausa antes do retorno.

Segundo Taylor e Nguyen (2003), o retorno visual não altera de modo significativo os resultados, sendo apenas útil quando este está próximo ao campo de vista do usuário. Observou-se também que o sistema de controle por redes neurais deve ser melhor adaptado para evitar condições de falso positivo, bem como estar apto a recebê-los sem prejudicar a navegação. Outra solução pode ser a inclusão de uma entrada rápida adicional para o usuário prevenir situações de emergência.

Shaw *et al.* (1990) observam que determinadas interfaces, como sopro e sucção, *joysticks* montados no queixo e movimentos da cabeça, ocasionam perda de movimentos desejados. Este fenômeno também é observado por Jacob (1991), que o reconhece como toque de Midas, por interferir nos graus de liberdade do usuário. Este fenômeno dificulta a adaptação e pode estar presente em interfaces rígidas.

Desta forma Shaw *et al.* (1990) propoem um sensor que utiliza o piscar de olhos como interface. Assim utiliza-se como sensor um par de emissor (LED) e receptor de infravermelho (fototransistor) ao lado de cada olho, permitindo detectar com qual olho o usuário pisca. Esta detecção ocorre por meio da intensidade do sinal infravermelho amostrado. Neste sistema, o LED emite infravermelho pulsado a 1 kHz, para evitar influência de interferências externas de luz.

Caso o olho esteja aberto, o infravermelho reflete na superfície do olho, e o fototransistor obtém este sinal para posterior filtragem (filtro passa faixa de 1 kHz) e conversão para nível lógico DC. Quando o olho é fechado, não há reflexão de infravermelho para o fototransistor. Esta aquisição permite obter dois estados: olho aberto ou fechado.

É necessário também considerar o período relacionado a eventos de piscar o olho de modo a evitar comandos errados. Assim, apenas são consideradas eventos de fechar de olhos de período maior que 0,5 segundo.

O procedimento de testes utilizado considerou quatro estados: olhos abertos, olhos fechados, olho direito fechado e olho esquerdo aberto. Por exemplo, para uma curva para a direita, inicialmente os olhos estão abertos, fecha-se o olho direito e o abre novamente. Os testes realizados demonstraram que o sistema não restringiu outros graus de liberdade do usuário, como movimentos da cabeça ou dos olhos, e que todos conseguiram se lembrar dos comandos e terminar o percurso.

Contudo, alguns usuários encontraram dificuldade para piscar, de modo que um dentre eles não se demonstrou capaz de fechar o olho direito de modo independente ao olho esquerdo. Quando este usuário desejava avançar, utilizava ambos os olhos, e ao realizar curvas utilizava seu olho esquerdo: para uma curva de 90° no sentido horário, percorria 270° no sentido anti-horário.

3.2. HaWCoS e a Utilização de Sensores Piezoelétricos

Um caso particular do emprego de uma interface para scan de opções é apresentado no projeto HaWCoS (*Hands Free Wheelchair Control System*), de Felzer e Freisleben (2002) e Felzer e Nordmann (2007), por permitir uma seleção ativa da opção desejada. No scan ativo é possível alternar entre os comandos desejados sem esperar o tempo de varrimento, aumentando o tempo de resposta.

O objetivo do projeto HaWCoS, é desenvolver uma interface que não utilize as mãos para o controle para cadeiras de rodas elétricas e não necessite aguardar os tempos de scan para selecionar opções. Deste modo, o usuário necessitará utilizar de apenas um grupo muscular que

possua a capacidade de se contrair voluntariamente, como músculos faciais, queixo ou dedo. Na implementação atual utiliza-se a musculatura da sobrancelha.

Esta implementação, por piezoelétricos associados a contrações musculares, é fundamentada na observação de Felzer e Freisleben (2002) de outras formas de interface e seus respectivos problemas. Como exemplo, segundo Gips (1998) a aquisição por EOG (potencial electro-oculografico) amostra as tensões musculares da ordem de 100 microvolts. Este baixo nível de tensão facilita a interferência de contrações musculares próximas.

O método utilizado por Felzer e Freisleben (2002) é sensível a contrações musculares, o que permite tempos de resposta superiores a métodos de EEG (eletroencefalografia), nos quais os tempo de resposta podem ser superiores a 2 segundos. Sistemas EEG também são sensíveis a contrações musculares próximas, resultando em falsos positivos, como demonstrado por Kirkup *et al.* (1997).

Interfaces que se utilizam de comando por voz também são sensíveis a outras fontes sonoras, como conversas com pessoas próximas ou locais com muitas fontes de ruído. Estas limitações podem impossibilitar a navegação em locais abertos ou conversar e dirigir ao mesmo tempo.

Desta forma, segundo Felzer e Nordmann (2007), a utilização de sensores piezoelétricos para as contrações musculares apresenta como vantagem uma maior robustez a fontes de ruídos. Estes sensores piezoelétricos são presos superficialmente a testa por meio de uma faixa elástica, como ilustrado na figura 3.2, permitindo converter a pressão mecânica recebida, pelas contrações musculares em uma tensão.



Fig. 3.2 - Sensor piezoelétrico e amplificador (fonte: Felzer e Nordmann, 2007).

Esta tensão é amplificada e enviada para um microcomputador laptop. A arquitetura mais recente de Felzer e Nordmann (2007) já permite substituir este laptop por uma interface embarcada na cadeira de rodas. O microcomputador, ou dispositivo embarcado, são os responsáveis pelo reconhecimento e execução dos comandos da interface. Para isto, são utilizados três estágios (ou modos) de interface: configuração, *switch* e controle, como ilustrado na figura 3.3. Há também um quarto estágio oculto, dedicado a testes.

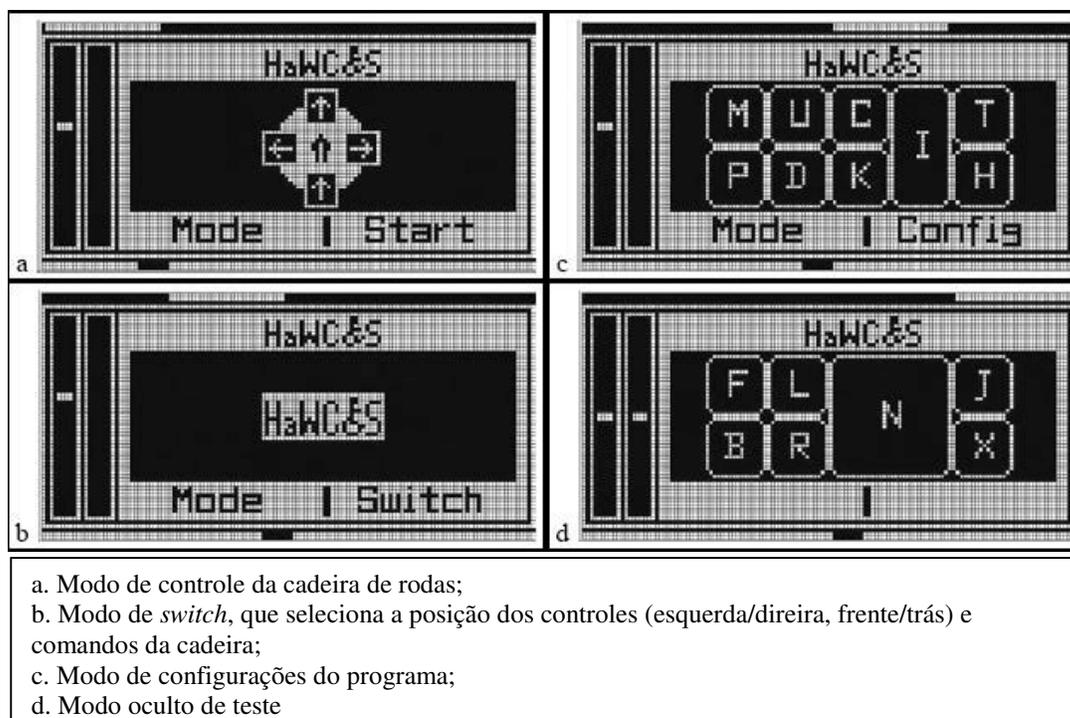


Fig. 3.3 - Estágios do sistema HaWCoS (fonte: Adaptado de Felzer e Nordmann 2007).

Estes estágios podem ser acessados quando necessário. Desta forma, é possível utiliza-los em uma ordem diferente ou avançar diretamente para o modo desejado. O modo de configurações ajusta a sensibilidade do sinal obtido do usuário, e os tempos para os comandos de clique simples (uma contração muscular) ou duplo clique (duas contrações musculares).

Para o ajuste de sensibilidade, são definidos o ganho e uma constante de *threshold*. Esta constante que permite reconhecer quando ocorre um evento de clique por meio da borda de subida do sinal, ou seja quando este ultrapassa a constante de *threshold*. Outro parâmetro que permite ajuste é o tempo entre os dois cliques, permitindo diferenciar um comando de clique simples ou duplo para cada usuário. É necessário observar que, quando uma função de duplo clique ocorre, o primeiro clique é ignorado. A figura 3.4 exemplifica estes parâmetros como a constante de *threshold* (T) e o reconhecimento de um evento de duplo clique (período definido como máximo de 1,2 segundo para duplo clique).

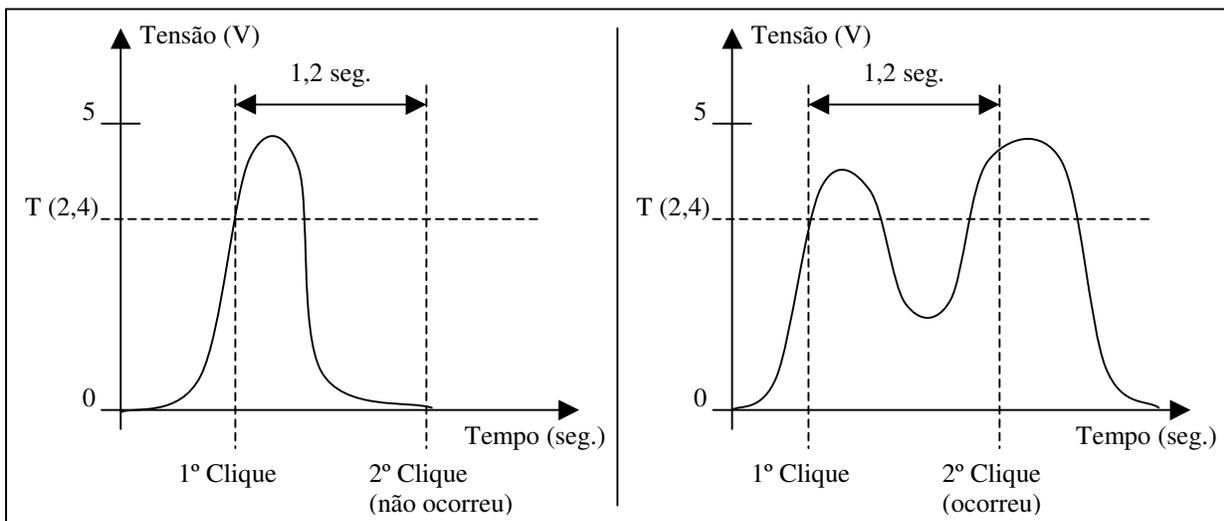


Fig. 3.4 - Reconhecimento de evento de clique simples (esquerda) e duplo clique (direita) (adaptado de: Felzer e Freisleben, 2002).

Para um ajuste adequado, é também necessário o correto posicionamento do piezoelétrico para cada usuário e suas respectivas características. Este ajuste é realizado de modo iterativo: o usuário utiliza o piezoelétrico em posições diferentes e contrai voluntariamente o músculo associado. A medida que os resultados permitem uma melhor sensibilidade ao músculo escolhido, define-se a melhor posição.

Experimentos com diferentes usuários demonstram que usualmente este tempo de posicionamento é inferior a 2 minutos. Este tempo também tende a ser reduzido a medida em que se conhece melhor as posições em que há uma maior sensibilidade para cada usuário.

Quando o sistema HaWCoS não se encontra conectado a cadeira de rodas, o modo de controle pode ser utilizado como simulação. Esta simulação permite treinar os comandos utilizados, de modo a testar se os ajustes da fase anterior estão definidos de modo adequado ao usuário. Segundo Felzer e Freisleben (2002), a simulação também permite o aprendizado para os modos de interface antes de realizar um controle real de uma cadeira de rodas.

A simulação possui dois indicadores gráficos de barra para representar os eventos de cliques: o primeiro para indicar o reconhecimento do clique e um segundo para o tempo decorrido entre dois cliques. Para a indicação de um clique utiliza-se um gráfico de barra com um indicador de *threshold*. Quando o nível de tensão supera este *threshold*, ocorre um evento de clique.

O segundo indicador representa o tempo entre dois cliques: trata-se de uma barra vertical que inicialmente é saturada a cada evento de clique, e decresce a medida em que o período definido para o duplo clique decorre. Quando o seu nível se esgota (fim do período de duplo clique) e nenhum evento de clique ocorre, define-se como um clique. Caso ocorra um segundo clique neste período, é considerado um duplo clique, como demonstrado na figura 3.4 (à direita), e o evento do primeiro clique é ignorado.

Estes indicadores de tempos para clique são necessários durante a adaptação do usuário com a interface, normalmente entre 15 a 20 minutos. Após este prazo, o indicador não é mais necessário. A fase de simulação recebeu aprimoramentos para a sua execução em microcomputador no artigo de Felzer e Nordmann (2007), que incluem um percurso virtual para testar a navegação.

No modo de controle da cadeira de rodas, os movimentos alternam-se entre as funções de avançar (frente/trás) e rotacionar (esquerda/direita) a cada clique. A cada evento de dois cliques alterna-se entre parar ou controlar movimentos (avançar ou rotacionar), como ilustrado na figura 3.5. Observa-se que o comando trás não é presente neste layout, e pode ser utilizado ao selecionar um layout diferente.

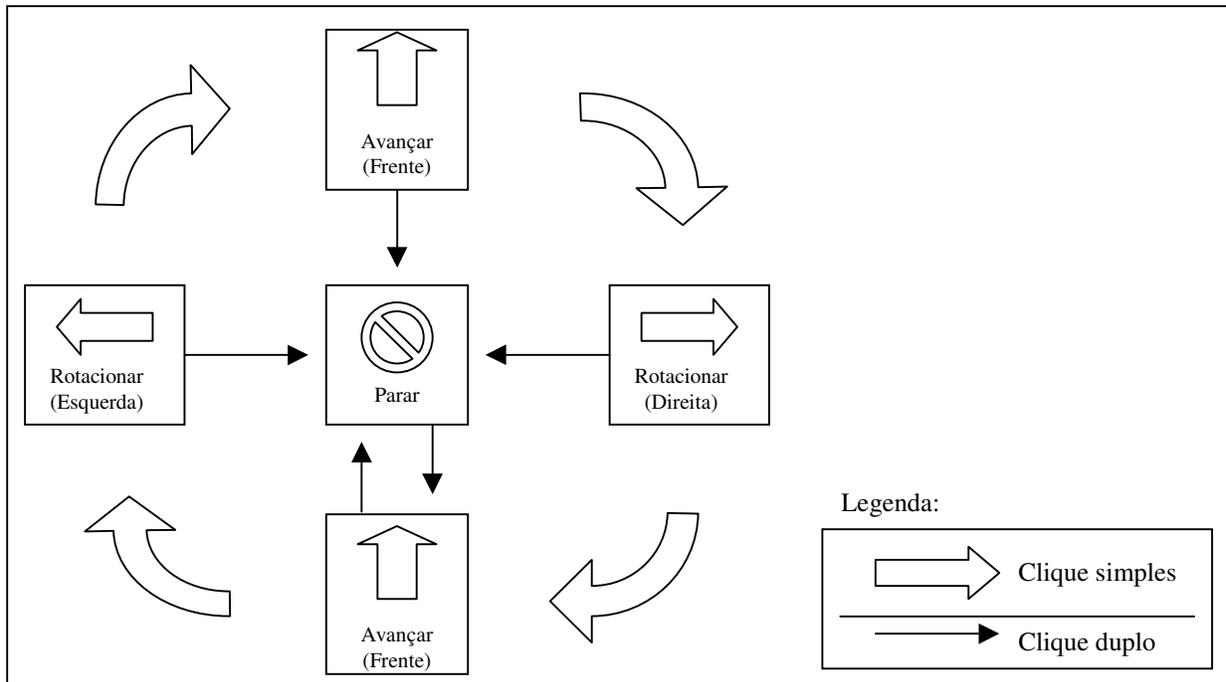


Fig. 3.5 - Controle do avanço por cliques (fonte: Adaptado de Felzer e Nordmann, 2007).

Na condição parar, é possível retornar e alternar entre os modos de simulação e ajuste a cada clique único. No estado avançar (para frente ou para trás) inicialmente desloca-se com metade da velocidade, e esta aumenta até a velocidade máxima em função do tempo. Esta aceleração permite facilitar a realização de manobras lentas, ao custo de uma redução da velocidade inicial de deslocamento em percursos maiores.

Segundo Felzer e Freisleben (2002), esta variação de velocidade não ocorre com sistemas tradicionais de *joysticks*, que permitem iniciar com a velocidade máxima. Deste modo, espera-se que o controle pelo sistema HaWCoS apresente um tempo maior que o controle tradicional para a execução de um mesmo percurso.

Para a realização de testes em situações de controle reais utilizou-se uma cadeira modelo Invacare Garant G24SX-Pró, que permite uma velocidade máxima de 50 metros por minuto. Realizaram-se 20 testes para o modo de controle por *joystick* e 20 testes para o modo de controle pelo HaWCoS. Desta forma, comparou-se os tempos gastos pelo controle de um *joystick* comum com o sistema HaWCoS na execução de três percursos, como ilustrado na figura 3.6:

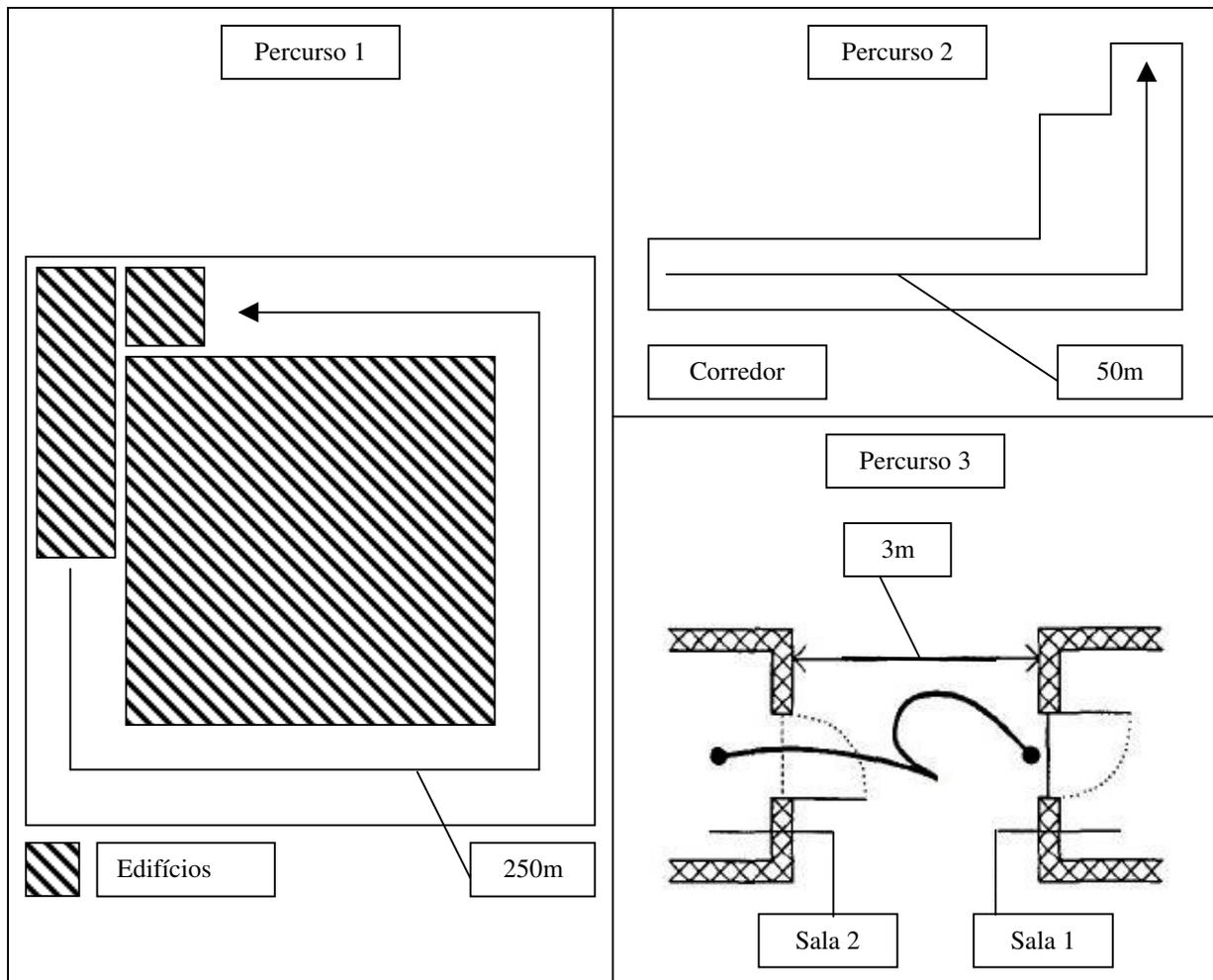


Fig. 3.6 - Percursos para teste (fonte: Felzer e Freisleben, 2002).

O primeiro percurso foi realizado em local aberto pavimentado de 250m, o segundo percurso é um deslocamento em edifício público, e o terceiro uma manobra em um corredor. No terceiro percurso, considera-se que inicialmente o usuário está de frente para a primeira porta, e realiza uma manobra para ficar de frente para a segunda porta.

A tabela 3.2 representa os tempos gastos em cada percurso, permitindo comparar os tempos gastos com o sistema HaWCoS e uma interface tradicional de joystick.

Tabela 3.2 - Comparação de tempos (segundos) entre percursos (fonte: Felzer e Freisleben, 2002).

Método \ Percurso (segundos)	Percurso 1	Percurso 2	Percurso 3
Joystick	312	67	Menor que 20
HaWCoS	441	98	Menor que 30
Aumento de Demanda (HaWCoS)	41%	46%	50%

Os tempos observados demonstram que a interface HaWCoS demanda em torno de 41% a 50% a mais que o tempo gasto para as interfaces convencionais de *joystick*. Contudo Felzer e Freisleben (2002) consideram que este acréscimo de tempo é aceitável, em vista que a interface utilizada possui apenas uma entrada. É também observado que o esforço necessário do usuário para o controle é reduzido, e permite que usuários com limitações físicas severas utilizem esta interface.

Testes realizados por Felzer e Nordmann (2007) nos percursos da figura 3.7 demonstram que há dificuldades para percorrer percursos retos.

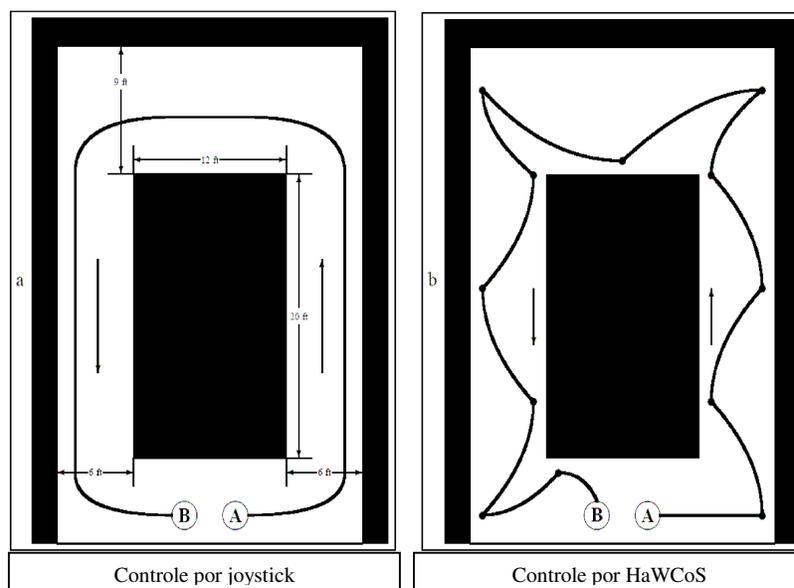


Fig. 3.7 - Trajetória da cadeira e sistema de controle (fonte: Felzer e Nordmann, 2007).

Estas dificuldades são associadas a dois fatores: o não alinhamento dos rodízios dianteiros (*caster wheel*), e a diferença de velocidade entre os motores de tração. Quando há um desalinhamento dos rodízios dianteiros, usuários com a interface de *joystick* são capazes de realizar compensações imediatas. Contudo seria necessário adicionar sensores de direção nas rodas dianteiras, para desta forma o sistema HaWCoS realizar as mesmas correções.

Outro fator de desvio se deve à diferença de velocidade com que cada um dos dois motores traciona as rodas traseiras. Para a correção deste desvio podem ser utilizados encoders nas rodas traseiras e deste modo realimentar a malha de controle.

Novos aprimoramentos foram realizados, por Felzer e Nordmann (2007), ao incluir um sistema embarcado que permite a utilização desta interface sem a necessidade de um computador. Este dispositivo embarcado utiliza um microcontrolador e um display LCD de 128 x 64 pixels, ambos alimentados por baterias da cadeira de rodas. Desta forma, tornou-se possível utilizar a mesma interface gráfica, já implementada no computador, bem como permitir escolher entre computador ou dispositivo *stand-alone*.

Segundo Felzer e Freisleben (2002) e Felzer e Nordmann (2007), as pesquisas futuras podem envolver o emprego de sensores para o aumento do grau de autonomia da cadeira de rodas, bem como o estudo de melhorias para esta técnica de interface.

3.3. Interface Remota por Computador Embarcado

A pesquisa de Ross (1996) propõe um controle remoto digital de eletrodomésticos e equipamentos de escritório por rádio frequência e infra vermelho. Este protótipo utilizou como arquitetura uma placa mãe com processador 8086, uma placa dedicada a reconhecimento e síntese de voz, e dois módulos de transmissão e recepção de dados, um para por infravermelho e um segundo para rádio frequência.

O módulo de infravermelho também possui um modo de aprendizado, que permite memorizar e reproduzir outros protocolos de comunicação infra-vermelha, para posterior controle de outras aplicações. Desta forma é possível controlar eletrodomésticos, de forma remota, por meio de uma interface adequada às características do usuário.

Testes realizados com os usuários demonstram que alguns preferem sistemas de sopro e sucção ao invés de reconhecimento de voz. Esta opção é resultado de um maior consumo de tempo para a vocalização dos comandos, e perda da privacidade quando se utiliza comandos associados a senhas, por exemplo.

Este projeto foi utilizado como referência para Ross e Sanford (1997) propor uma interface homem-máquina para controle remoto, fundamentando-se em dois princípios:

- Construir um computador que para ser “vestido” pelo usuário. Este computador deve possuir interfaces de entrada modulares e intercambiáveis em função das necessidades do usuário.
- Desenvolver um conjunto mínimo de softwares que permitam a comunicação com protocolos de diversos produtos presentes em residências.

O desenvolvimento deste novo protótipo utiliza como metodologia: identificar as interfaces adequadas para cada usuário, avaliar estas soluções por meio dos critérios de prática de uso do projeto universal, e selecionar a interface que apresente a melhor prática de uso. Desta forma, Ross e Sanford (1997) citam as opções: teclados, matrizes de teclas, *touch-pads*, telas de toque, softwares de reconhecimento de voz (associados a uma placa de som interna), sopro e sucção, sistema de rastreamento da posição dos olhos e técnicas EMG (eletromiografia) e EOG (electro-oculografia).

3.4. Utilização da Informação Facial

Reconhecendo que a maior parte das intenções humanas aparecem na face, Adachi *et al.* (1998) propõem uma interface que se utiliza dos movimentos faciais observados por uma câmera. Restringindo-se as expressões faciais e a direção da cabeça, é possível utilizar duas abordagens para a interface: na primeira controla-se de uma cadeira de rodas, e na segunda abordagem são emitidos alertas de obstáculos não observados pelo usuário.

Nesta segunda abordagem, mensagens de alerta podem ser enviadas ao usuário quando objetos se aproximam em uma direção diferente daquela observada pela face, facilitando o desvio de obstáculos. Desta forma, os comandos de movimento podem vir de uma outra interface, como a voz por exemplo.

Na primeira abordagem, controlar a cadeira de rodas, a interface pode utilizar os movimentos faciais como meio de confirmação de movimentos autônomos. Neste caso, o usuário pode acenar a cabeça para concordar com um movimento de avançar. Outro meio de confirmar os movimentos pode ocorrer por um *joystick* com atuadores internos: quando é sugerido um deslocamento para frente, este se inclina para o mesmo sentido. Desta forma, caso o usuário não deseje avançar, basta segurar o *joystick* em uma posição neutra (centro).

Outro meio de controle reconhece a posição da face, e desta forma, realiza os movimentos de controle da cadeira de rodas ao acompanhar as rotações da cabeça. Como exemplo, se o usuário move a cabeça para a direita, a cadeira se desloca para a direita. Para a obtenção da direção da face, são observados 10 pontos referenciais, envolvendo olhos, nariz, queixo e face. Estes pontos são acompanhados de modo a se obter a direção e respectivos movimentos de um plano representado pela disposição destes pontos sob a face.

É reconhecido, por Adachi *et al.* (1998), que normalmente o usuário realiza movimentos da cabeça por motivos alheios a necessidade de controle da cadeira de rodas. Para distinguir estes movimentos dos necessários para o controle, fenômeno de “toque de midas”, apenas são interpretados as situações de movimentos lentos seguidos de uma parada. Desta forma, utilizou-se um filtro de suavização, ajustado pelo número de amostras (*frames*) por segundo obtidos.

Neste filtro, amostra-se a variação de posição da cabeça por meio dos *frames*: para uma quantidade menor de amostras (inferior a 5); e para uma sensibilidade maior, utiliza-se uma amostragem maior (superior a 20).

Assim, caso a sensibilidade do filtro seja alta, movimentos rápidos da cabeça serão adquiridos com maior frequência, resultando em deslocamentos indesejados. De modo recíproco, caso a sensibilidade do filtro seja baixa, evita-se um maior número movimentos indesejados, embora o tempo de resposta do sistema aumente (torna-se mais lento). Optou-se por utilizar 10 amostras como um valor otimizado para esta relação. Um exemplo desta filtragem é demonstrado na figura 3.8:

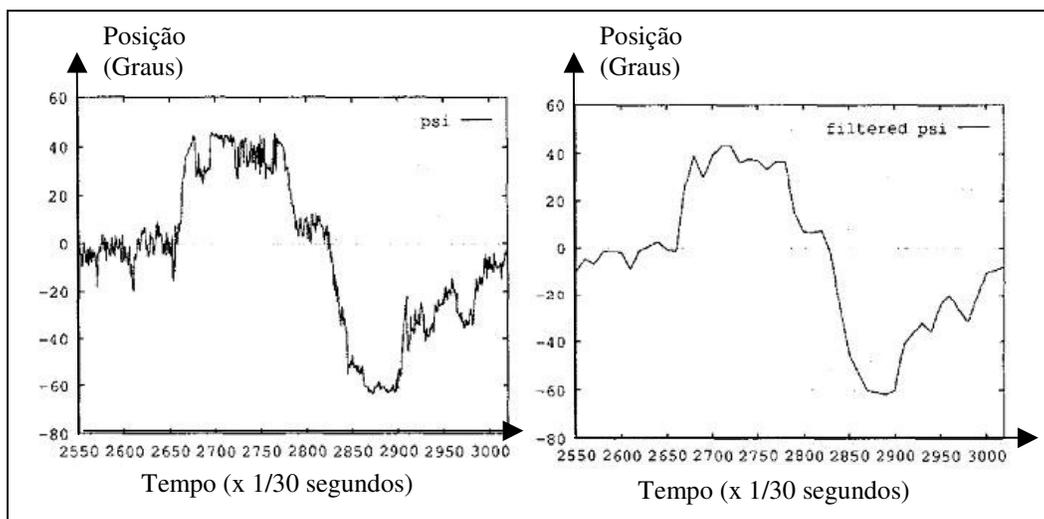


Fig. 3.8 - Sinal sem filtragem (esquerda) e com filtragem (direita), (fonte: Adachi *et al.*, 1998).

Os testes realizados demonstram que os tempos de resposta são lentos para o usuário se desviar de obstáculos durante a navegação. Deste modo, esta função é auxiliada por sonares, mediante uma expressão de confirmação do usuário, para a navegação autônoma. As informações obtidas pelo sonar, como a detecção de um obstáculo em determinada direção, permitem inibir movimentos da cadeira para a direção de uma parede, por exemplo.

Como sugestões para outros meios de interface, Adachi *et al.* (1998) sugerem o reconhecimento de expressões faciais, acompanhamento da posição dos olhos e o mapeamento prévio do local para uma navegação facilitada.

3.5. Eletrodos para Leitura de Posição dos Olhos

Em casos que o grau de limitação impossibilita o uso de sensores de movimento pelo corpo, é possível utilizar eletrodos sensíveis a estímulos gerados pela movimentação dos músculos dos olhos, permitindo uma leitura da sua posição. O projeto EagleEyes, apresentado por Gips (1998), utiliza como sensor um conjunto de eletrodos que avaliam a posição dos olhos por leituras de potenciais elétricos. Estes potenciais são amplificados e enviados para interface com o computador.

Testes com pessoas normais demonstram um tempo de 15 a 30 min. para se adaptar a interface, e escrever uma mensagem pequena no teclado virtual do monitor. Segundo Gips (1998), esta interface também foi testada com grande sucesso em 24 crianças com limitações severas, focando aspectos de diversão, comunicação e educação.

Como sensores são utilizados eletrodos sensíveis ao potencial electro-oculografico (EOG), também conhecido como potencial *electronystagmo-graphic* (ENG), correspondente ao ângulo do olho em relação à cabeça. Estes cinco eletrodos são dispostos na face, como ilustrado na figura 3.9, de modo que o par horizontal obtém o movimento horizontal e o par vertical obtém os movimentos verticais, ambos os pares possuindo um mesmo eletrodo de referencial.



Fig. 3.9 - Exemplo de utilização do *EagleEyes* (fonte: Gips e Olivieri, 2007).

A aquisição de dados é realizada por amplificadores diferenciais, em ganho da ordem de 10000. Este sinal amplificado é então adquirido por um sistema de aquisição (A/D) de um computador Macintosh ou Windows. Considerando que os sinais amostrados são da ordem de 100 microvolts, Gips (1998) reconhece que os sensores são sensíveis a movimentos de outros músculos, como piscar de olhos, movimentos da cabeça, movimentos faciais, interferência de outros equipamentos elétricos ou reações químicas entre a pele e o eletrodo.

No sistema EagleEyes é necessário ajustar os ganhos por eletrodo, que variam de usuário para usuário, pois o sistema emula o controle de um mouse. Esta emulação permite o seu uso em aplicações convencionais, indicando um clique quando o usuário observa um determinado ponto por um tempo maior. É relatado o fenômeno toque de Midas descrito por Jacob (1991), no qual comandos são executados de modo involuntário, apresentando-se como solução o uso do piscar de olhos voluntário para executar a função. Contudo deve ser observado se o usuário é capaz de piscar voluntariamente.

Usuários novos são ensinados a utilizar o EagleEyes, mantendo a cabeça fixa e, movimentando apenas os olhos para controlar um cursor em uma tela branca. Posteriormente um jogo de computador simples é utilizado para o treinamento. Testes realizados com 12 estudantes de faculdade e 10 professores e funcionários demonstraram que, após uma hora, já é possível obter um acerto de 90% a 100% no jogo de treinamento; e 11 dos estudantes e 8 dos funcionários e professores conseguiram escrever a mensagem Hello EagleEyes.

Esta capacidade de adaptação varia muito em função do usuário (de 15 min ou até meses). Como exemplo, crianças paralíticas desde o nascimento, e que nunca interagiram fisicamente com objetos, encontram grande dificuldade até reconhecerem que podem controlar o cursor. Na interface EagleEyes foi possível observar que cada usuário adquire uma forma própria de se comunicar, de forma que não é necessário seguir exatamente o movimento dos olhos, mas a forma com a qual a pessoa interage com esta interface.

Considerando que não é possível para a maioria dos usuários utilizarem um teclado virtual de 30 teclas, dada à acurácia necessária, estas teclas foram agrupadas em grupos. Para o caso de usuários com limitações severas, as perguntas se resumem a sim ou não (ex: “A sua letra está no grupo ABCD?”).

Segundo Gips (1998), é possível controlar objetos reais por meio do EagleEyes, como carrinho de controle remoto ou cadeiras de rodas, como a *Wheelesley* de Yanco e Gips, (1997). No projeto de Yanco e Gips, (1997) a posição dos olhos é associada a uma interface gráfica em um computador com setas: quando o usuário observa uma das setas por um determinado tempo, entende-se como um clique. Nesta cadeira de rodas *Wheelesley* os testes demonstram que pode ser necessário adicionar uma entrada adicional como um piscar voluntário de olhos, ou um botão que possa ser utilizado pelo usuário para desligar a interface. Estas medidas permitirão evitar comandos acidentais quando o usuário observa o ambiente local apenas. Outra observação sugere retirar a interface gráfica para usuários com maior experiência neste sistema, mantendo apenas o controle pelo EagleEyes.

Segundo Gips (1998), um ponto a ser estudado é como melhorar a inteligência do sistema EagleEyes para facilitar o controle do usuário. Como exemplo, utilizar os movimentos rápidos (30 a 120ms) que o olho humano é capaz de realizar enquanto observa uma cena ou lê um texto no computador como forma de prever onde o usuário deseja mover o cursor na tela.

Reconhecendo duas fontes de distúrbio para a aquisição de sinais e adaptando as características do sistema de interface EagleEyes, Law *et al.* (2002) realizaram otimizações para reduzir o número de sensores e facilitar o ato de “vestir” esta interface. A primeira fonte de distúrbios de leitura é referente a movimentos inconscientes dos olhos (que raramente estão parados), e a segunda fonte de distúrbios é relacionada a variações de condutividade da pele de indivíduo para indivíduo como resultados de movimentos como piscar de olhos.

Deste modo, o projeto de Law *et al.* (2002) contornou a primeira fonte de distúrbio ao interpretar os sinais recebidos como zero ou um, e não mais como uma emulação de mouse e o posicionamento contínuo em cada eixo. A segunda fonte de interferência foi contornada ao reduzir-se o número de eletrodos utilizados, de cinco sensores para três, e a sua colocação em um dispositivo semelhante a um pequeno capacete.

Os movimentos da mandíbula são observados pela presença de um potencial elétrico muito acima do observado pelos olhos. Desta forma o EOG obtido pelos três sensores permite obter o posicionamento horizontal dos olhos e a movimentação da mandíbula, relacionando todos os pares de sensores a um ponto referencial.

A aquisição do sinal se faz por meio de eletrodos de superfície (tipo metal-metal “chloride”), e amplificadores operacionais para filtrar a componente DC (*offset* relacionado à diferença de potencial da posição dos eletrodos), e amplificar em 72 dB a componente AC (resultante do estímulo muscular). Este conjunto permite um sistema estável, desde que a posição e sensibilidade dos eletrodos bem ajustada para o usuário. Uma comparação entre o método de disposição utilizado pelo EagleEyes e por Law *et al.* (2002) permite observar que os sensores verticais foram retirados, como ilustrado na figura 3.10:

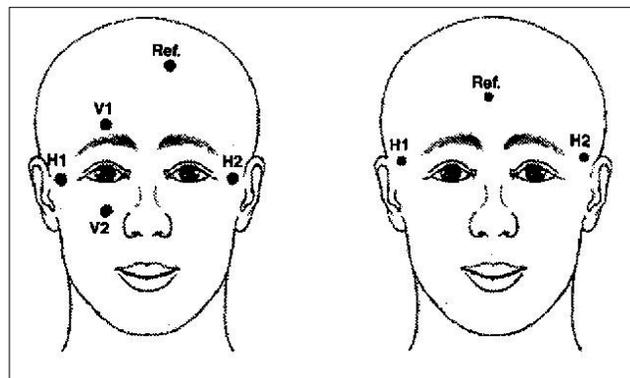


Fig. 3.10 - Comparação entre disposição dos sensores pelo método EagleEyes e Law *et al.* (2002), esquerda e direita respectivamente (fonte: Law *et al.*, 2002).

Desta forma, para desloca-se para frente ao se movimentar a mandíbula, e para a direita e esquerda ao olhar para a direita e esquerda, respectivamente, como ilustrado na tabela 3.3:

Tabela 3.3 - Movimentação pelo método de Law *et al.* (2002), (fonte: Law *et al.*, 2002).

Comando	Interação
Curva para esquerda	Olhar para a esquerda
Curva para direita	Olhar para a direita
Movimentar-se para frente	Movimentar o queixo
Movimentar-se para trás	Não implementado

O movimento andar para trás não foi implementado por motivos de segurança, posto que uma pessoa com grande grau de limitação física não é capaz de observar com clareza obstáculos que se encontram para trás.

3.6. Eletroencefalografia da Componente Alfa

Diversamente das técnicas de EMG e a EOG, que são dedicadas a leitura do potencial elétrico muscular, a eletroencefalografia (EEG), que consiste em analisar os sinais elétricos gerados pelo cérebro, utiliza-se de eletrodos dispostos na superfície do couro cabeludo. Estes sinais, devidamente tratados e filtrados, permitem diagnósticos médicos e estudos de *biofeedback*. Este estudo de meios de *biofeedback* permitirá desenvolver novas interfaces, como o cursor de computador controlado por EEG, demonstrado por Wolpaw *et al.* (1991).

Segundo Kirkup *et al.* (1997), embora tenha sido observado um progresso na área de *biofeedback* nos últimos anos, ainda são encontradas dificuldades na sua implementação. Como exemplo, pode ser citado o tempo necessário para treinamento, que pode variar de dias a meses e dificuldades para a amostragem do sinal. Estas dificuldades de amostragem se associam a características de impedância dos eletrodos, em alguns casos a alta impedância dificulta a obtenção do sinal e em outros, há um mau contato entre eletrodo e usuário.

Observando estas dificuldades, Kirkup *et al.* (1997) propõe uma interface que utiliza-se de dois eletrodos, e de uma técnica de *biofeedback* que favoreça um aprendizado amigável. Os dois eletrodos são configurados como bipolar, e dispostos na parte traseira do couro cabeludo, próximo as regiões occipital-temporal. A utilização dos eletrodos neste local permite uma aparência mais discreta, evitando demonstrar os traços de limitação do usuário.

Como *biofeedback* utiliza-se a componente alfa do espectro do eletroencefalografia, que possui de 3 a 8Hz. A sua presença não necessita de treinamento do usuário, e ocorre quando se fecha os olhos por mais de um segundo. Após este período, esta componente tende a aumentar em amplitude até se estabilizar. Para os testes utilizou-se um grupo de 11 mulheres e 10 homens, instruídos para permanecerem sem se movimentar (relaxados) e fechar os olhos por 5 segundos.

A EEG foi observada por meio de 24 eletrodos, contudo apenas o sinal dos dois eletrodos da região occipital-temporal foram estudados. Os resultados permitiram concluir que todos os usuários apresentaram um aumento da atividade alfa após fechar os olhos. Nos testes, esta frequência alfa foi observada entre 8Hz e 12Hz após 1,3 segundo ou mais após os fechar dos olhos.

Possuindo como base estes dados, Kirkup *et al.* (1997) desenvolveram um sistema que reconhece esta componente alfa e responde com um sinal de infravermelho. Esta configuração se assemelha a um controle remoto e é representada pela figura 3.11:

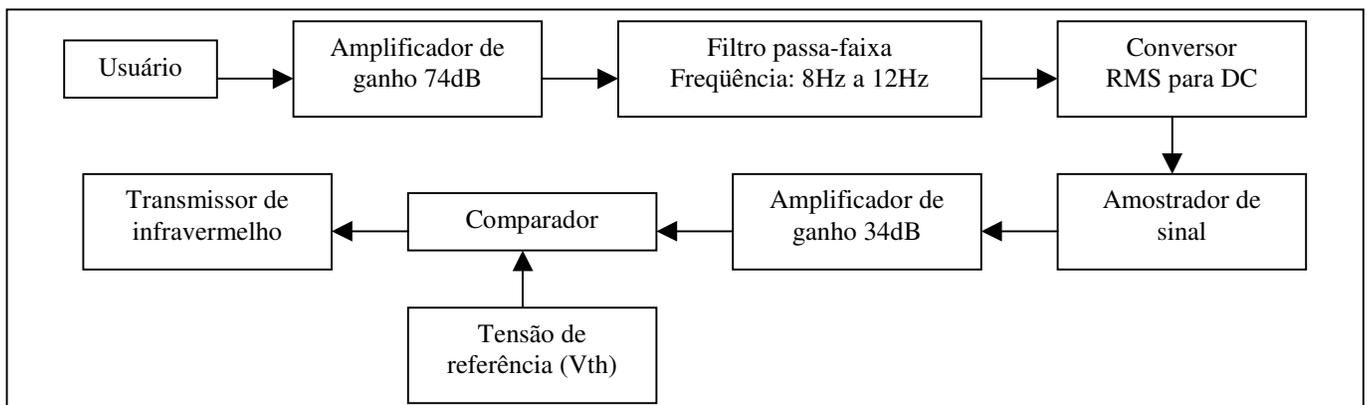


Fig. 3.11 - Diagrama do sistema de controle pela componente alfa (fonte: Kirkup et al., 1997).

Os experimentos realizados demonstraram situações de falsos positivos, ou seja, quando os olhos estão abertos e observa-se uma tensão de saída do comparador associada a olhos fechados (presença de componente alfa). Segundo Kirkup *et al.* (1997), estes eventos estão associados a interferências de músculos próximos aos eletrodos, como movimentar ou piscar dos olhos, movimentos da cabeça, mandíbulas e fontes externas de ruídos.

Observaram-se também, situações de falso negativo, ou seja, quando o usuário está com os olhos fechados e não é detectado o sinal do comparador em nível alto. Para corrigir ambos os casos, de falso positivo ou falso negativo, é realizado o ajuste da tensão de referência V_{th} (ilustrada na figura 3.11) para cada usuário. Este ajuste é realizado de forma iterativa, por meio de vários testes de fechar e abrir de olhos com o usuário.

Como exemplo, um dos usuários apresenta uma tensão entre 0,7V a 1,1.V para olhos fechados e 3V a 4V para olhos abertos, sugerindo um referencial de comparação (V_{th}) de 1,6V a 2,0V. A figura 3.12 demonstra que usuários diferentes apresentam um aumento da componente alfa em taxas de crescimento e amplitudes diferentes.

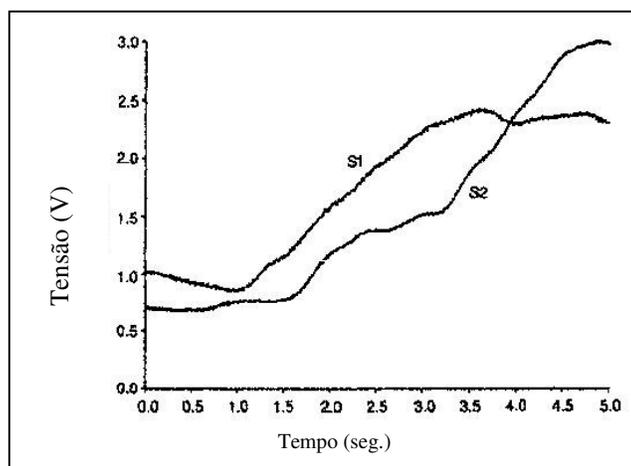


Fig. 3.12: Amostragem de componente alfa (tensão amplificada em função do tempo) para dois usuário diferentes (fonte: Kirkup *et al.*, 1997).

Desta forma, Kirkup *et al.* (1997) observam que, embora o posicionamento dos sensores não seja crítico, é necessário um ajuste de referencial de comparação para cada usuário (V_{th} na figura 3.12). Após o ajuste da V_{th} , testes realizados com dois usuários demonstram que estes são capazes de realizar as atividades de chaveamento em períodos de 2 a 3 segundos. Contudo, quando submetidos a situações de estresse, este tempo tende a ser superior a 5 segundos.

Observou-se que, para um mesmo usuário, os valores de tensão da componente alfa aumentam de 1V para 3V de modo aproximadamente proporcional em função do tempo. A taxa de crescimento constante da componente alfa pode permitir mais de uma saída de controle, desde que ocorra um ajuste adequado entre os níveis de tensão de referência para cada saída (V_{th}). Além do ajuste adequado do valor V_{th} , outro meio desenvolvido para corrigir os falsos positivos opera em paralelo ao sistema EEG e consiste em reconhecer interferências de músculos próximos pelo método da EMG. Este método reconhece que as componentes EMG possuem uma amplitude muito superior a EEG e se localiza em frequências próximas a 15Hz. Deste modo, utiliza-se uma arquitetura semelhante a utilizada na figura 3.12 operando em paralelo e dedicada a leitura da EMG. Quando um sinal relativo a contração muscular (EMG) é detectado, as leituras EEG são desabilitadas; e reabilitadas assim que este sinal proveniente de estímulos musculares não é mais presente.

Os testes realizados, com a implementação deste dispositivo adicional de EMG, demonstram que não houveram falsos positivos no período testado de 15 minutos. Contudo é observada a necessidade em definir uma tensão de referencial adicional para o sistema EMG em paralelo, bem como o inconveniente de não ser possível obter leituras da componente alfa durante distúrbios EMG.

3.7. Implantes Neurais e Comunicação Bidirecional

Pesquisas de interfaces neurais em animais, como a utilização de neurônios de uma lampreia por Graham-Rowe (2000) para o controle de robôs; e os implantes realizados por Chapin *et al.* (1999) e Talwar *et al.* (2002), que permitiram receber e enviar sinais de cérebros de ratos, demonstraram a possibilidade de utilizar implantes como meio de interface homem-máquina.

Atualmente, também é possível encontrar exemplos de implantes para interfaces homem-máquina bem sucedidos, tanto para o envio quanto para o recebimento de informações. Dentre estes exemplos podem ser citados o implante coclear para casos de surdez profunda (homem recebe informações), apresentado pela USP (2007); e o controle de cursores de computadores (homem envia informações), como demonstrado por Kennedy *et al.* (2000).

Observando ser possível receber e enviar informações por meio de implantes, Gasson *et al.* (2002) propõe um implante para interface bidirecional entre homem e máquina. Este implante constitui-se de uma matriz de 100 eletrodos sobre uma base de 4 x 4 mm, ilustrada na figura 3.14, implantada em um nervo periférico do braço esquerdo de um dos autores (K. Warwick). Desta forma foi possível o acesso a neurônios responsáveis por atividades sensoras (da pele) e motoras (músculos).

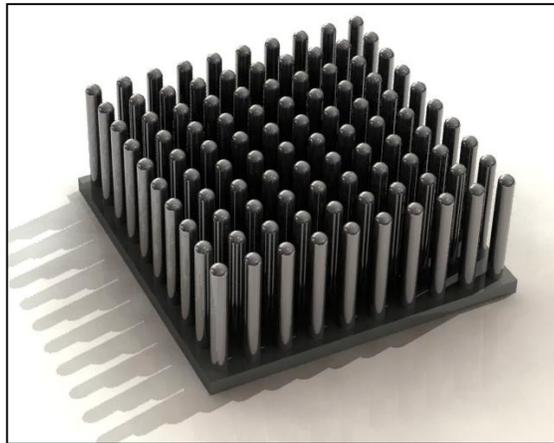


Fig. 3.13: Implante Bidirecional e Eletrodos (fonte: adaptado de Gasson *et al.*, 2002).

Segundo Gasson *et al.*, 2002, os sinais obtidos por meio dos eletrodos são amplificados e filtrados pela técnica *butterworth* em 5ª ordem, em passa-faixa de 250Hz a 7,5KHz, para um ganho de 5000. Após este tratamento, um microcontrolador realiza a conversão analógica para digital e posterior envio para o transmissor de rádio frequência (RF). A recepção de dados para o usuário ocorre de modo inverso: a recepção em RF é convertida em pulsos para o eletrodo correspondente, de modo a estimular conjuntos de neurônios associados a pele.

Observou-se que a utilização do implante não resultou em perda da capacidade motora ou sensora do usuário implantado, segundo Gasson *et al.* (2002). Esta interface permitiu o controle de três dispositivos: uma prótese moderna Southampton Hand, uma cadeira de rodas e um conjunto de três robôs. Para o controle de um conjunto de três robôs utilizaram-se as funções reagrupar ou afastar, em função dos estímulos do usuário.

A prótese utilizada possui um conjunto de sensoreamento e controle sensível a força e escorregamento de objetos, permitindo a pega de objetos com o mínimo de força necessária. O implante permitiu facilitar o controle desta prótese, bem como fechar a malha de controle ao receber informações de seus sensores.

No controle da interface com a cadeira de rodas utilizaram-se sensores para evitar colisões associados a dois métodos de interface, em ambos os casos observou-se uma margem de acerto de 90% em relação aos movimentos desejados pelo usuário.

- *No primeiro método:* os eletrodos produzem valores discretos para a movimentação na direção escolhida (frente, trás, esquerda e direita).

- *No segundo método:* utiliza-se apenas uma entrada, que permite varrer as direções a cada novo estímulo. Desta forma a direção desejada é escolhida quando o usuário interrompe o varrimento por um determinado período.

Segundo Gasson *et al.* (2002), o ponto utilizado para a instalação da prótese foi definido pelo seu fácil acesso, ainda que distante do cérebro. Deste modo, implantes realizados diretamente no cérebro podem obter resultados diferentes. É relevado que, para a execução deste implante, foi necessária autorização das autoridades do país local (Reino Unido) nos procedimentos da neurocirurgia e aplicações de interfaces bidirecionais. Deste modo, os trabalhos futuros também devem considerar os aspectos éticos envolvidos.

Um aspecto observado por Gasson *et al.* (2002) é a sensação de incorporar o braço mecânico como parte de seu corpo. Esta constatação facilita a utilização do braço artificial e os autores sugerem novas pesquisas neste sentido.

3.8. Aplicações com Implantes Cerebrais

As pesquisas de Nicolelis (2001) possuem como foco os aspectos relacionados ao desenvolvimento de interfaces homem-máquina por meio do cérebro, definidas como HBMI (*hybrid brain-machine interface*). Segundo Nicolelis (2001), futuramente estas interfaces permitirão o controle de dispositivos virtuais, eletrônicos e mecânicos de modo a possibilitar novas possibilidades terapêuticas para restaurar as capacidades motoras, sensoriais e cognitivas (como resultado das duas anteriores). Assim é possível definir dois tipos de aplicação para as HBMI's em função do usuário: receber (tipo 1) ou enviar informações (tipo 2).

As aplicações em que a HBMI envia dados ao usuário por estímulos no cérebro relacionam-se a atividades sensoriais, como um exemplo típico pode ser citado o implante coclear, descrito em USP (2007). De forma análoga a leitura e interpretação em tempo real das atividades cerebrais do usuário permite o controle de dispositivos artificiais, como braços robóticos. É interessante observar que Nicolelis (2001) define que aplicações clínicas envolvem uma combinação entre enviar e receber dados, como os implantes para usuários com epilepsia.

No desenvolvimento de uma HBMI que recebe informações do usuário, é necessário resolver duas questões: como ler estes dados, e como transmiti-los ao dispositivo a ser controlado. Para a aquisição de dados é necessário o monitoramento de um grupo de neurônios cerebrais e processar esta informação por algoritmos dedicados a extrair os comandos desejados. O monitoramento da informação pode ser realizado sob o couro cabeludo, pelo método do eletroencefalografia (EEG), como já utilizado para a movimentação de cursores de computador na pesquisa de Wolpaw *et al.* (1991).

Infelizmente as leituras realizadas por métodos pouco intrusivos, como EEG, apresentam um sinal de saída correspondente a um grupo de milhões de neurônios. Deste modo torna-se difícil isolar a leitura relacionado ao comando desejado, pois os sinais motores estão presentes apenas em determinadas populações de neurônios. Contudo Nicolelis (2001) ressalta a presença de um fenômeno, denominado plasticidade, no qual outros neurônios podem enviar informações associadas a movimentos, a medida em que são treinados pelo usuário.

Desta forma, segundo Nicoletis (2001), um desafio maior que a escolha do local adequado para a colocação do implante é o desenvolvimento de algoritmos adequados a interpretar os comandos do usuário, pois estes mudam a sua forma em diferentes intervalos de tempo.

Um exemplo de aplicação clínica para o HBMI são os casos de epilepsia, nos quais as pesquisas realizadas demonstram que normalmente há um padrão de sinal cerebral que antecede os ataques em intervalos de 2 a 5 minutos (em função do paciente). Assim, segundo Nicoletis (2001), a utilização de HBMI para pacientes em que os ataques podem ser substancialmente reduzidos por impulsos elétricos já apresentam resultados significativos. Nos demais casos, em que os ataques são inibidos por medicamentos, estuda-se o implante de micro-bombas de medicamentos no paciente.

Os estudos de mapeamento cerebral em humanos e primatas já permite identificar as regiões em que se concentram as atividades relacionadas a movimentação, sugerindo que os implantes para HBMI devem localizar-se próximos a determinadas famílias de neurônios. Desta forma, os experimentos realizados por Chapin *et al.* (1999) em ratos por meio de grupos de 50 a 100 pontos de leitura, com interpretação em tempo real, demonstraram ser possível o controle de um braço mecânico.

Para a realização dos experimentos em ratos, as interações associadas à movimentos motores bem sucedidos foram recompensadas com alimentos, permitindo ao rato associar o comando cerebral a uma recompensa. Um aspecto relevante constatado em Chapin *et al.* (1999) é que, após certo treino, os ratos apresentam contrações motoras quase imperceptíveis e concentram-se em gerar apenas estímulos cerebrais, que são necessários para as interações HBMI.

Esta constatação de Chapin *et al.* (1999) é reafirmada por Nicoletis (2001), que apresenta como possibilidade a incorporação destas novas formas de interação ao usuário como parte de seu corpo. Estas novas formas de interação podem permitir a usuários com limitações severas, como não possuir atividade muscular voluntária, a realizar interações com o meio no qual se localiza.

3.9. Filtragem e Condicionamento de Sinal

Após a escolha do tipo de interface com o usuário, estuda-se qual é melhor método para filtragem e condicionamento dos dados amostrados dos sensores sob três aspectos: Desta forma, este tópico exemplifica os aspectos associados a filtragem e condicionamento de sinal para as interfaces já descritas.

3.9.1. Filtrar Eventuais Distorções provenientes do Meio Externo

O local no qual o usuário se localiza pode interferir na leitura dos sensores utilizados na forma de oferecer referências cruzadas, como variação da luminosidade, interferências sonoras ou ruídos na alimentação elétrica. Como exemplo deste tipo de filtragem, pode ser citado Shaw *et al.* (1990) que utiliza um emissor em 1 kHz, e um receptor com filtro passa faixa na mesma frequência (1 kHz), evitando interferências do infravermelho proveniente do meio externo.

3.9.2. Condicionar uma Característica do Sensor ou do Sinal Desejado

Para sinais de baixa potência, provenientes dos sensores, pode ser necessária a sua amplificação associada ao ajuste de sensibilidade antes de sua aquisição. Como exemplo, o método EOG utiliza eletrodos para capturar tensões muito baixas, de aproximadamente 100 microvolts, segundo Gips (1998).

Outro exemplo da necessidade de amplificação e condicionamento de um sensor é demonstrado na pesquisa de Kirkup *et al.* (1997), para obtenção da componente alfa por um par de eletrodos. Nesta aquisição são necessários diversos passos de amplificação, filtragem e condicionamento para se obter o sinal desejado.

3.9.3. Ajustar uma Interface às Características do Usuário

Considerando que cada indivíduo possui características próprias, tanto comportamentais quanto físicas, é necessário que tanto a interface quanto o sistema de controle se adaptem a estas diferenças. Esta adaptação permite incluir diferentes usuários para o uso de uma mesma interface e obter uma resposta de controle semelhante.

Desta forma, é possível citar Min *et al.* (2002) que realizou um levantamento de quais graus de liberdade poderiam ser utilizados e seus respectivos cursos por meio de sensores magnéticos de posição. Também Law *et al.* (2002) observou ser necessário ajustar a sensibilidade dos eletrodos utilizados.

Técnicas de filtragem mais sofisticadas, como a pesquisa de Abellard e Khelifa (2004), podem incorporar redes neurais artificiais para a obtenção dos comandos desejados pelo usuário. Nesta pesquisa, o algoritmo de redes neurais foi abordado pela técnica redes de Petri (*Petri Net*) que facilitou a sua implementação em FPGA (*Field-Programmable Gate Array*). As redes de Petri, definidas por Alhmana (1983), permitem o controle de dados em situações de multiprocessamento, como: compartilhamento de barramento, o controle de conflitos de memória e a troca de dados entre processadores.

A Implementação desta rede neural pelos pesquisadores Abellard e Khelifa (2004) ocorreu por intermédio do projeto FRACAH, que adaptou uma cadeira de rodas comercial com diversos sensores: *on-off*, *joysticks*, comandos de cabeça, por pé, por sopro, por voz, sonares e câmeras para reconhecimento de imagem. Esta diversidade de sensores permite que o sistema se adapte ao usuário, e não o contrário. A utilização de diversos sensores também permite obter uma redundância de sinais de entrada.

Como exemplo de redundância, citam a possibilidade de reconhecer expressões do usuário, de medo ou surpresa, por meio de uma câmera. Esta interpretação permite identificar situações de insegurança por meio das expressões do usuário. Para a implementação da interface, Abellard e Khelifa (2004), utilizaram um *joystick* associado a uma rede neural treinada,

prioritariamente, para corrigir dois aspectos do usuário: limitações de movimentação e tremores na mão.

As limitações de movimentos podem ser corrigidas por meio de uma multiplicação por uma constante proporcional nas direções em que estas limitações ocorrem. Para a correção do tremor, Abellard e Khelifa (2004) sugerem duas abordagens:

- A primeira se destina a usuários com pouca amplitude de movimentos involuntários, reconhecendo e delimitando zonas de movimento.

- Uma segunda abordagem destina-se a usuários com maior amplitude de movimentos involuntários (tremor). Neste método, realiza-se uma amostragem das últimas posições do *joystick* que é comparada com os erros de movimentação comuns do usuário.

Desta forma, realiza-se o ajuste do número de amostragens como forma de otimizar uma relação que permite maiores margens de reconhecimento (menos amostragens) e a redução de falsos positivos (maiores amostragens). Estes ajustes de amostragem, definidos em valores próximos de 5 ou 6, permitiram obter um ponto de ótimo com confiabilidade de 83,3% e 83,5%, respectivamente.

Os tópicos seguintes descrevem algumas soluções de interface para usuários com severo grau de limitações; métodos para obtenção dos tempos de interação na relação entre usuário e interface; e uma descrição da utilização de filtros e condicionamento de sinal para interfaces. Para a delimitação do escopo dos tópicos seguintes consideram-se dois aspectos: primeiramente observa-se que as soluções propostas priorizam usuários incapazes de operar interfaces convencionais de cadeiras de rodas motorizadas, como *joystick*.

Um segundo aspecto é que este capítulo possui foco em interfaces de entrada, considerando-se que os usuários são capazes de utilizar interfaces de saída voltadas à visão, como *displays* CRT, LCD e indicadores com LED's. Caso o usuário não seja apto a utilizar uma interface voltada à visão é também possível utilizar-se interfaces tácteis (mecanismos com vibração ou braile) ou sonoras (*beep's* ou vocalização por software dedicado).

Capítulo 4

Sistemas de Navegação e Sensoreamento

Diversamente da escolha do método de interface, cujo foco são os graus de liberdade que o usuário possui, o sistema de auxílio à navegação deve focar as necessidades geradas pelas limitações do usuário. Como exemplo, a redução de esforço do usuário ao precisar desviar-se de obstáculos e a compensação dos escorregamentos nas rodas. Deste modo, este tópico descreve o que são funções de auxílio à navegação, os graus de autonomia e exemplos aplicação em cadeiras de rodas.

Estas funções de auxílio a navegação, também conhecidas como primitivas, permitem uma navegação facilitada pela colaboração entre usuário e cadeira de rodas. Nesta colaboração, as funções primitivas reduzem o esforço do usuário (físico e mental) ao realizar tarefas como seguir paredes, desviar-se de obstáculos e deslocar-se para um determinado local.

A escolha das funções primitivas a serem implementadas depende dos objetivos de cada projeto, capacidade de processamento implementada, necessidades do usuário, tipo e modo de implementação dos sensores. Segundo Bourhis e Agostini (1998b), um sistema de auxílio à navegação eficaz deve ser simples e utilizar o mínimo de comandos possível, para que o aprendizado ocorra de forma facilitada, caracterizando o conceito de interface consistente apresentado por Preece *et al.* (2002).

Outra característica, que é ponto comum entre diversos autores, observa que o sistema de auxílio à navegação deve sempre permitir a intervenção humana, ainda que a navegação possua grande autonomia. Esta observação ocorre tanto por motivos de segurança (Bourhis e Agostini, 1998a), quanto para aumentar a confiança do usuário (Bourhis e Agostini, 1998b), ou mesmo para prevenir erros de falsos positivos da interface (Yanco 1998a, descrevendo características do EagleEyes).

Um sistema de auxílio à navegação, que pode possuir uma ou mais funções de auxílio ao usuário, aumenta o seu grau de autonomia à medida em que controla um maior número de funções antes associadas ao usuário. Segundo Rasmussen (1983), a autonomia de um usuário ou sistema de navegação pode ser classificada três níveis:

1. *Skill Based Behavior*: Reações por reflexos, como desvio de obstáculos;
2. *Rules Based Behavior*: Reações padronizadas associadas a situações conhecidas, como seguir paredes;
3. *Knowledge Based Behavior*: Reações em função do conhecimento de situações semelhantes e empregadas em situações desconhecidas, como rotear o melhor caminho para um determinado local.

Bourhis e Agostini (1998b) dividem estes níveis de autonomia sob dois pontos de vista: usuário como supervisor e sistema de auxílio a navegação como supervisor, como indicado na tabela 4.1:

Tabela 4.1 - Métodos de controle em função do grau de autonomia (fonte: Adaptado de Bourhis e Agostini, 1998b).

Nível de autonomia	Humano como supervisor	Sistema de auxílio a navegação (SAN) como supervisor
1	Usuário desvia-se de obstáculos não encontrados pelo sonar	O SAN desvia-se de obstáculos não encontrados pelo usuário
2	Usuário ajuda na execução de primitivas (seguir parede), indicando o melhor caminho para desviar de obstáculos	Em função do local, o SAN sugere o melhor primitivo ao usuário. O primitivo escolhido pelo usuário será executado apenas pelo SAN.
3	O usuário escolhe qual é o melhor roteamento de caminho mapeado pelo SAN para um determinado local	Supõe-se que o SAN possui melhor conhecimento do local que o usuário. Deste modo, em função do local desejado pelo usuário, o SAN indica o melhor caminho ou avisa quando o usuário se desvia do caminho adequado.

Desta forma é possível reconhecer diferentes tipos e graus de autonomia para funções de auxílio a navegação, como também observado por Yanco (1998a) na determinação uma hierarquia entre os comandos primitivos, de forma que um comando tipo desviar-se de obstáculos possua prioridade sobre seguir corredor.

O emprego de primitivas, ainda que utilizados no 1º nível de autonomia já é suficiente para uma melhora significativa para a navegação. Segundo Yanco (1998a), ao realizar testes acrescentando funções como evitar obstáculos e controle de derrapagem, em uma interface de scan de botões, observa-se um ganho de 25% de tempo de percurso e 71% menos cliques se comparado a sistemas convencionais sem auxílio ao usuário. Bourhis e Agostini (1998a) também observam ser possível desenvolver sistemas com menor custo à medida em que se aumenta a colaboração humana, e se reduz o processamento necessário.

Partindo da premissa que sistemas de interface que utilizam uma entrada e varredura por LED's para pessoas tetraplégicas são lentos, Bourhis e Agostini (1998a) propoem um sistema de navegação que combina primitivas e mapas em dois graus de autonomia:

- *Automático*: avaliando o caminho que está sendo percorrido (por exemplo: cozinha para o banheiro), reconhecendo portas fechadas e outros obstáculos, sugerindo novos caminhos e facilitando o controle por uma entrada; contudo necessitando de um mapa local;

- *Semi - Automático*: quando não há mapa presente, e a navegação automática não é possível, permitindo usar apenas os primitivas como seguir parede ou desviar-se de obstáculos. O sistema também pode propor primitivas, como seguir parede ao invés de simplesmente evitar obstáculos, sugerindo também uma hierarquia entre os primitivas resultante da situação encontrada.

Os tópicos seguintes descrevem exemplos de cadeiras de rodas que se utilizam de primitivas como meio de auxílio ao usuário. Inicia-se pela descrição do exemplo mais antigo encontrado de funções de auxílio para navegação de cadeiras de rodas.

Esta arquitetura, desenvolvida por Madarasz *et al.* (1986), pode ser utilizada como referência para os modelos posteriores por incluir as funções básicas necessárias para a navegação.

4.1. Origens da Navegação para Cadeiras de Rodas

Entre as primeiras iniciativas para a implementação de graus de autonomia em cadeiras de rodas motorizadas, podem ser citados paradigmas da década de 80. Como exemplo destes paradigmas, Madarasz *et al.* (1986) utiliza de técnicas de navegação de veículos autônomos e as adapta para cadeiras de rodas, contudo são observadas três diferenças entre a navegação para veículos autônomos (não tripulados) e para cadeiras de rodas motorizadas:

Como primeira diferença observa-se que veículos autônomos normalmente navegam, e planejam suas rotas, em apenas um andar de um local previamente mapeado, (como um prédio, por exemplo). Cadeiras de rodas com sistemas de navegação devem estar aptas a mudanças de andares, para isto é necessário considerar diferentes mapas sobrepostos no planejamento de sua rota.

A segunda diferença se relaciona à necessidade de uma maior liberdade de navegação para cadeira de rodas, se comparado a veículos autônomos, visto que estas não devem seguir caminhos pré-determinados por fios ou faixas pintadas sob o solo. É também observado por Madarasz *et al.* (1986) que veículos autônomos convencionais não precisam se desviar de obstáculos inesperados como pessoas, ao contrário da navegação de uma cadeira de rodas, onde esta capacidade deve estar presente.

O terceiro ponto em que cadeiras de rodas e veículos autônomos se diferem, é que todas as decisões (roteamento de caminhos, desvios de obstáculos e demais funções) devem ser tomadas pelo sistema de processamento embarcado, no caso da cadeira de rodas. Quando se utiliza um veículo autônomo não tripulado, é possível a centralização de determinadas decisões em computadores externos, ainda que esta opção encareça e torne o sistema dependente da estabilidade da comunicação entre os computadores fixo e o embarcado.

Na implementação de Madarasz *et al.* (1986), caso o usuário esteja apto, o controle da cadeira motorizada pode ser realizado por meio de um *joystick* padrão, em caso contrário o sistema de navegação será utilizado. Desta forma, a arquitetura implementada utiliza para processamento um PC-IBM portátil com 320Kbytes de RAM, dois drivers de disquete, interface paralela, conversores A/D e uma interface exclusiva para uma câmera que opera em até 11 frames por segundo. Este computador foi programado utilizando módulos de programas que comunicam entre si, mas que também podem ser inseridos ou retirados de forma independente. A câmera utilizada possui 128x128 pixels de resolução e um ângulo de visão de 35 graus.

Para a informação de distância, utilizou-se um sonar Polaroid montado sob uma plataforma acima do usuário da cadeira de rodas. Considerando que a plataforma do sonar é rotacionada por um motor, é possível varrer 360 graus, em intervalos de 3 graus, cada 30 segundos e obter a leitura de objetos a uma distância máxima de 35 pés (aproximadamente 10,6 metros). No processo de deslocamento, Madarasz *et al.* (1986) observaram desvios da trajetória e também utilizaram encoders associados as rodas, permitindo corrigir as velocidades dos motores.

4.1.1. Roteamento do Caminho

O sistema de navegação deve estar apto a definir a melhor rota para, com esta informação, escolher uma seqüência adequada de primitivas para se deslocar a posição desejada. Para o roteamento do melhor caminho Madarasz *et al.* (1986) combinaram a memorização de um mapeamento prévio do local com a captura, por câmera, de obstáculos em movimento e referenciais descritos no mapeamento, como a numeração das portas.

A numeração das portas está presente no mapa da memória do veículo, e na forma de discos pretos com séries de pontos brancos presos nas portas para referencial. Desta forma, o mapeamento representa a localização de escritórios, corredores e elevadores em uma organização de três planos de informação. No primeiro plano de informação é determinado o número do andar (prevendo o uso de elevadores), no segundo plano o número das paredes, e no terceiro plano a localização das salas. Como exemplo pode ser citado o mapa da figura 4.2:

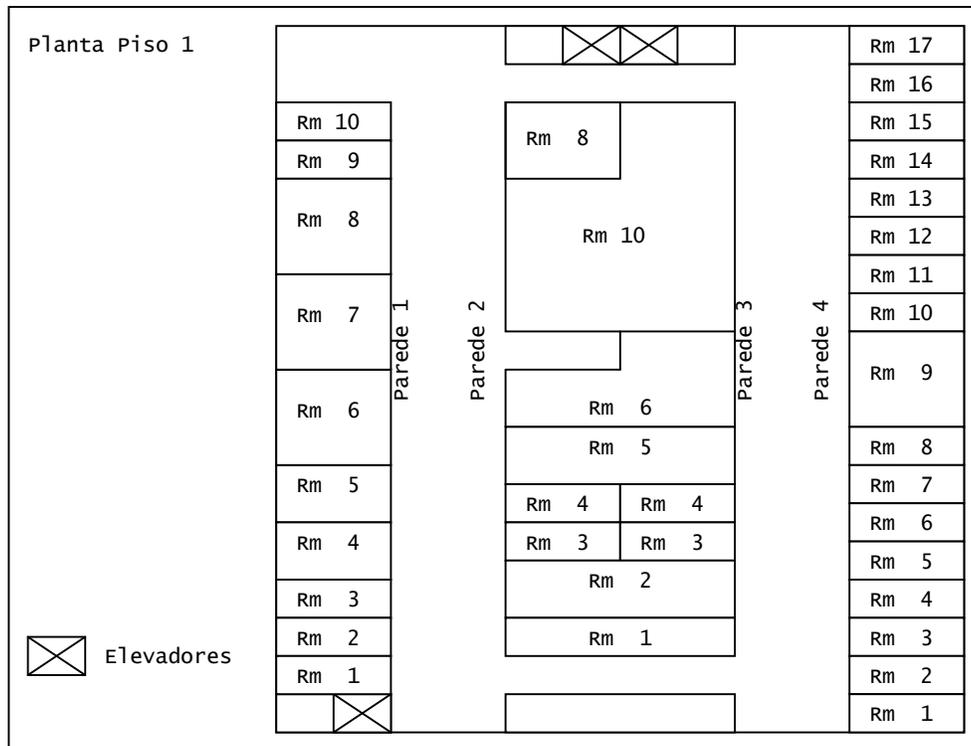


Fig. 4.1 - Exemplo de um mapa de escritório (fonte: Adaptado de Madarasz *et al.* 1986).

No decorrer do processo de navegação, obstáculos não indicados no mapa podem ser encontrados. Quando um novo obstáculo é encontrado, a este é atribuída uma variável incrementada a cada novo obstáculo identificado. Assim, as variáveis associadas a obstáculos são incrementadas, em uma unidade, a cada evento de novo obstáculo encontrado. Considerando que obstáculos com maior índice são mais antigos, e com maior probabilidade de não existência, o sistema de planejamento para navegação seleciona um caminho que possua os obstáculos com maior índice.

A navegação ocorre em duas etapas: planejamento e movimentação. Na primeira etapa, que corresponde ao planejamento, realiza-se o roteamento do melhor caminho em função do mapa local memorizado e do índice de cada obstáculo já detectado.

Este roteamento resulta em uma série de instruções do tipo: seguir em frente até um determinado referencial (corredor, sala, referência conhecida), subir elevador (alterna o mapa) e realizar curvas (90° ou -90°). Na fase de deslocamento utilizam-se estas instruções de movimentação, geradas na fase anterior, e a capacidade de identificar e desviar-se de obstáculos.

Caso não seja possível desviar-se de um obstáculo, retorna-se novamente ao planejamento da rota.

4.1.2. Sensoreamento

Como citado no tópico 4.1, para a detecção de referenciais e desvio de obstáculos o sistema de sensoreamento utilizado é composto de um sonar giratório e uma câmera fixa. Segundo descrito por Madarasz *et al.* (1986), a câmera permite uma imagem detalhada do local, contudo não pode ser utilizada como fonte primária de informação para a navegação.

Como motivos para a não utilização da câmera como fonte primária de informação podem ser citados: a não existência da informação de profundidade (distância), o grande tempo de processamento necessário e a fatores que dificultam a interpretação das imagens obtidas. Um dos fatores que dificultam o reconhecimento da imagem são variações na iluminação. Este fator foi corrigido por meio de uma fonte de luz auxiliar no módulo, permitindo compensar as imagens obtidas em diferentes momentos do dia, ajustando-se a luz solar.

Realizaram-se também testes ao associar um projetor de linha à esta fonte luminosa, permitindo uma detecção tridimensional de partes de objetos por meio do reconhecimento de imagem sobre as distorções na linha projetada. Deste modo, o sensoreamento por visão foi utilizado para detectar objetos conhecidos (para referencial), detectar as luzes no painel do elevador, alterações no local (movimentação de pessoas) e para manter a cadeira centralizada no corredor (reconhecendo e acompanhando os rodapés).

É reconhecido por Madarasz *et al.* (1986) que a utilização de duas câmeras, para obtenção de profundidade por estereogramas, associa uma demanda computacional inviável na época do projeto. Desta forma, utilizou-se um sensor tipo sonar para complementar as funções de sensoreamento para a navegação. Assim, o sonar é utilizado para determinar a distância entre a cadeira de rodas e os objetos próximos, evitando colisões, e para a orientação em relação as paredes, quando a cadeira de rodas se encontra em local desconhecido.

Segundo Madarasz *et al.* (1986), a utilização do sonar resulta em dois tipos de erros de leitura: o aumento de tamanho dos objetos e distorções na distância real. O aumento do tamanho de objetos relaciona-se a um grande ângulo de cone (25°), caso a varredura ocorra com incrementos menores que este ângulo, os objetos observados podem ser observados mais de uma vez dentro deste cone.

Esta amostragem ambígua resulta em ampliar o tamanho de objetos, e será ampliada a medida que o ângulo de cone também for maior. As leituras falsas, também observadas por Holland (2004) se relacionam a reflexões em cantos de paredes, como podem ser observadas na figura 4.2.

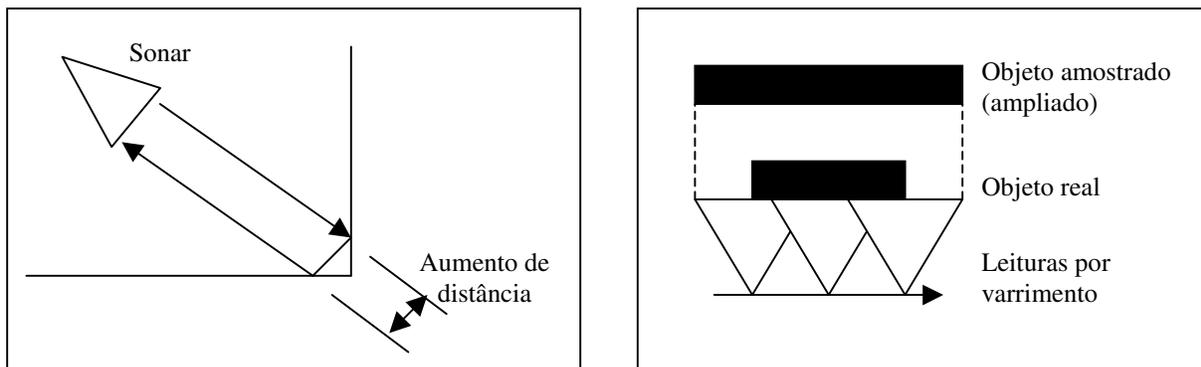


Fig. 4.2 - Erros de leitura de sonares (fonte: Adaptado de Holland, 2004).

A utilização de sonares também deve considerar a altura do objeto a ser detectado, pois caso este esteja em altura superior (mesas) ou inferior (degraus ou bolsas), estes não serão detectados, como observado por Holland (2004) e Prassler *et al.* (1999).

4.1.3. Controle de Movimento

A incerteza de posicionamento no controle de veículos autônomos está diretamente associada a qualidade de suas características construtivas. Desta forma, Madarasz *et al.* (1986) consideram que cadeiras de rodas como sistemas que oferecem baixa certeza no processo de navegação, pois desviam-se da rota planejada com facilidade devido a fatores construtivos.

Quando a cadeira de rodas é manual, o usuário realiza correções de trajetória de modo inconsciente, pois a malha de controle é fechada pela sua visão. Contudo para sistemas de navegação é necessário adquirir informações do local, por meio de sonar ou reconhecimento de imagens. A implementação de Madarasz *et al.* (1986) utiliza reconhecimento de imagens para reconhecer e se alinhar aos rodapés pretos das paredes nos corredores, como ilustrado na figura 4.3.

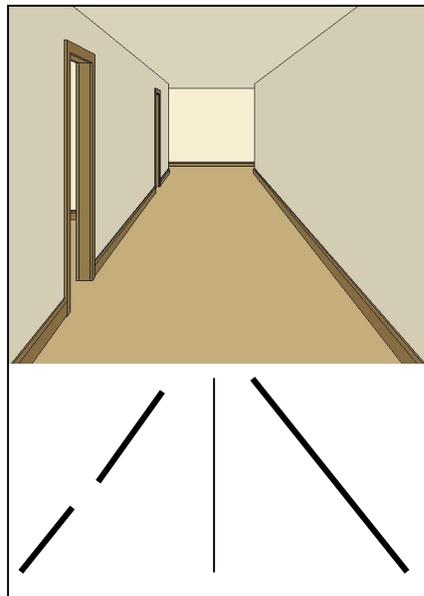


Fig. 4.3 - Utilização de rodapé preto como referencial (fonte: Adaptado de Madarasz *et al.*, 1986).

Esta solução reduz o grande processamento necessário, utilizado na solução clássica, que consiste em reconhecer toda a imagem para posteriormente planejar a rota. Contudo, mesmo utilizando um algoritmo de reconhecimento de imagens otimizado, a aquisição e reconhecimento de imagens ainda é limitado em dois frames por segundo (FPS). Esta baixa taxa de amostragem permite uma velocidade máxima segura de aproximadamente 0,254 metros por segundo, 10 polegadas por segundo.

Testes realizados em um corredor de 70 pés de comprimento (aproximadamente 21 metros) demandam um tempo médio de 98 segundos para ser percorrido. Este tempo aumenta na presença de obstáculos, como pessoas ou objetos, e alterações na iluminação das janelas durante o dia.

4.1.4. Orientação

Durante a movimentação, fatores como desvios entre a edificação real e a mapeada, ruídos nos sensores, falhas no hardware e incertezas no movimento do veículo podem resultar em desvios de trajetória e respectiva desorientação do veículo. No procedimento adotado por Madarasz *et al.* (1986), para situações de desorientação, o veículo deve procurar a parede mais próxima e segui-la até encontrar um referencial conhecido.

Para encontrar a parede são realizadas duas varreduras de 360°, com incrementos de 6°, por meio do sonar e demandando 30 segundos por varredura completa. Após executar estas duas varreduras (1 minuto), estas são comparadas para evitar a associação de obstáculos móveis e paredes. As etapas seguintes são: comparar as distâncias obtidas pelo sonar para reconhecer uma parede ou corredor, ilustrado na figura 4.4; posicionar a cadeira de rodas paralelamente e retornar ao sistema de reconhecimento de imagens para a navegação.

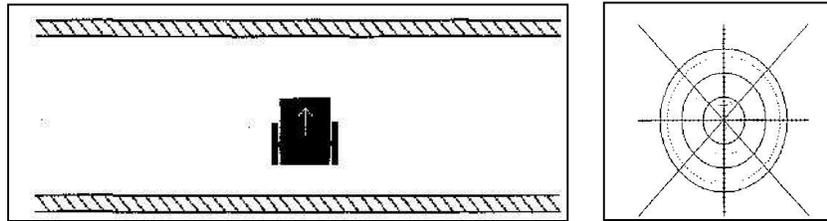


Fig. 4.4 - Reconhecimento das paredes de um corredor por sonar (fonte: Madarasz *et al.*, 1986).

Para a validação do sistema de orientação, Madarasz *et al.* (1986) realizaram mais de 100 testes sob diversas condições de luz e volume de pessoas no local. As pessoas presentes no local incluem tanto observadores passivos que percorrem o local, quanto indivíduos que buscaram desorientar o veículo, como ao passar a mão sob o sonar.

Em situações simples, como a ilustrada na figura 4.4, o procedimento de orientação realiza apenas duas iterações: varrimento do local com o sonar e movimentação, demandando 2,5 minutos. Caso o veículo esteja presente em um canto de corredor este tempo varia entre 3 a 6 minutos, ao utilizar aproximadamente 5 iterações, para reconhecer quais são as paredes do corredor e orientar-se.

Quando o número de iterações para se localizar ultrapassa de 15, como locais com muitos espectadores, o sistema pára de se mover e emite um sinal sonoro para requerer ajuda. Deve ser relevado que estes tempos, da ordem de minutos, se associam principalmente as características da capacidade do processamento embarcado disponível na época do projeto (1986).

4.1.5. Aceitação e Trabalhos Futuros

Os testes realizados por Madarasz *et al.* (1986), que ocorreram em horário e ambiente normal de trabalho, demonstraram que inicialmente as pessoas do local demonstram fascinação. Em um segundo momento as reações tornaram-se de irritação ao conviver com o veículo. As reações finais demonstram que as pessoas reconheceram o veículo como parte do edifício. O consenso entre as pessoas do local demonstrou que, se o veículo é útil, este pode ser reconhecido como algo normal.

4.2. Os projetos TAO – 1 e TAO – 2

Segundo Gomi e Griffith (1998), há duas abordagens de desenvolvimento robôs autônomos: o modo cartesiano e o modo comportamental. Robôs cartesianos são o método mais tradicional de desenvolvimento, baseando-se no reconhecimento do local seguido pelo cálculo da rota e posteriores cálculos cinemáticos e dinâmicos.

Para o desenvolvimento dos projetos TAO-1 e TAO-2 utilizou-se uma abordagem baseada no comportamento, semelhante a uma arquitetura *subsumption* que será descrita no tópico 5.3. A abordagem comportamental pode ser empregada em situações reais de deslocamento diário e permite reduzir a necessidade de recursos computacionais, quando comparada a uma abordagem tradicional.

A abordagem comportamental assemelha-se a arquitetura *subsumption* pois em ambas não é observado um programa central para coordenar a tomada de decisão, como ilustrado na figura 4.5:

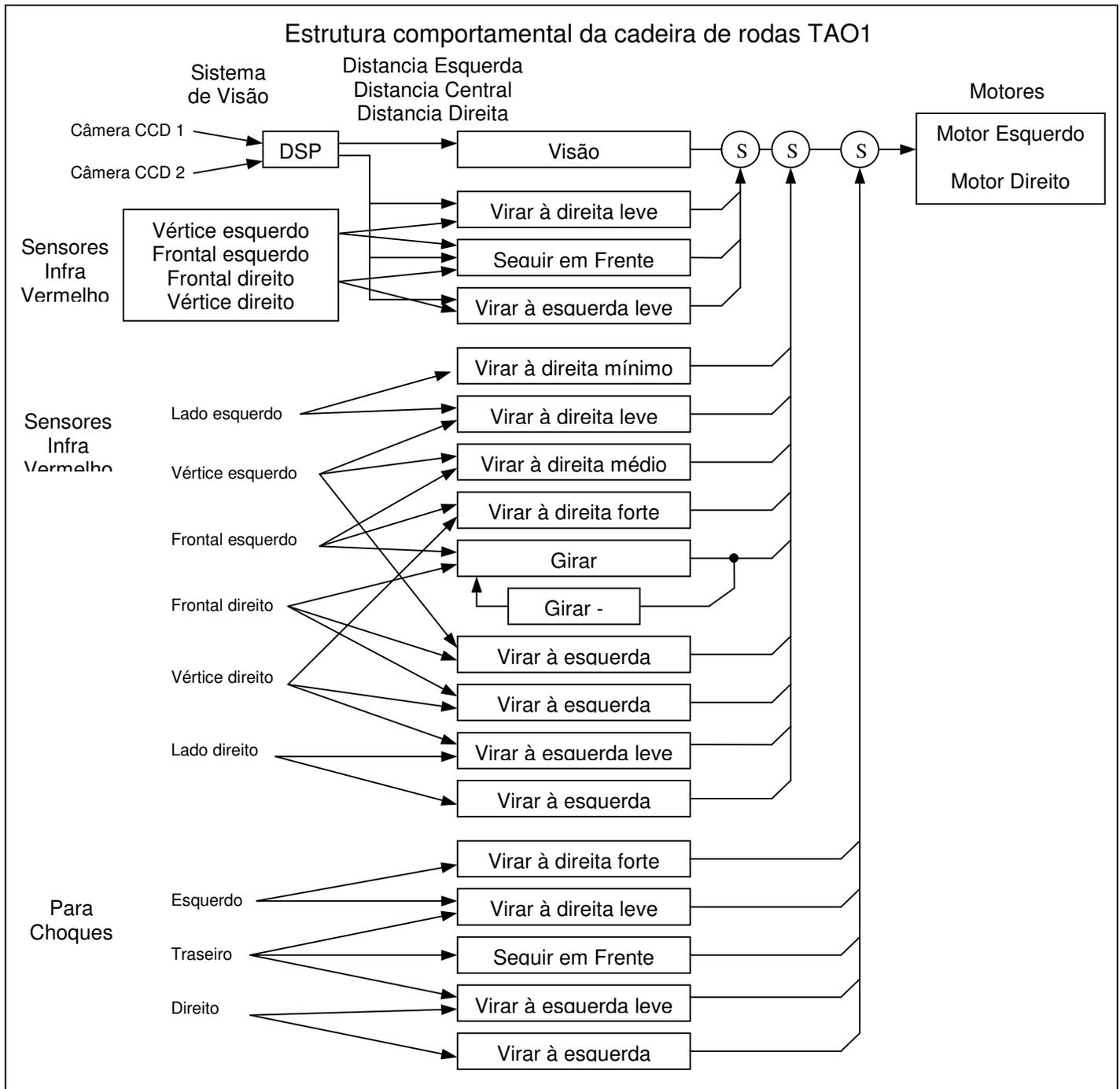


Fig. 4.5 - Arquitetura comportamental do projeto TAO-1 (fonte: Gomi e Griffith, 1998).

Na abordagem comportamental utilizada por Gomi e Griffith (1998) para a leitura de sensores, todos os módulos comportamentais sempre possuem um ou mais sensores associados as suas entradas, diferindo-se da arquitetura *subsumption*. Em uma arquitetura *subsumption*, como a apresentada por Brooks (1986) também estão presentes associações entre comportamentos e sensores, contudo a saída de um comportamento pode ser a entrada de um comportamento seguinte, algo não observado na arquitetura de Gomi e Griffith (1998) para a leitura de sensores.

A abordagem comportamental de Gomi e Griffith (1998) permitiu uma significativa redução nos tempos de implementação, pois as características dos sensores não precisam estar com grande uniformidade, se comparado a uma abordagem convencional (robôs cartesianos). Os autores Gomi e Griffith (1998) ressaltam que a arquitetura *subsumption* de Brooks (1986) também apresenta esta vantagem para a implementação de sensores, devido a semelhança entre ambas.

Na arquitetura utilizada no projeto TAO-1 observa-se também uma redução nos tempos de calibração dos sensores, como também demonstrado por Brooks (1986) pela *Subsumption Architecture*. Como exemplo, é possível o emprego de sensores infra-vermelhos de menor preço, e menos homogêneos em relação a potência e área coberta pelo emissor. Esta não homogeneidade entre os sensores não influencia significativamente o comportamento do sistema como um todo.

O projeto TAO, desenvolvido por Gomi e Griffith (1998) também possui o objetivo de desenvolver uma metodologia para projetar, implementar e testar um sistema adicional de autonomia para cadeiras de rodas comerciais. Neste sistema procurou-se implementar cinco funções de auxílio a navegação:

- *Evitar colisões*: implementada pela combinação de câmeras tipo CCD e sensores infra-vermelho. Quando a cadeira encontra um obstáculo, parado ou em movimento, primeiramente há uma redução de velocidade, seguida de uma parada ou desvio caso necessário.
- *Passar por um corredor estreito*: quando paredes próximas de um corredor são detectadas, o sistema de navegação buscará uma navegação paralela às mesmas.

- *Atravessar uma porta estreita*: determinadas doenças, como o mau de Parkinson, impossibilitam o controle fino do *joystick* para transpor locais estreitos. Neste caso o sistema de navegação reduz a velocidade e atravessa o portal evitando colisões com as paredes.

- *Manobrar em um local estreito*: quando a cadeira de rodas se encontra confinada em um determinado local, esta procura encontrar uma saída. O usuário também pode sobrepor-se a esta função e indicar um caminho que reconhece como mais adequado.

- *Navegação por pontos referenciais*: As câmeras podem ser utilizadas para identificar referenciais em um local, e desta forma navegar de uma localidade para outra. É ressaltado que não foi necessário inserir referenciais artificiais externos para a execução desta função.

No processo de amadurecimento deste projeto, inicialmente implementaram-se as funções evitar colisões e passagem por corredor estreito em 22 de agosto de 1995, na exposição IJCAI'95 em Montreal. Nesta data (1995), outras duas funções como atravessar uma porta estreita e manobrar em um local estreito, ainda apresentavam alguns problemas em sua execução; e a navegação por pontos referenciais ainda não havia sido implementada.

Atualmente todas as cinco funções de auxílio à navegação foram implementadas nos projetos TAO-1 e TAO-2, mas permitindo que o *joystick* se sobreponha e o sistema priorize os comandos do usuário sobre as funções auxiliares. O projeto TAO-1 foi construído sob uma cadeira de rodas motorizada comercial modelo 760V, e o projeto TAO-2 utilizou um modelo Suzuki MC-13P. Esta segunda versão apresentou as seguintes mudanças em relação o projeto TAO-1:

- Ajustes nos comportamentos da programação utilizada no projeto TAO-1;
- Mudanças na disposição dos sensores de colisão dianteiros para favorecer curvas mais fechadas;
- A construção mecânica da Suzuki MC-13P permitiu realizar curvas menores em raios menores que o modelo anterior. Esta otimização permitiu desativar o pára-choque sensor de colisões traseiro por ser raramente utilizado;

- Dois pára-choques laterais foram acrescentados, para evitar colisões com pessoas próximas durante curvas.

Em 1996, realizaram-se testes de navegação com o protótipo TAO-2 avaliando a performance em locais fechados e em locais abertos com neve. Os testes em locais fechados foram realizados em um ginásio de uma escola primária com várias pessoas, onde não se observou nenhuma colisão. Outros testes realizados em corredores resultaram em colisões apenas com portas de vidro, pois estas não foram detectadas por câmeras ou sensores de infra vermelho. Observou-se que sonares podem corrigir esta limitação.

Em alguns casos, Gomi e Griffith (1998) observaram que podem ocorrer contradições entre dois ou mais comportamentos. Como exemplo é citada a aproximação de um portal: quando os sensores de infra-vermelho esquerdo e direito detectam obstáculos inexistentes pela reflexão de seus próprios sinais no portal, ocorrem eventos de interferência cruzada. Este conflito resulta em um comportamento conhecido como “oscilação de Braitenberg”, descrito por Braitenberg (1984). Neste caso específico, sensores direcionados para a frente enviam um comando que autoriza seguir em frente.

Os experimentos, segundo Gomi e Griffith (1998), demonstraram que é possível a navegação em locais fechados como escritórios, corredores e desviar-se de obstáculos fixos ou móveis a uma velocidade adequada ao usuário. Contudo os autores Gomi e Griffith (1998) observam a necessidade de se realizar melhorias para a navegação em locais mais diversificados. Como exemplo, podem ser citadas otimizações na função de navegar por referenciais para permitir uma adaptação a uma maior diversidade de locais.

4.3. NavChair e a Combinação de Primitivas

O desenvolvido por Simpson *et al.* (1998) possui como objetivo uma navegação colaborada entre operador e sistema de navegação, de modo a reduzir os esforços motores e cognitivos do usuário. Desta forma, utilizou-se uma cadeira de rodas motorizada comercial, na qual adaptaram-se três unidades: um computador padrão IBM-PC (486 / 33MHz), um conjunto de 12 sensores ultra-sônicos, montados à frente da cadeira, e um módulo de interface com o usuário.

Para a interface com o usuário, implementaram-se duas opções de controle, por voz ou por *joystick*. Os comandos de voz foram implementados por meio do Verbex, que é um equipamento externo para reconhecimento de voz, como descrito por Edison (1996) e Khyber Technologies (2007).

No processo de implementação do sistema de reconhecimento de voz buscou-se implementar poucos comandos. Esta decisão deve-se ao reconhecimento de que, ao implementar muitos comandos, há uma maior dificuldade de recordar-se, inutilizando algumas funções. Deste modo, implementaram-se os comandos: parar, avançar para frente e para trás, e três velocidades para rotacionar para direita ou esquerda;.

Para a detecção de obstáculos próximos, são utilizados sonares associados a encoders. Desta forma, é possível realizar um mapeamento, com auxílio do método EERUF, desenvolvido por Borenstein e Koren (1992), para a redução de erros de leituras de sonares.

É interessante observar que, ainda que Simpson *et al.* (1998) não apresenta o conceito *subsumption* em seu projeto, é possível observar características desta arquitetura presentes em seu trabalho. Como exemplo, observa-se que o projeto NavChair organiza-se em níveis de autonomia, equivalentes a camadas *subsumption*: algoritmos de assistência a navegação (camada 0), modos de operação (camada 1) e seleção de modo (camada 2). Desta forma, a técnica de navegação apresentada por Simpson *et al.* (1998) será mantida em sua estrutura original, e aspectos relacionados a uma integração *subsumption* serão descritos em um tópico dedicado.

4.3.1. Algoritmos de Assistência à Navegação do NavChair

Em conjunto com o sensoreamento, navegação é realizada por uma combinação de dois métodos: *Minimum Vector Field Histogram* (MVFH) e *Vector Force Field* (VFF).

Minimum Vector Field Histogram

Esta técnica é uma adaptação do método *Vector Field Histogram* (VFH), já utilizado para veículos autônomos. O sistema VFH considera que o veículo utilizado é circular e possui deslocamento omnidirecional, o que pode dificultar a execução de tarefas como a passagem por portais. Outra característica da técnica VFH é que esta permite desviar-se de obstáculos com o mínimo de redução em velocidade, tornando-a favorável a veículos autônomos. Contudo, desvios rápidos podem resultar em movimentos abruptos e inesperados para o usuário, tornando-se uma característica indesejável a cadeiras de rodas.

Deste modo, Simpson *et al.* (1998) propõe uma abordagem MVFH, que combina os comandos do usuário com a capacidade de desviar-se de obstáculos. Esta abordagem identifica a presença de obstáculos, representando-os em uma matriz, em função da sua incerteza. A matriz de incerteza permite a geração de um historiograma na forma polar, como ilustrado na figura 4.6 à direita pela curva “h”.

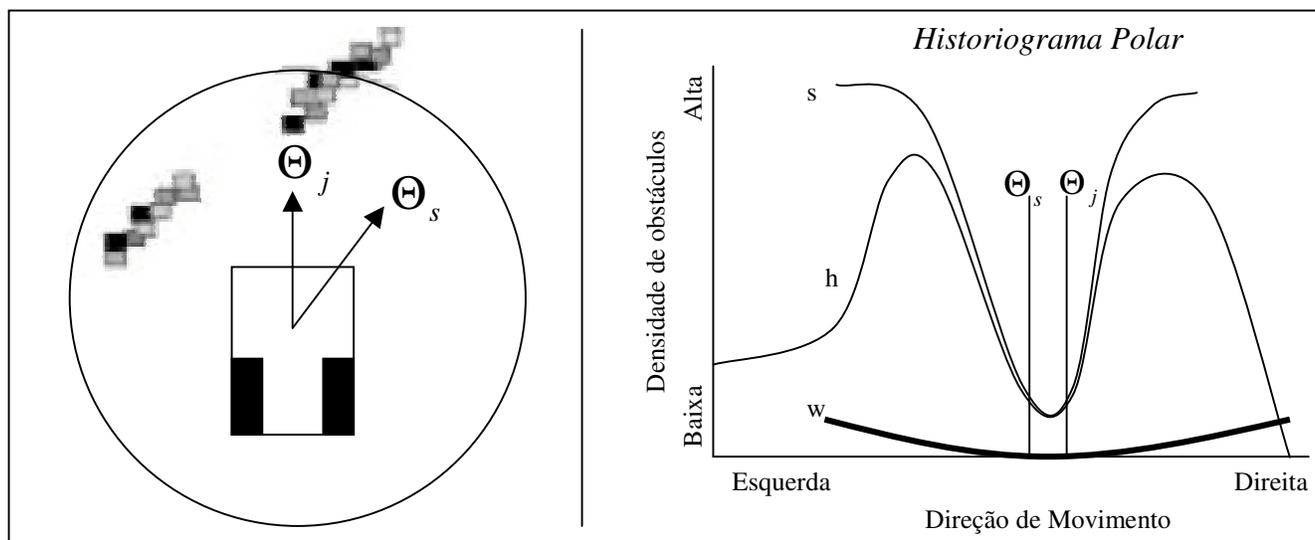


Fig. 4.6 - Método MVFH (fonte: Simpson *et al.*, 1998).

Este historiograma de obstáculos (h) é somado à curva (w), obtida pelo joystick que representa a direção desejada pelo usuário. A soma das curvas “w” e “h” permite obter a direção de movimento pela curva “s”, que permite combinar a direção desejada pelo usuário (O_j) com o desvio de obstáculos (O_s). A velocidade da cadeira de rodas é, deste modo, determinada em

relação à proximidade de objetos na direção do movimento. O método MVFH também considera as dimensões da cadeira, permitindo aproximar-se sem colisão de obstáculos enquanto a cadeira se movimenta.

Outro aspecto do método MVFH é que este permite alterar o peso da curva “w”, representada pelo comando do usuário, permitindo ajustar o grau de autonomia do usuário em relação a curva de detecção de obstáculos “h”.

Vector Force Field

O método VFF (*Vector Force Field*), ilustrado na figura 4.8, é utilizado em conjunto com o método MVFH para considerar as dimensões da cadeira de rodas em relação a posição de obstáculos próximos.

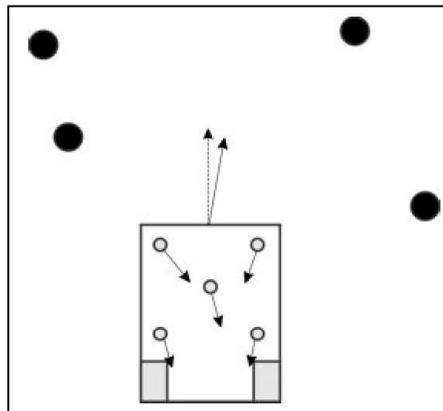


Fig. 4.7 - Método Vector Force Field (fonte: Simpson *et al.* 1998).

O método VFF localiza cinco pontos, na cadeira de rodas, para disposição de vetores de repulsão a objetos. Estes vetores são função da localização e proximidade dos objetos próximos. A soma dos cinco vetores de repulsão resulta em um vetor capaz de corrigir a direção de movimentação da cadeira de rodas.

4.3.2. Modos de Operação

Estes métodos utilizados para navegação auxiliam a navegação da NavChair, de modo a sustentar o desenvolvimento de três modos de operação para auxiliar a navegação do usuário.

Desvio Comum de Obstáculos (GOA)

Este é o modo de navegação padrão no sistema NavChair, e dedica-se ao desvio de obstáculos por meio das técnicas MVFH e VFF. O desvio comum de obstáculos é o modo de navegação que permite maior liberdade de controle ao usuário.

A utilização do modo GOA apresenta como vantagem manter uma distância mínima de obstáculos. Contudo, comparando o modo GOA a uma navegação sem assistência controlada por um usuário com experiência, observa-se que o modo GOA associa velocidades menores e tempos maiores para deslocar-se a um mesmo ponto.

Passagem por Porta

O modo de passagem por porta é utilizado em situações de transpor dois obstáculos próximos, como o portal ilustrado na figura 4.9, nas quais o modo GOA não pode ser utilizado. Desta forma, quando um modo de passagem por porta é iniciado, o método de navegação VFF é desabilitado, e utiliza-se apenas o método MVFH.

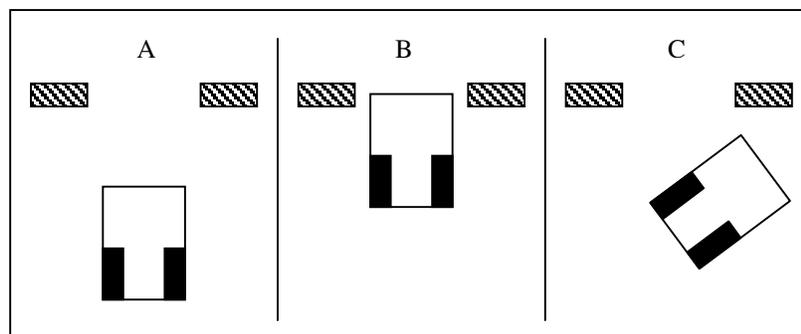


Fig. 4.8 - Utilização do modo passagem por porta (fonte: Simpson *et al.*, 1998).

Neste caso, o método MVFH reduz a velocidade da cadeira e a curva “h” torna-se um vale de amplitude acentuada, permitindo sobrepor uma prioridade em desviar-se de obstáculos aos comandos do usuário. É observada a possibilidade, na figura 4.9c, ao usuário abortar o modo passagem por porta ao apontar uma direção diferente do portal.

Este modo de passagem por porta demonstrou ser possível transpor obstáculos com distância mínima de 81,3 cm em 70% dos testes. Contudo, Simpson *et al.* (1998) observam que a performance do usuário experiente é superior à apresentada pelo modo de passagem de porta (tempo e índice de acerto). Esta queda de performance no modo de passagem de porta deve ser relevada, posto que permite a usuários com maiores limitações físicas transpor obstáculos próximos.

Seguidor de Parede

O modo seguidor de parede é iniciado quando o usuário direciona a cadeira de rodas próxima a uma parede, e desativado quando o usuário se afasta da parede. Neste modo, os sistemas MVFH e VFF são desabilitados, e a cadeira NavChair utiliza-se de um conjunto de sensores de sonar para manter-se próxima a parede. Outro conjunto de sonares é responsável por identificar obstáculos à frente e reduzir ou parar a cadeira, caso necessário. Este modo, seguidor de parede, permite velocidades maiores que o modo GOA, contudo associa um tempo duas vezes maior que o utilizado por um usuário com experiência.

4.3.3. Seleção de Modo

Para a seleção do modo de operação utiliza-se uma combinação de dois algoritmos “adaptação sugerida pelo ambiente” (ECA) e “adaptação baseada na localização” (LBA). Estas duas técnicas são combinadas por meio de modelamentos de probabilidade e incerteza, resultantes das informações do local no qual a cadeira se encontra.

Os testes realizados demonstraram que os algoritmos de seleção ECA e LBA evitaram constantes mudanças do modo de navegação, proporcionando maior conforto ao usuário. Observou-se também que o tempo de processamento para a escolha do modo de navegação não

interferiu nas demais funções da cadeira de rodas, permitindo uma resposta em tempo real nas tomadas de decisão.

Após a realização dos testes, Simpson *et al.* (1998) observaram uma queda de desempenho quando são utilizadas funções de auxílio, mas justificam que estas funções permitem a inclusão de usuários com limitações severas.

4.4. Abordagens VOA e BOA para o Desvio de Obstáculos

A pesquisa desenvolvida por Becker *et al.* (2005) e apresentada no Cobem 2005, demonstra duas abordagens para a navegação e desvio de obstáculos para robôs autônomos: *Velocity Obstacle Approach* (VOA), proposta por Fiorioni e Shiller (1993); e BOA, proposta por Becker (2000). Estas abordagens permitem a robôs autônomos detectar e desviar-se de obstáculos, móveis ou imóveis, mantendo a rota para um local pré-definido.

A técnica VOA relaciona cada obstáculo detectado a um círculo de raio R_o e o robô a um círculo de raio R_v . A soma das dimensões do robô e do obstáculo, inscritas em seus respectivos círculos, permite obter um terceiro círculo de raio R . Este círculo de raio R permite gerar um cone de colisões que possui sua origem na posição do robô, como ilustrado na figura 4.9.

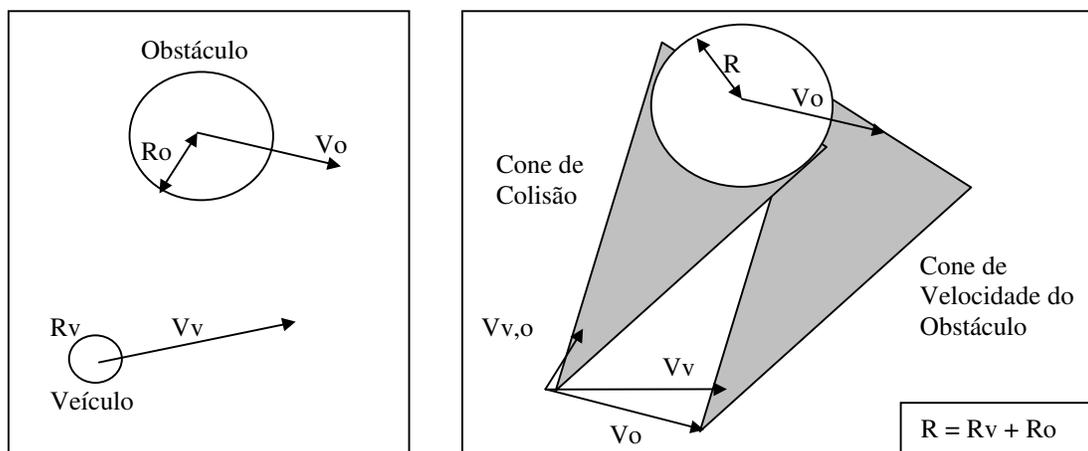


Fig. 4.9 - Obtenção do cone de colisões para um obstáculo fixo (fonte: Becker, 2000)

Considerando a velocidade de deslocamento do obstáculo (V_o) é necessária uma translação da posição do cone de colisão, permitindo obter o Cone de Velocidade do Obstáculo que é independente da velocidade do robô V_v . Desta forma, quando a velocidade do robô (V_v) encontra-se dentro do Cone de Velocidade do Obstáculo, é considerada uma rota de colisão. Assim, as relações entre o vetor de velocidade do robô (V_v) e o cone de velocidades do obstáculo resultam em três situações avaliadas a cada intervalo de tempo:

- *Rota de colisão:* Quando o ponto apontado pelo vetor V_v localiza-se dentro do cone de velocidades do obstáculo, esta situação ocorre na figura 4.9;
- *Rota de desvio:* Quando o ponto apontado pelo vetor V_v localiza-se fora do cone de velocidades do obstáculo;
- *Rota de desvio mínimo:* Quando o ponto apontado pelo vetor V_v localiza-se sobre uma das linhas do cone de velocidades do obstáculo.

Em uma situação real é necessário também considerar as características dinâmicas do robô, como o torque necessário nos motores para a realização de uma determinada curva. Desta forma, sobrepõe-se ao ponto A, e cones de colisão, um polígono com as velocidades máximas e mínimas possíveis para avançar e realizar curvas. Esta sobreposição permite delimitar as rotas de desvio por meio das áreas externas aos cones, demonstrando em quais condições de direção e velocidade para o deslocamento do robô (vetor “ V_a ”) evita-se os cones de colisão.

Assim a figura 4.10 demonstra que é possível decidir entre acelerar ou reduzir a velocidade em função de objetivos maiores, como deslocar-se no menor intervalo de tempo ou menor consumo de baterias.

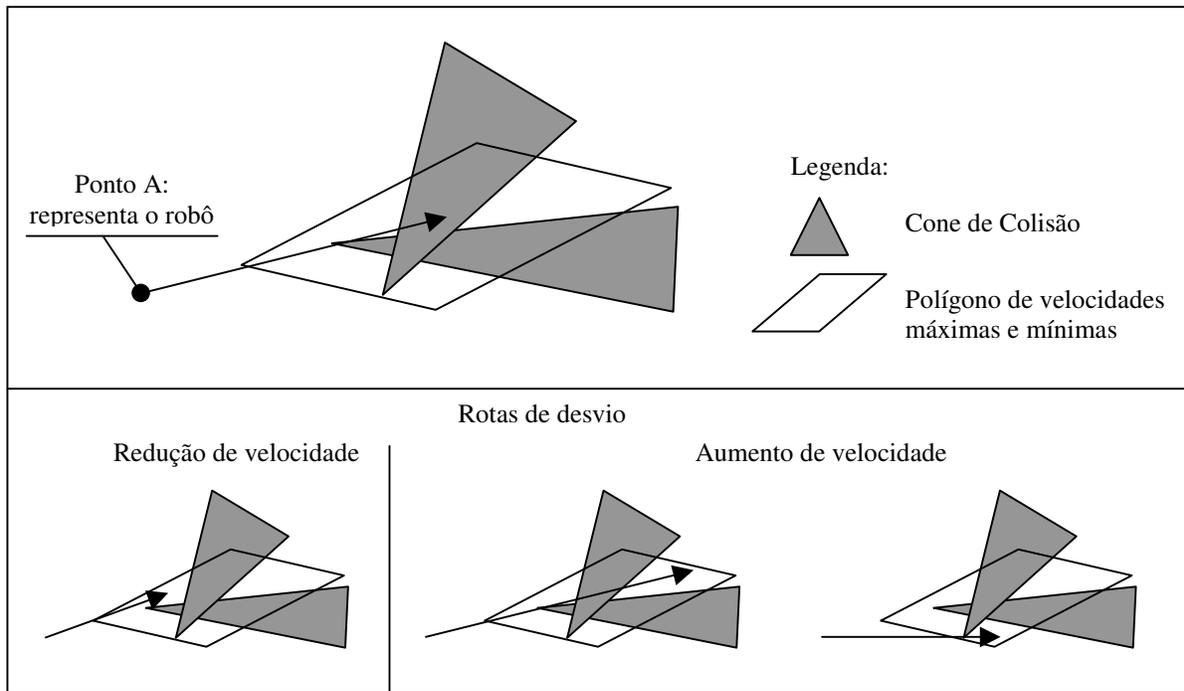


Fig. 4.10 – Determinação de rotas de desvio pelo método VOA (fonte: Becker *et al.*, 2005).

4.4.1. Abordagem BOA

Mantendo características da técnica de Fiorioni e Shiller (1993), que utiliza cones de colisão para projetar obstáculos e um vetor de velocidade do robô restrito a um polígono, Becker (2000) observa ser possível desvincular as características de translação e realização de curvas no robô modelado e contruído para a realização de experimentos. Esta desvinculação permite a Becker (2000) substituir o polígono que limita as velocidades e direções por círculos para diferentes módulos de velocidade V_a , propondo uma nova abordagem para o desvio de obstáculos fixos ou móveis denominada BOA.

Nesta nova abordagem, ilustrada na figura 4.11, mantém-se o vetor de velocidade do robô (V_a) e o mesmo conceito utilizado na construção dos cones de desvio de objetos, contudo a utilização de círculos de velocidade permite evitar uma constante alteração no vetor de velocidade V_a . Desta forma, é possível desviar-se ao alterar a direção de movimento e, apenas quando não for possível encontrar uma solução adequada para o desvio de um determinado obstáculo, alterar a velocidade de deslocamento do robô.

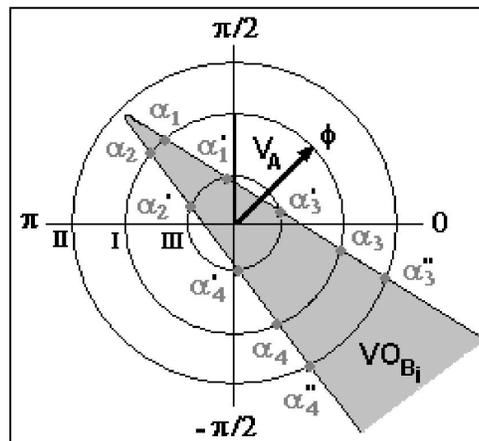


Fig. 4.11 – Utilização da abordagem BOA para o desvio de obstáculos (fonte: Becker *et al.*, 2005).

Os testes realizados por Becker *et al.* (2005), com a abordagem BOA, consistiram em uma simulação em ambiente MATLAB e posterior implementação em robô real. O robô real utilizou como sensores dois Sick LMS 200 2D a laser; 6 sensores de sonar dispostos à frente, nas laterais e atrás do robô, e um sensor de colisão montado no pára-choque. Em ambos os casos, robô simulado ou construído, observou-se a capacidade de locomoção autônoma em corredores estreitos ou largos.

4.5. MAid e a Navegação em Locais Muito Populosos

Pesquisas realizadas por Prassler *et al.* (1999) com usuários de cadeiras de rodas, demonstraram que determinados recursos de navegação implementados não são utilizados. Como exemplo estão os recursos seguir parede e passar por portais, nos quais os usuário preferem controlar a cadeira de modo manual. O projeto MAid, baseia-se em uma cadeira do rodas comercial equipada com um conjunto de sensores e um sistema de navegação inteligente. Como sensores são utilizados:

- Um par de encoders para as rodas,
- Um giroscópio de fibra óptica (Andrew RD2030),
- Um sistema constituído de 3 módulos de 8 sonares cada (um a frente e os demais nas laterais),
- Dois sensores infravermelho para distâncias curtas,
- Um *range-finder* a laser (modelo SICK 2D, PLS200), montado em uma base removível à frente da cadeira.

A integração destes sensores, com exceção do *range-finder*, foi realizada por *fieldbus* com o computador responsável pela navegação e controle. Segundo Prassler *et al.* (1999), o *range-finder* possui uma porta serial dedicada para interface com o computador, pois amostra uma maior quantidade de dados por intervalo de tempo que os demais sensores.

O sistema de navegação apresenta dois modos de operação: semi-autônomo e autônomo. No modo semi-autônomo, definido como NAN por Prassler *et al.* (1998), a cadeira evita colisões em locais fechados, como banheiros, facilitando a interação para usuários com dificuldades motoras. O modo autônomo, que Prassler *et al.* (1999) denomina WAN, dedica-se a uma navegação autônoma em locais muito populosos (movimentados) como concursos, shoppings ou centros de convenções. Neste modo (WAN) o usuário indica a posição desejada, e um conjunto de instruções como seguir parede e desviar-se de objetos é iniciado.

A arquitetura de controle e navegação ocorre em três níveis: controle básico, nível tático e nível estratégico:

- O nível de controle básico recebe e processa valores relativos ao controle dos motores associados ao deslocamento e velocidade da cadeira de rodas. Desta forma, este nível recebe um vetor com informações da velocidade de translação e rotação desejadas pelo nível tático.
- O nível tático, por sua vez, é composto de três sub-módulos: detecção de movimento, seguidor dos movimentos e cálculo de rotas evasivas. Estes sub-módulos em conjunto observam objetos, estimam o seu deslocamento e, desta forma, definem as rotas para evitar colisões.
- O nível estratégico é responsável por definir a próxima meta para o deslocamento. Embora nesta implementação seja o usuário que define os próximos objetivos, Prassler *et al.* (1999) observam que futuramente esta função poderá definir uma seqüência de metas de deslocamento, compondo um planejamento da rota.

Neste processo de deslocamento para um determinada local predefinido, há uma prioridade para o desvio de obstáculos. Deste modo, o caminho realizado entre o ponto de partida e de chegada pode não ser definido como uma reta, reconhecida a necessidade de desvios.

A avaliação do sistema de navegação foi realizada em uma estação de trens nos horários mais movimentados. Desta forma, o usuário define a distância e aponta o *joystick* para a posição desejada e o sistema de navegação desvia-se dos objetos ou pessoas no caminho. Nos testes não foram observadas colisões com nenhuma pessoa, mesmo em situações de grande tráfego. Nestas condições de grande tráfego, o deslocamento é interrompido até que o tráfego seja reduzido a uma intensidade que permita voltar ao movimento.

As únicas colisões observadas se relacionam a objetos pequenos sobre o chão, como mochilas ou malas, que não foram detectados por meio dos sensores. Como correção para os projetos futuros, Prassler *et al.* (1999) citam a utilização de dois sensores tipo sonar montados nos suportes para os pés da cadeira.

4.5.1. Detecção de Objetos e Planejamento de Rotas de Desvio

A detecção de objetos ocorre pela constante comparação, em função do tempo, dos dados amostrados pelo *range-finder* na forma de *grid* de células. Nestas matrizes, denominadas de *time stamp* por Prassler *et al.* (1999), cada célula descreve uma pequena região real. Para a observação de movimentos, é realizada uma comparação entre as *time stamp* memorizadas para encontrar suas alterações em função do tempo.

Deste modo, a comparação de uma seqüência temporal de diferentes *time stamp* permite obter a posição, velocidade e aceleração de um determinado objeto, ao estimar o seu deslocamento como linear. Esta análise permite definir quais objetos estão em rota de colisão. Para situações de muitos objetos, é atribuída uma preferência na definição da rota de desvio para aqueles objetos que estão mais próximos da cadeira de rodas.

Para a definição desta rota de desvio avaliam-se também as velocidades máximas possíveis para deslocamento, de modo a propor a opção de cruzar a frente do obstáculo ou reduzir a velocidade de deslocamento da cadeira de rodas.

Capítulo 5

Integração entre os Sistemas Envolvidos

A revisão bibliográfica demonstra que, em geral, cada autor possui foco na solução de um problema específico e, deste modo, apresenta uma solução específica de interface ou método para navegação. É também observado que os sistemas de navegação utilizados, em geral, são derivações de sistemas de navegação de veículos autônomos, como exemplos o projeto de citado Madarasz *et al.* (1986) e Simpson *et al.* (1998).

Segundo Yanco (1998b), após o desenvolvimento destas soluções centradas em problemas específicos, a fase seguinte para o amadurecimento de um projeto será a integração de diversas soluções para resolver problemas mais complexos. Esta integração ocorre por meio de uma padronização, que deve ser desenvolvida como resultado de uma análise dos pontos comuns entre os projetos. Deste modo, será necessário integrar diversas áreas de pesquisa relacionadas à interface, controle e navegação.

Outro ponto a ser observado é o momento adequado para realizar a padronização de um conjunto de módulos nos campos de interface, navegação e controle de um projeto. Esta padronização deve ocorrer somente após o desenvolvimento de soluções nestes três campos, evitando utilizar conjunto de soluções restritas a padrões.

Considerando a diversidade nas características de cada usuário, é necessário compartilhar os aspectos positivos de cada solução ao definir métodos que permitam a sua fácil integração. A utilização de protocolos em comum, associados as características de modularidade, permitem a integração de diferentes interfaces, sistemas de navegação e controle. Desta forma, este tópico apresenta meios para combinar diferentes soluções em interface, navegação e sistemas de controle.

5.1. Protocolo de Comunicação M3S

É comum observar que cada autor ou fabricante sugere soluções particulares, ou seja, os módulos de interface com usuário, sistema de navegação e módulos de potência normalmente são incompatíveis.

Considerando-se que usuários portadores de limitações possuem características particulares entre si, são necessários métodos de interface e suporte para a navegação diferentes. Observa-se então a necessidade de evitar a incompatibilidade entre os dispositivos utilizados. Um meio para evitar problemas de incompatibilidade é a utilização de uma arquitetura modular, que possa ser interligada por um mesmo protocolo de comunicação.

Assim, a integração proposta por Nelisse (1998) possui como objetivos:

- Facilidade para uso,
- Utilização segura,
- Flexibilidade: facilidade para adaptações para as necessidade do usuário,
- Custos baixos para instalação, manutenção e adaptação,
- Prover suporte para interfaces de entrada e sistemas de controle já existentes,
- Suporte para projetos de futuros dispositivos.

Para alcançar estes objetivos, é possível utilizar uma arquitetura modular para uma fácil integração e modificação dos dispositivos utilizados. As alterações relacionadas a modificar, inserir ou retirar módulos envolvidos, são resultados de eventuais necessidades do usuário que possam surgir com o tempo de uso de uma cadeira de rodas elétrica. Este conceito também está presente nas pesquisas de Gomi e Griffith (1998) e Harvey (1992) sob a forma de interface e robótica evolutiva, respectivamente.

O protocolo de integração proposto por Nelisse (1998) é o M3S (*Multiple Master Multiple Slave*). Este protocolo é reconhecido pelo comitê da ISO como uma interface serial para o controle de cadeiras de rodas e, segundo Nelisse (1998), esforços vêm sendo realizados para a sua padronização. O protocolo M3S utiliza três pares de linhas: um para comunicação serial a 250kbit/s, um segundo para distribuição de energia, e um terceiro para sistema de segurança.

O par de fios associado ao sistema de segurança é composto de um fio para o botão *dead man* (gera evento quando desligado) e um segundo para o comando *key switch* (gera evento quando ligado). Estas duas linhas permitem parar qualquer ação ou desligar o sistema em qualquer circunstância de emergência.

Caso necessário, é possível integrar mais de um dispositivo em um único módulo, de modo a utilizar uma única linha de comunicação. De modo semelhante, com a inserção de módulos adicionais, é também possível utilizar comunicações *wireless* entre dispositivos para facilitar a integração a módulos externos (computadores e barramentos residenciais).

Nesta arquitetura é possível distinguir quatro tipos de componentes:

- *Central de controle e configuração*: componente único e relacionado à configuração de todos os demais componentes e gerenciamento da informação que circula no barramento de dados;
- *Saída*: componentes relacionados a funções de atuação (*drivers* de controle).
- *Intermediário*: componentes que relacionam os sistemas de entrada e saída, (sistemas de evitar obstáculos, navegadores).
- *Entrada*: componentes relacionados à interfaces com o usuário ou sensores (*joystick*, infravermelho, reconhecimento de voz).

A integração entre sistemas, presentes na figura 5.1, é realizada por meio de uma Central de Controle e Configuração, definida como CCM por Nelisse (1998). Quando o sistema é inicializado pela CCM, cada módulo recebe um código (*device number*), que permite a sua identificação. Este processo é chamado de *Serial Number Arbitration Protocol* (SNAP), é possível atribuir o comportamento de cada módulo, ou seja, a forma com a qual ocorrerá a sua interação. Para isto, utiliza-se um software padrão IBM-PC que permite inserir e configurar os módulos pela sua interface gráfica.

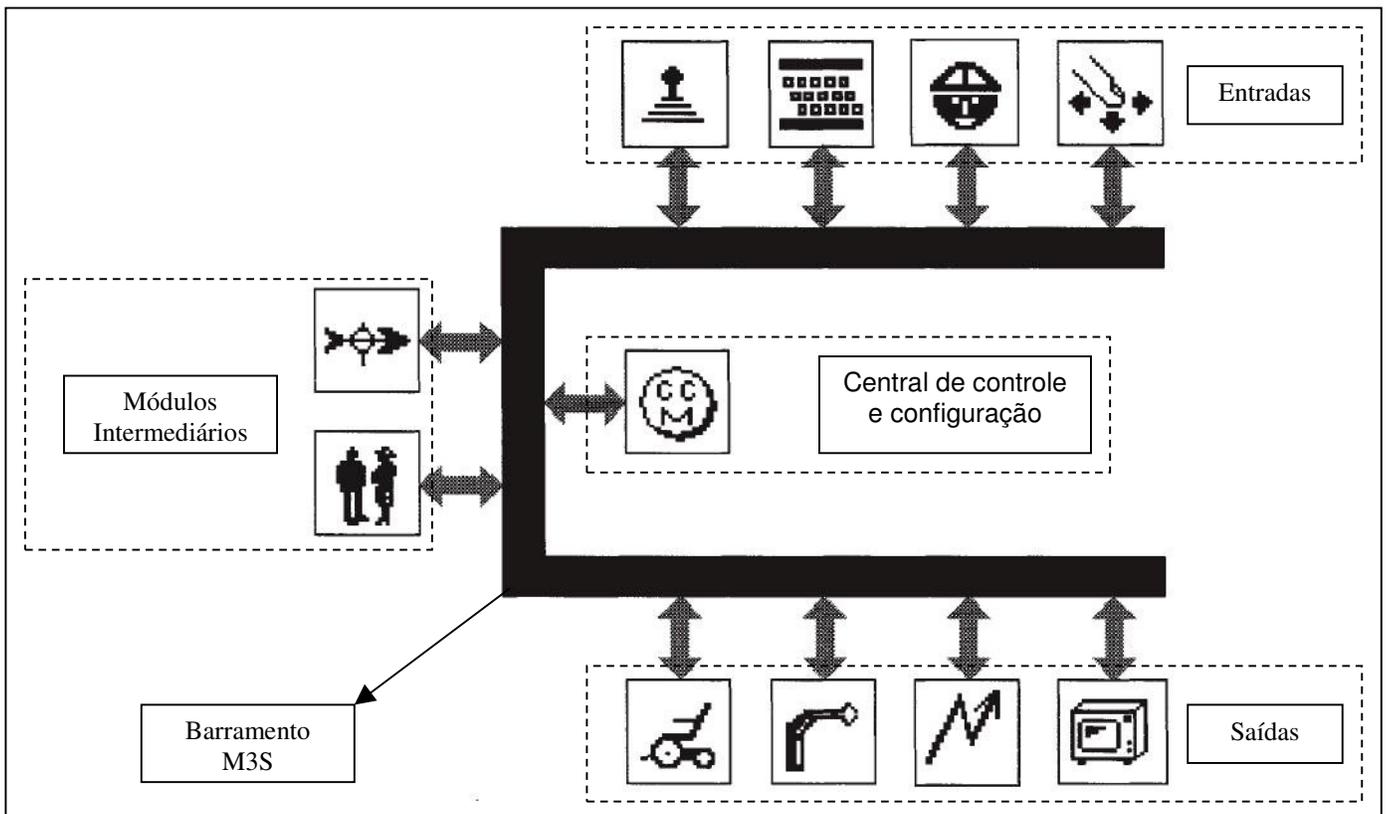


Fig. 5.1 - Arquitetura M3S para protocolo de integração (fonte: Nelisse, 1998).

Deste modo, é possível inserir mais de uma entrada, bem como definir quais serão reconhecidas como dispositivos de segurança. Caso o sistema implementado seja simples, como uma entrada e uma saída, o modo plug and play presente na CCM dispensará esta configuração manual. Os dispositivos inseridos serão detectados pela Central de Controle Configuração e o sistema já estará apto a funcionar.

5.2. Utilização de uma Hierarquia de Entradas

Em situações em que muitos dispositivos de entrada são utilizados, como *joysticks*, sonares, reconhecimento de voz e botões de emergência é necessário utilizar uma lógica para evitar conflitos. Uma opção, semelhante a arquitetura *subsumption*, é o emprego de uma lógica de controle hierárquico que permita uma integração de todos os módulos de interface com o usuário de modo intuitivo e consistente. Esta lógica considera que determinadas entradas possuem predominância sobre outras quando utilizadas, e se mantêm como inoperantes até que um evento ocorra.

É necessário observar que a integração hierárquica não permite que níveis de prioridade baixa gerem ações durante a ocorrência de eventos de prioridade mais alta, o que não ocorre em uma integração *subsumption*. Na integração *subsumption* há a presença de funções supressoras, que permitem a laços de prioridade menor suprimir laços de prioridade maior, por um intervalo de tempo pré-definido.

Como exemplo, um controle hierárquico que combina as entradas de *joystick* e sonar para desvio de obstáculos seguirá os comandos do *joystick* até que um evento de detecção de objeto ocorra. Neste caso, os movimentos em direção ao objeto realizados pelo *joystick* são desabilitados até que o objeto se afaste ou o usuário se afaste do mesmo. Para a construção de uma hierarquia de controle, definem-se as interfaces que apresentam maior robustez e são associadas à segurança em níveis maiores de prioridade e os sistemas menos robustos nos níveis mais baixos, como representado na tabela 5.1:

Tabela 5.1 - Exemplo de hierarquia para interface.

Interface	Tipo de evento	Prioridade
Botão de emergência	Botão pressionado	1 – Alta
Sonar	Obstáculo detectado	2
<i>Joystick</i>	Mover-se para fora da posição zero	3
Controle de voz	Voz gritada	4
Teclado	Novo comando reconhecido	5
Controle de voz	Novo comando reconhecido	6 – Baixa

A confiabilidade e performance deste sistema de controle está diretamente relacionadas aos primeiros níveis da hierarquia: caso o sonar seja muito suscetível a falsos positivos (detectar objetos inexistentes) por situações de referências cruzadas, o sistema encontrará dificuldades ao se movimentar. O mesmo acontece se o *joystick* possuir uma má regulação gerando ruídos indicando a sua saída da posição zero, de modo a se sobrepor às entradas do teclado e controle de voz.

Na tabela 5.1 também é possível observar o controle de voz em dois níveis diferentes da hierarquia. A escolha de comandos de voz como baixa hierarquia (6) deve-se à baixa confiabilidade no reconhecimento de voz, de modo que pode ser necessária a intervenção por sonar ou *joystick* quando ocorre um comando indesejado. Observa-se também a presença do reconhecimento de voz gritada em uma hierarquia mais alta (4), pois este tipo de evento associa uma incerteza menor no seu reconhecimento e relaciona-se a situações de emergência como o comando “Parar!”. Em ambos os casos é possível sobrepor o controle de voz por meio do *joystick* ou sonar, contornando situações de falsos positivos, ou não reconhecimento da interação do usuário.

Desta forma, a integração entre os módulos ocorre ao receber os comandos de uma hierarquia inferior e repassá-los para a hierarquia superior seguinte caso não ocorra nenhum evento na hierarquia presente. Como exemplo, um diagrama representativo de módulo inibidor associado a um sonar que gera eventos ao detectar obstáculos é ilustrado na figura 5.2:

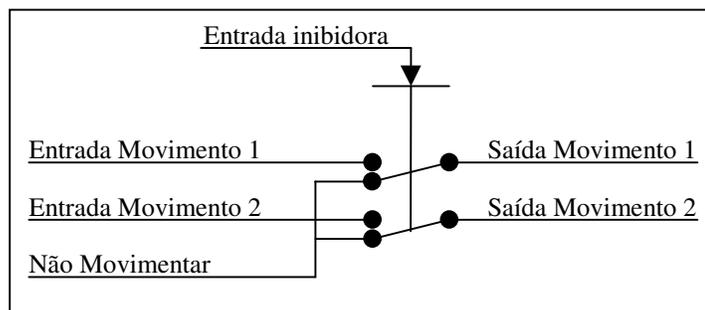


Fig. 5.2 - Representação de evento em módulo para hierarquia de sonar.

Em uma situação normal de operação, os comandos de entradas são transmitidos diretamente à saída. Caso ocorra um evento do sonar, o módulo inibidor sobrepõe seu comando as hierarquias inferiores.

5.3. Abordagem *Subsumption*

Segundo Brooks (1986), soluções tradicionais (cartesianas) para o desenvolvimento de sistemas autônomos de navegação subdividem os módulos necessários verticalmente, como ilustrado na figura 5.3:

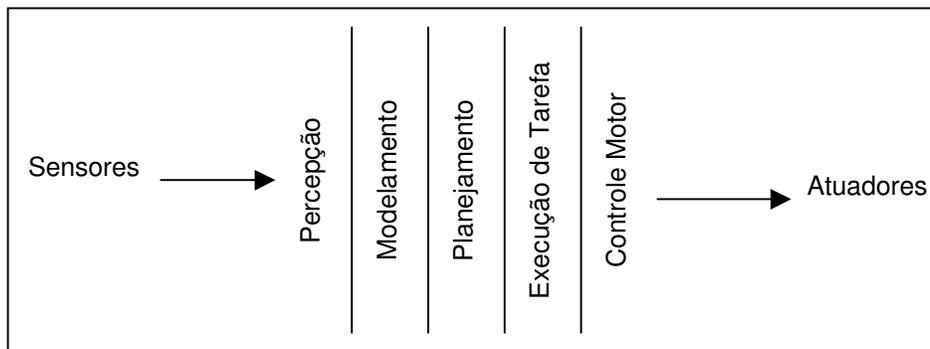


Fig. 5.3 - Abordagem tradicional para sistemas de navegação (fonte: Brooks, 1986).

Deste modo, o fluxo de dados segue um caminho rígido em seu processamento:

1. *Ambiente externo para robô*: por meio da percepção intermediada associada aos sensores;
2. *Atravessando cada subsistema interno*: os dados obtidos pelo sensoreamento atravessam todas as subdivisões, da percepção ao controle motor, como ilustrado na figura 5.3;
3. *Controlando os motores*: após os dados obtidos atravessarem todas as subdivisões do sistema de processamento, obtém-se o controle para os motores e atuadores. Este controle é resultado da percepção do meio externo e das metas definidas para o robô e permite a interação com o meio externo.
4. *Fechando o ciclo*: após um ciclo de percepção, processamento e controle, a interação com o meio externo retorna do ambiente ao robô. Este retorno pode ser uma nova

posição, resultante de um deslocamento; a colisão com um obstáculo, entre outros eventos;

Esta abordagem demanda a presença de uma unidade central, responsável por processar a informação proveniente dos sensores e interfaces com o usuário, e agregando uma maior complexidade ao sistema.

Outra característica, observada por Gomi e Griffith (1998) e Brooks (1986), descreve a alteração das características, ou a substituição de componentes presentes na arquitetura cartesiana (tradicional). Quando componentes são alterados ou substituídos, é necessário que as mudanças decorrentes não interfiram nos módulos próximos associados. Outra forma para manter a funcionalidade, de um sistema cartesiano, é compensar as mudanças das características novas nos componentes associados a estas mudanças.

Observando esta dependência, entre os componentes, presente em uma arquitetura rígida, Brooks (1986) propõe uma nova arquitetura para organizar os módulos envolvidos no sistema de controle de robôs autônomos. Deste modo procurou-se descentralizar o processamento, para permitir que cada módulo seja inserido ou retirado sem a necessidade de reprogramação do sistema pelo usuário.

Para esta descentralização, assume-se que cada módulo possui um comportamento característico e que um conjunto de módulos resulta em uma camada de processamento paralelo associada a uma função (ou nível de competência).

Estas camadas são organizadas horizontalmente por níveis de competência, como ilustrado na figura 5.4:

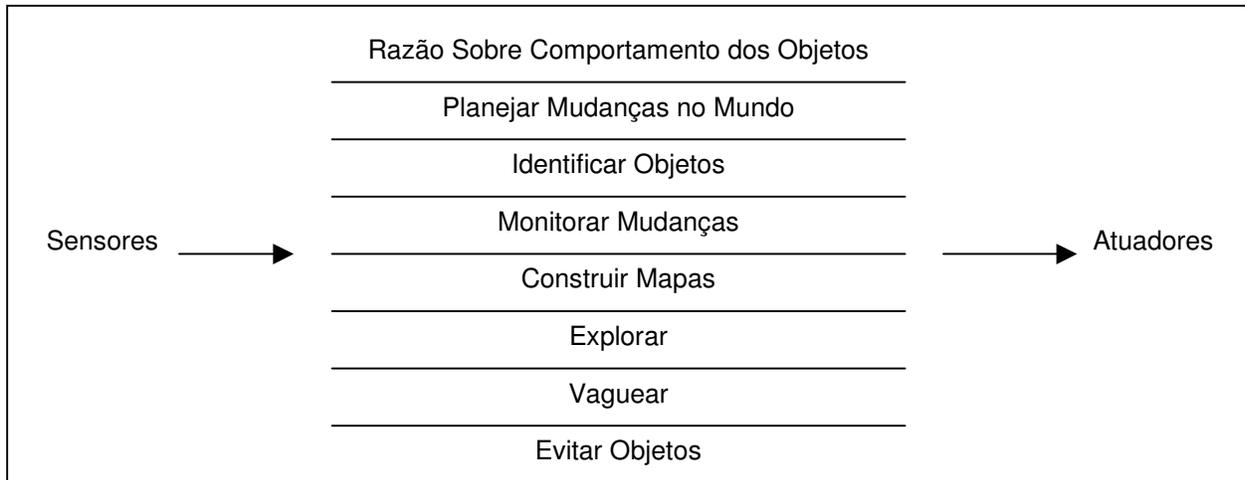


Fig. 5.4 - Abordagem subsumption para sistemas de navegação (fonte: Brooks, 1986).

É interessante observar que a composição de módulos simples permite a execução de tarefas complexas, como demonstrado por Simon (1969) ao afirmar que:

“Comportamentos complexos (e úteis) não são necessariamente um produto de um sistema complexo de controle. Preferivelmente, um comportamento complexo pode ser o simples reflexo de um ambiente complexo.”

Detalhando esta subdivisão horizontal, é possível observar a presença dos módulos comportamentais, como ilustrado na figura 5.5 que demonstra a camada mais baixa e destinada ao controle do robô:

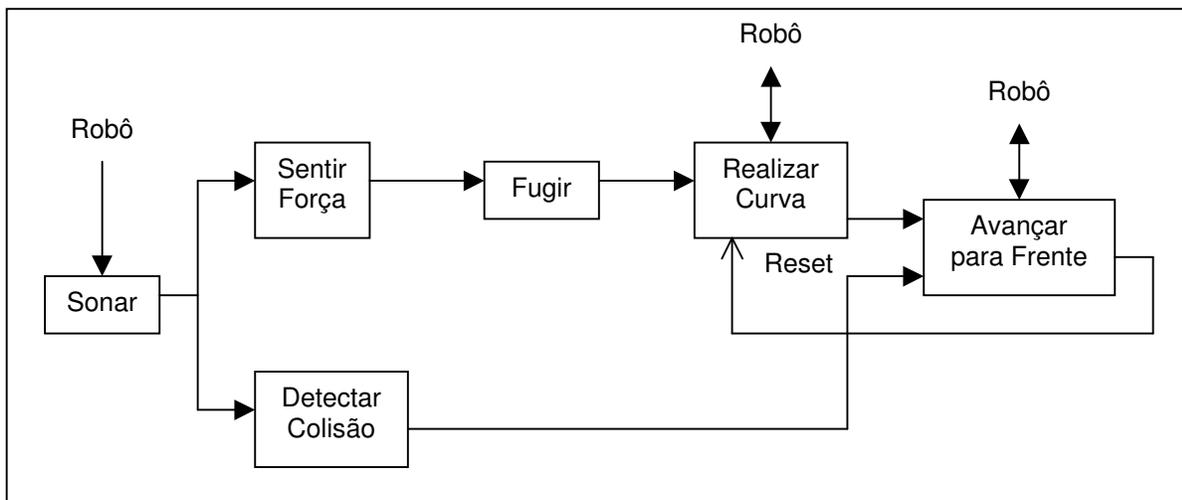


Fig. 5.5 - Camada 0, destinada ao controle (fonte: Brooks, 1986).

A camada 0 exemplificada é composta de um módulo comportamental destinado a evitar colisões (quando obstáculos muito próximos são detectados), e um conjunto de módulos comportamentais para evitar a aproximação de obstáculos. O módulo comportamental “detectar colisões” envia um comando de parar à função “avançar para frente” a cada evento de detecção de obstáculo muito próximo. Este comportamento evita colisões durante a movimentação do robô.

Outro conjunto de módulos monitora obstáculos próximos (“sentir força”), e compara com uma distância segura pré definida (“fugir”). Caso a distância seja insegura, envia-se um comando para “realizar curva” e posteriormente “avançar para frente” (quando a curva for concluída). Deste modo é possível desviar-se de obstáculos móveis e evitar colisões durante a movimentação.

O desenvolvimento e *debug* de uma arquitetura *subsumption* é realizado em ciclos de adicionar, testar, corrigir e adicionar novas camadas a medida em que as camadas (níveis de competência) inferiores apresentam o resultado esperado. Deste modo, no processo de adição de módulos comportamentais ou camadas, há uma hierarquia que permite suprimir ou inibir

informações de módulos ou camadas de hierarquias inferiores por um período de tempo predeterminado. Como exemplo deste sistema adição de módulos e camadas, a figura 5.6 representa um exemplo de adição de uma camada vaguear (1) à camada de controle já existente:

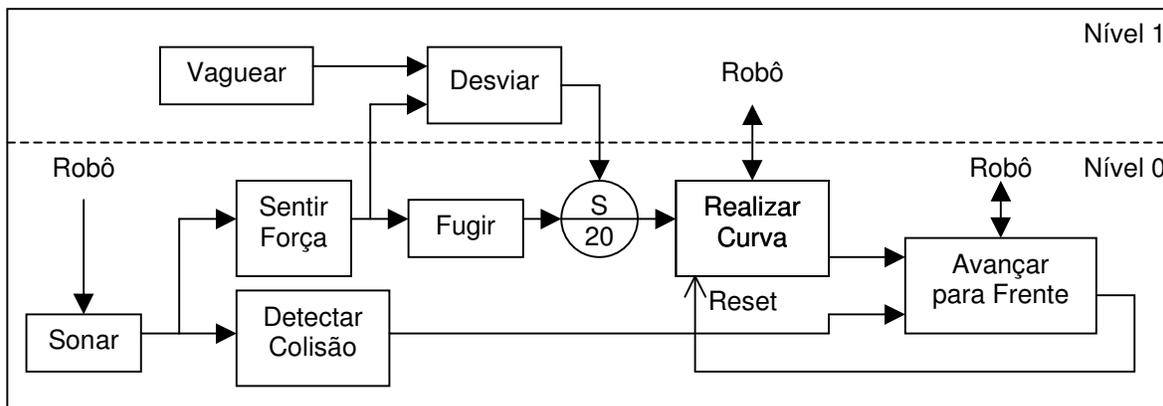


Fig. 5.6 - Integração entre Camadas 0 e 1 (fonte: Brooks, 1986).

Neste exemplo para integração de camadas, é possível observar uma entrada supressora (S/20) para integrar o nível 1 ao nível 0. Esta entrada suprime o comportamento “fugir” por um determinado intervalo de tempo, indicando outra direção para movimentar-se, por meio dos comportamentos “vaguear” (escolhe direção para locomover) e “desviar” (mantém distância de objetos).

Observa-se que, durante o período em que o comportamento fugir é suprimido, o comportamento “detectar colisão” continua habilitado, evitando colisões com objetos próximos. Deste modo, a combinação entre as camadas 0 e 1 permite que o robô navegue sem uma rota definida, ao mesmo tempo em que se desvia de obstáculos.

Este robô possui ainda uma camada 2, dedicada a explorar o local por meio de informação visual, permitindo-lhe planejar rotas de caminho em função da imagem obtida por câmeras. Observa-se que na data de publicação estas câmeras são emuladas por um sensor de sonar dedicado a encontrar corredores.

A integração desta camada 2 às camadas 0 e 1 é demonstrada na figura 5.7:

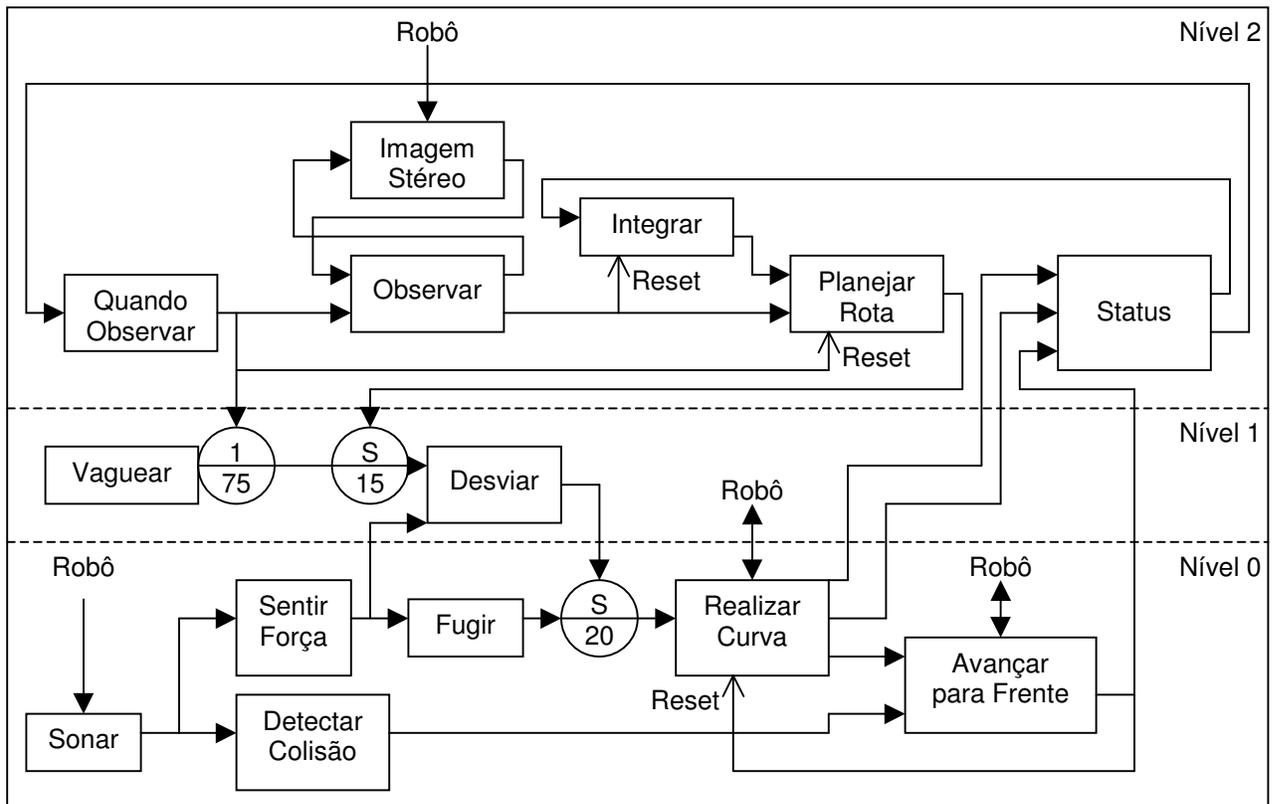


Fig. 5.7 - Integração entre camadas 0, 1 e 2 (fonte: Brooks, 1986).

Assim, o conceito da modularidade presente nas arquiteturas comportamental de Gomi e Griffith (1998), e *subsumption* de Brooks (1986), permitem um processamento descentralizado e paralelo entre os módulos de interface e auxílio a navegação. Reconhecidas as características presentes em arquiteturas *subsumption*, este tópico descreverá outros exemplos de sua implementação.

5.3.1. O robô RAM-1

A tese de dissertação de mestrado de Jacobo (2001), desenvolvida na Faculdade de Engenharia Mecânica da Unicamp e orientada pelo professor Dr. Pablo Siqueira Meirelles, possuiu como objetivo desenvolver e implementar um sistema de controle, em arquitetura *subsumption*, para um robô autônomo microcontrolado. O robô RAM1, ilustrado na figura 5.8, possui dois sensores de colisão e outros dois sensores para presença de solo (evitando a sua queda de um degrau ou mesa, por exemplo).

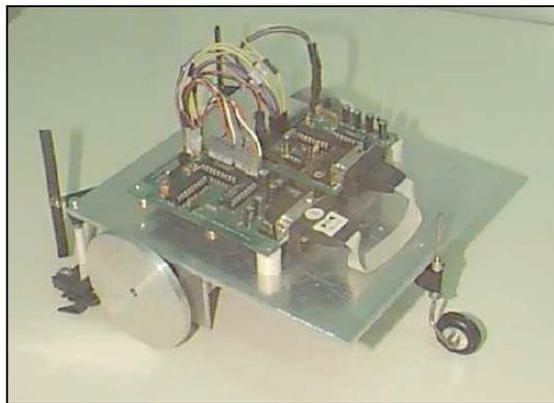


Fig. 5.8 - Robô RAM1 (fonte: Jacobo, 2001).

Como processamento utilizam-se dois microcontroladores PIC 16F84 associados respectivamente aos módulos: Sb Step Control (para controle dos motores a passo), e Sb 9902_2 (para o *firmware* de sensoreamento e controle do robô).

Neste robô implementaram-se as camadas evitar obstáculos (nível 0) e vagar sem direção definida (nível 1). Estas camadas permitiram que o robô RAM1 inicialmente vague, em linha reta, sem uma direção definida até encontrar um obstáculo. Quando encontra-se um obstáculo (colisão ou fim de piso), o comportamento evitar, que se sobrepõe ao comportamento vagar, controla os motores para recuar e realizar uma curva para desviar-se do obstáculo. Concluído o desvio, o comportamento vagar é novamente habilitado, permitindo que este escolha uma nova direção para deslocar-se livremente até a detecção de um novo obstáculo.

Segundo Jácobo (2001), o objetivo proposto no trabalho foi alcançado: implementar a arquitetura *subsumption* na construção de um robô móvel autônomo. Este robô também permite ser utilizado como base de testes para novos componentes como motores, sensores, controladores e baterias. A arquitetura *subsumption* implementada por meio de módulos comportamentais, que são executados por *polling* ao varrer os sensores, demonstrou-se prática e eficiente para a construção de robôs autônomos.

É também concluído por Jácobo (2001) que o robô RAM-1, por ser caracterizado como um sistema mecânico não-holonômico, com restrições cinemáticas não integráveis, não pode ser estabilizado por meio das leis de controle com realimentação de estados.

5.3.2. O Aprendizado de Comportamentos com o Robô Obelix

O projeto de Mahadevan e Connell (1991) observa que há dificuldades em programar os módulos comportamentais em uma arquitetura *subsumption*. Desta forma, propõe-se que novos comportamentos podem ser aprendidos pelo robô, poupando o esforço do programador ao evitar uma análise profunda nas relações entre o robô e o meio no qual se realiza a tarefa.

Para a avaliação deste aprendizado realizam-se experimentos na forma de aprendizado reforçado. A técnica de aprendizado reforçado consiste em realizar experimentos e utilizar sensores para avaliar situações de sucesso (ou insucesso), atribuindo valores a um índice presente em cada módulo de comportamento. Desta forma, construiu-se o robô Obelix para testar a capacidade de aprendizado da arquitetura *subsumption* em uma tarefa de transportar caixas.

Para isto, este robô possui 8 sonares, um sensor infravermelho e um sensor de excesso de corrente nos motores (quando o robô está preso por um obstáculo e força avançar). A leitura dos sensores é realizada de modo discreto: os sensores de sonar possuem dois estados, próximo e longe, resultando em 16 bits de informação. Considerando que os outros dois sensores associam mais um bit de dados cada um, totalizam-se 18 bits de informação ao somar-se todos os sensores.

Na coordenação e execução da tarefa, utilizou-se uma arquitetura subsumption organizada em três camadas: “encontrar a caixa”, “empurrar a caixa” e “não forçar movimentação quando preso”, como ilustrado na figura 5.9:



Fig. 5.9 - Módulos do robô Obelix (Fonte: Adaptado de Mahadevan e Connell, 1991).

Assim, quando o robô é ligado, as camadas 1 e 0 não geram eventos, iniciando a camada 2. “Encontrar Caixa” inicia-se movimentando o robô para encontrar objetos próximos. Esta camada procura caixas por meio dos bits correspondentes a informação “obstáculos próximos do sonar”.

Quando um objeto à frente é encontrado, este é testado para ser definido como caixa ou não. Para isto é realizada uma comparação entre os bits referentes aos 8 sonares, avaliando a forma do objeto encontrado. Caso a forma seja adequada, recebe-se um *feedback* F positivo e a primeira camada, “empurrar caixa”, é habilitada. Esta primeira camada procura manter a caixa à frente do robô, a medida em que este a empurra. Durante este processo, caso a caixa saia da frente do robô Obelix, ocorre um *feedback* P negativo. A medida em que desloca-se a caixa, mantendo-a corretamente em frente do robô recebem-se *feedback* P positivos.

Durante o processo de procura por caixas, obstáculos podem ser encontrados, impossibilitando a movimentação da caixa. Nestas situações a camada 0 deve realizar um desvio de modo a permitir que contorne-se o obstáculo. A cada evento de contorno de obstáculo bem sucedido recebe-se um *feedback* U positivo, e caso não seja possível contornar receber-se *feedback* U negativo.

Estes *feedbacks* são avaliados estatisticamente e permitem otimizar as camadas comportamentais, resultando em uma maior performance na execução da tarefa desejada. Deste modo, Mahadevan e Connell (1991) comparam os tempos gastos para o aprendizado com uma abordagem monolítica com os obtidos na abordagem comportamental.

A abordagem monolítica, utilizada antes da abordagem comportamental descrita, utiliza apenas um módulo para gerenciar todas as subtarefas envolvidas, de modo a estar ativa durante todo o processo de controle. Esta abordagem monolítica foi substituída pela comportamental. Deste modo os testes realizados, em ambas as abordagens, demonstraram um tempo de aprendizado para a abordagem *subsumption* equivalente a metade do tempo necessário em uma abordagem monolítica.

5.3.3. Aprendizado por meio de *Subsumption*

A pesquisa de Maes e Brooks (1990) demonstra uma arquitetura *subsumption* capaz de escolher, por meio de aprendizado, quais comportamentos são mais adequados a uma determinada situação. Nesta abordagem utilizaram-se 12 comportamentos predefinidos para o um robô com dois graus de liberdade em cada uma das suas de 6 pernas, totalizando 12 graus de liberdade, como ilustrado na figura 5.10.

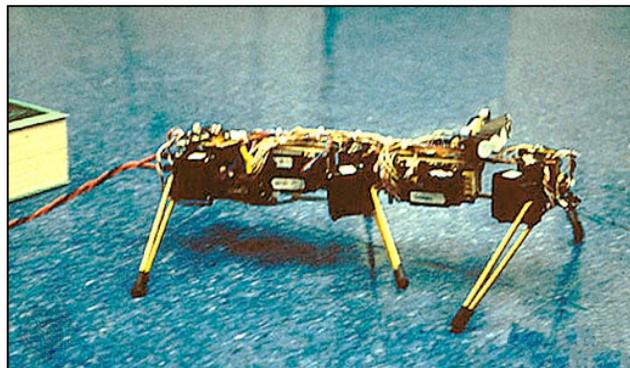


Fig. 5.10 - Robô Genghis, MIT (fonte: Britannica Concise, 2008).

Estes 12 comportamentos são todos atribuídos para o controle de cada uma das seis pernas:

- *Avançar robô*: mover para trás, subir, para frente e descer a perna;
- *Recuar robô*: mover para frente, subir, para trás e descer a perna;

Para o treinamento deste robô atribuíram-se duas variáveis, cujos valores são resultados de tentativas bem ou mal sucedidas (*feedback* positivo ou negativo, respectivamente) em relação a cada um dos 12 comportamentos. A geração de *feedback* negativo utiliza dois sensores de contato, dispostos na base do robô e acionados quando há contato entre a base do robô e o solo. Para a geração de *feedback* positivo avalia-se o movimento de avanço na posição desejada, por meio de um encoder associado a um fio preso ao robô. Estes dois sensores são representados na figura 5.11:

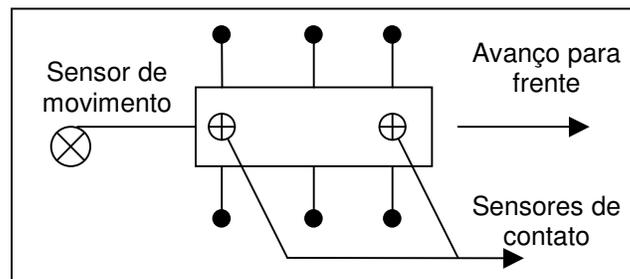


Fig. 5.11 - Sensores de feedback para o robô Genghis (fonte: Maes e Brooks, 1990).

Considerando que os módulos comportamentais alternam-se durante o processo de aprendizagem, é necessário utilizar índices para determinar a sua frequência de uso e necessidade de otimização. Desta forma, associa-se um índice para relevância e outro para confiabilidade em cada módulo comportamental.

O índice de relevância associa-se ao número de vezes em que foi possível obter um *feedback* positivo em uma situação de módulo comportamental ligado. Deste modo, índices de relevância maiores associam-se a módulos comportamentais que apresentam maior ocorrência de *feedback* positivo, quando em operação. Este índice de relevância resulta nas chances de um determinado módulo ser selecionado como ativo.

O segundo índice utilizado para seleção de módulos no processo de aprendizagem é o índice de confiabilidade, que relaciona a chance de ocorrer um *feedback* positivo quando um

determinado módulo é utilizado. Considerando que este índice relaciona probabilidade seu valor é de -1 a 1, incluindo valores fracionários. Este índice de confiabilidade resulta nas chances de um determinado módulo ser otimizado ou não.

Desta forma, realizaram-se dois experimentos para avaliar a capacidade de aprendizado: o primeiro utilizando 6 módulos comportamentais e o segundo utilizando 12 módulos comportamentais.

Os primeiros experimentos utilizaram apenas os comportamentos “avançar robô”, que não são capazes de gerar um *feedback* negativo de forma isolada, mas permitem obter *feedback* positivo quando o robô desloca-se para frente. Como resultado do aprendizado foi possível observar um deslocamento na forma de tripés: o robô mantém sempre três pontos de apoio, demonstrando um deslocamento similar ao apresentado por insetos.

Procurando aumentar o grau de dificuldade, Maes e Brooks (1990) habilitaram também os módulos comportamentais “recuar robô”. Estes comportamentos, desde que em movimento coordenado, não geram *feedback* negativo, contudo também não geram *feedback* positivo. Desta forma, o aprendizado deve inibir o uso de funções que não agregam *feedback* positivo, por meio do índice de relevância que permite utilizar um número maior de vezes os comportamentos bem sucedidos

Desta forma, observa-se que a técnica de integração entre camadas *subsumption* por supressão em tempos constantes, proposta inicialmente por Brooks (1986), difere-se do método de aprendizado, apresentado por Maes e Brooks (1990), no qual é possível utilizar intervalos de tempo que se adaptam a situação em função dos *feedbacks* obtidos. Esta adaptação, por *feedback* externo, permite inclusive atribuir períodos que tendem a suprimir módulos não favoráveis a uma determinada tarefa.

Capítulo 6

Construção e Implementação de Protótipos

A revisão bibliográfica, apresentada nos capítulos anteriores, sustentou a construção e implementação de protótipos para interfaces e sistemas de auxílio a usuários de cadeiras de rodas motorizadas. Deste modo, inicialmente desenvolveram-se duas arquiteturas de interface para reconhecimento de voz e uma técnica de filtragem para *joystick*.

Posteriormente ao desenvolvimento destes protótipos, em trabalho conjunto com o pesquisador Torsten Felzer, que é desenvolvedor do projeto HaWCoS, foi possível adaptar e implementar esta técnica de interface em protótipo utilizando um microcontrolador PIC 16F877A.

O protótipo desenvolvido com o conceito HaWCoS pôde ser implementado na forma de dois protótipos: inicialmente em um pequeno veículo, e posteriormente em um kit de motorização para cadeira de rodas. Para uma implementação consistente da interface desenvolvida com o conceito HaWCoS, para o kit de motorização, definiu-se um protocolo para modularizar os componentes envolvidos.

Esta modularização é resultado de uma padronização do protocolo de comunicação entre sistemas envolvidos, possibilitando a utilização de interfaces diferentes sem alterações nos *drivers* dos motores ou em qualquer outro módulo. Observa-se que, neste capítulo, a seqüência de apresentação dos protótipos é organizada cronologicamente. Deste modo, o primeiro protótipo

desenvolvido é o que realiza reconhecimento de voz para a interface com o usuário. Assim, a técnica de integração entre interfaces e sistemas de auxílio a navegação apresentada segue a proposta de Yanco (1998b) na qual a padronização entre os módulos envolvidos ocorre posteriormente ao desenvolvimento dos módulos.

6.1. Reconhecimento de Voz

Para situações de grave limitação motora, como a tetraplegia, é necessário o emprego de uma interface que seja sensível às demais formas de interação que o usuário possui, como voz, posição dos olhos ou movimentos dos ombros. A utilização de reconhecimento de voz e sons permite manter consistência com comandos utilizados no cotidiano, como “frente” e “trás”. De forma semelhante a pesquisa de Bilmes *et al.* (2006), o reconhecimento de voz também deve reconhecer sons emitidos pelo usuário como “ans” e “uns”, caso este não possa se comunicar.

Seguindo os passos de Preece *et al.* (2002) para o desenvolvimento de interfaces, a primeira fase teve como objetivo delimitar o escopo do projeto: desenvolver uma interface simples e intuitiva para o controle de cadeiras de rodas por voz. Este escopo permitiu definir que, para o controle da cadeira de rodas, o usuário deverá utilizar poucos comandos, favorecendo a simplicidade e o seu aprendizado. Deste modo definiram-se cinco comandos necessários: frente, trás, esquerda, direita e parar, que possam ser executados por meio de palavras ou sons emitidos pelo usuário.

Reconhecidos o escopo e as necessidades do usuário, desenvolveu-se um protótipo funcional. Este protótipo permitiu realizar simulações de controle e observar aspectos desta interface, como a transparência entre a função solicitada e o comando realizado, e estudar novas aplicações além do controle de cadeiras de rodas.

O protótipo inicial consistiu em um conjunto formado por três elementos: as interfaces de entrada, os sistemas de controle e sistemas controlados como representado pelo diagrama da figura 6.1:

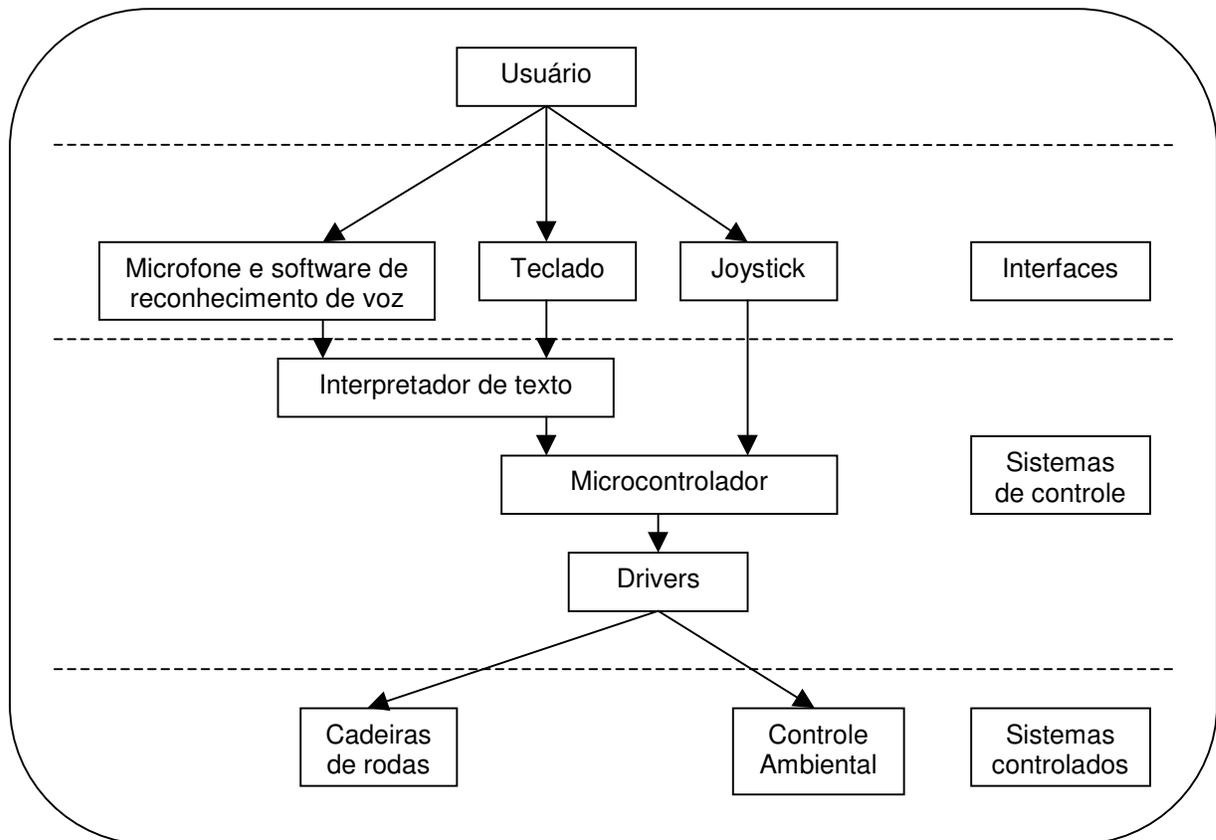


Fig. 6.1 - Arquitetura do primeiro protótipo de reconhecimento de voz.

* Controle Ambiental: controle de portas, lâmpadas, tomadas, ar condicionado, etc.

As interfaces de entrada são: a vocalização e o reconhecimento de palavras por software dedicado (IBM ViaVoice®), a digitação da palavra em teclado, ou o controle direto por um joystick. Reconhecendo que este primeiro protótipo foi implementado em um computador com um software para reconhecimento de voz, foi possível reconhecer e armazenar uma maior diversidade de palavras que no segundo protótipo.

Por meio do software de reconhecimento de voz observou-se inclusive uma maior facilidade em reconhecer palavras, já armazenadas na biblioteca do software; se comparado a sons, pois estes devem ser treinados para posterior reconhecimento. Deste modo, foi possível

uma relação direta entre comando e palavra vocalizada: a palavra frente vocalizada corresponde a uma movimentação também para frente.

Os comandos utilizados nesta sintaxe assemelham-se a forma com a qual o usuário conversaria com uma pessoa para encaminhá-la à uma posição desejada. Deste modo é possível manter a consistência de comandos e favorecer o aprendizado. Assim, os comandos podem ser digitados pelo usuário, ou reconhecimento de voz, e adquiridos por um software de interpretação de texto em tempo real.

Este software de interpretação de texto foi desenvolvido neste projeto e possui uma biblioteca de comandos gerada em tempo de programação. Nesta biblioteca é possível relacionar palavras a comandos, que são transmitidos por meio de uma interface serial (RS232) para um microcontrolador externo. O microcontrolador externo, embarcado em um protótipo de robô, é o responsável pelo controle dos *drivers* e pela recepção direta dos comandos do *joystick*, que possui uma prioridade maior que os comandos textuais.

A prototipagem da interpretação de textos permitiu encontrar novas aplicações, além do controle de cadeiras de rodas, como ligar ou desligar lâmpadas ou tomadas de uma residência por meio de uma central de reconhecimento de comandos. Contudo este primeiro protótipo de reconhecimento de voz apresentou como desvantagens a necessidade de utilizar um computador dedicado para o reconhecimento de voz e digitação.

A necessidade de um computador e software para reconhecimento de voz torna este protótipo impróprio a aplicações embarcadas, nas quais a mobilidade e consumo de energia são fatores fundamentais. Outra dificuldade encontrada foi a necessidade em adquirir um software de reconhecimento de voz (aproximadamente R\$ 350) e um computador laptop (aproximadamente R\$1700), pois ambos encarecem significativamente o custo do projeto. Este aumento de custo; pode tornar esta interface inacessível a uma grande parte dos seus possíveis usuários.

A solução encontrada foi desenvolver um conjunto de *hardware* e *firmware* para um microcontrolador que incorpore os elementos presentes no computador, de forma a eliminar este último, permitindo maior mobilidade e significativa redução de custos na interface final. Por

meio deste conceito, o segundo protótipo incorporou as funções de reconhecimento de voz e interpretador de texto no *firmware* do microcontrolador. Desenvolveu-se também um hardware para amplificar o sinal de microfone para uma faixa de tensão que possa ser reconhecida. A arquitetura deste segundo protótipo é demonstrada na figura 6.2, na qual o joystick é considerado opcional e representado por uma linha pontilhada:

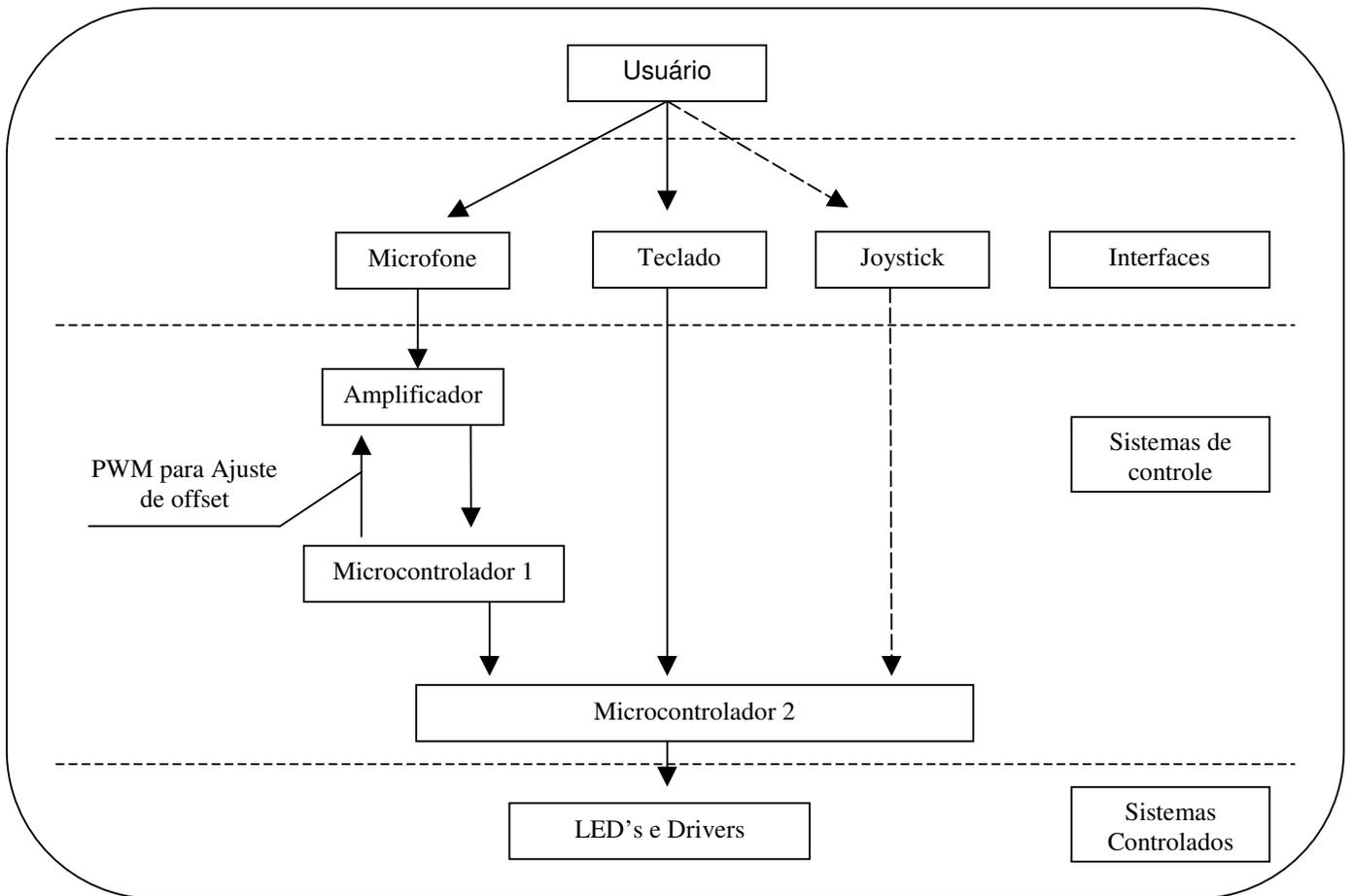


Fig. 6.2 - Arquitetura do segundo protótipo de reconhecimento de voz

É necessário observar que esta arquitetura utiliza dois microcontroladores e um amplificador:

- *Amplificador*: Recebe o sinal do microfone e o amplifica para o primeiro microcontrolador. Possui um ganho constante ajustado por potenciômetro e, quando o sistema é ligado, pode ser necessário um ajuste de *offset* do sinal de entrada por meio do microcontrolador 1;
- *Microcontrolador 1*: Recebe o som amplificado do microfone na forma de interrupções de bordas de subida (frequência), e conversão analógica para digital (10 bits de resolução). Retorna o ajuste para *offset* para o amplificador, e o comando associado ao som vocalizado;
- *Microcontrolador 2*: Recebe as informações das três entradas e, por meio da prioridade de cada uma, controla os LED's e drivers.

Desta forma, a amostragem e interpretação de palavras ou sons emitidos pelo usuário é realizada apenas pelo microcontrolador 1, que repassa os comandos associados para o microcontrolador 2.

Para o desenvolvimento do algoritmo de reconhecimento de voz, inicialmente vocalizaram-se cinco fonemas em um microfone padrão IBM-PC. Estes fonemas foram amplificados e apresentados em um osciloscópio, permitindo observar que cada fonema possui uma forma de onda própria e periódica. Esta amostragem, contudo, não se demonstrou adequada pois o osciloscópio utilizado não permite gravar o sinal obtido para posterior comparação. Reconhecendo esta necessidade, um segundo experimento utilizou-se de uma software para edição de áudio, goldwave v3.24, que permitiu gravar cinco fonemas distintos, em intervalos de tempo de 100ms, para demonstrar a sua forma de onda.

A gravação e comparação entre os fonemas a, e, i, o, u ilustrada na figura 6.3 demonstrou que todos apresentam uma componente periódica, de aproximadamente 10ms. Utilizou-se como parâmetros de aquisição uma entrada mono em 16bits de resolução e aquisição a 48KHz para amostrar intervalos de 100ms plotados em um gráfico com escala vertical de 5 ms por divisão, a escala vertical foi mantida para todos os gráficos, contudo o software utilizado não lhe atribui nenhuma unidade.



Fig. 6.3 – Comparação entre fonemas ‘a’, ‘e’, ‘i’, ‘o’, ‘u’, respectivamente de cima para baixo.

É possível observar que a componente periódica possui uma forma de onda associada a cada fonema, de modo a apresentar um número maior de bordas de subida para o fonema ‘a’ (aproximadamente 4 bordas), se comparado ao fonema ‘i’ (2 bordas), como ilustrado na figura 6.4.

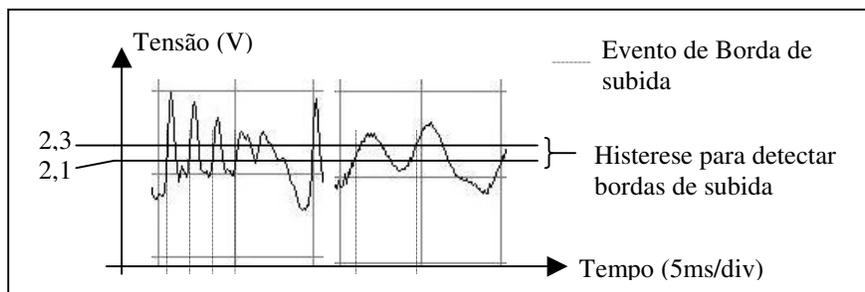


Fig. 6.4 – Aquisição das bordas de subida, para os fonemas ‘a’ e ‘i’.

O reconhecimento de que cada fonema possui um número característico de bordas de subida, permitiu desenvolver um algoritmo que amostra o número de bordas de subida para cada intervalo de 100ms (amostragens de 10ms representam apenas um ciclo) e obtém uma frequência instantânea.

Para uma amostragem adequada do número de bordas de subida, antes de adquirir novas palavras, o usuário deve solicitar ao microcontrolador um ajuste de offset no amplificador em local fechado com o mínimo de fontes sonoras possível. Este ajuste de offset permite que a posição central do sinal (*offset*) esteja em torno de 2,2 volts, pois a borda de subida utilizada é sensível a uma elevação de tensão de 2,1v para 2,3 volts, considerando uma histerese nesta posição central. Para o ajuste de *offset* o microcontrolador utiliza-se da entrada analógica, que recebe o som amplificado, retira uma média entre as amostras e, deste modo, é possível aumentar ou reduzir o período de trabalho da saída PWM.

A saída do PWM, por sua vez, é convertida para um sinal de tensão contínuo, de 0 a 5v e aplicado à entrada inversora de um amplificador operacional. Desta forma, mantendo o ganho de amplificação, o aumento do ciclo de trabalho da saída PWM reduzirá a tensão na saída do amplificador. Assim caso não haja fonte sonora, e a média obtida na tensão de saída do amplificador esteja acima de 2,3 volts, o microcontrolador ajustará automaticamente um ciclo de trabalho maior na saída PWM para ajustar o *offset*.

Após a fase de ajuste do PWM, inicia-se uma fase de “treinamento”. Nesta fase, cada comando (som ou palavra) deve ser vocalizado dez vezes para se obter uma representação única para o mesmo (obtida pela sua curva de frequências). Esta representação em curva de frequências permite associar cada comando a uma função de movimento (frente, trás, esquerda, direita ou parar), permitindo interpretar cada palavra ou som vocalizado e responder com um comando.

Considerando que cada palavra vocalizada, ou som emitido pelo usuário, possui uma seqüência única de frequências instantâneas, o *firmware* possui uma pequena biblioteca de palavras. Esta biblioteca de palavras permite analisar a palavra vocalizada pela sua curva de frequências ao avaliar o número de bordas de subida, representadas por uma tensão de 2,1 volts para 2,3 volts, em 8 intervalos de 100 ms cada.

A biblioteca que relaciona as curvas de frequência de cada palavra ou som vocalizado foi levantada pela média das suas curvas de frequências. Desta forma, para cada palavra ou som novo é realizada uma amostragem na fase de treinamento de 10 curvas das quais obtém-se uma curva média, como ilustrado na figura 6.5:

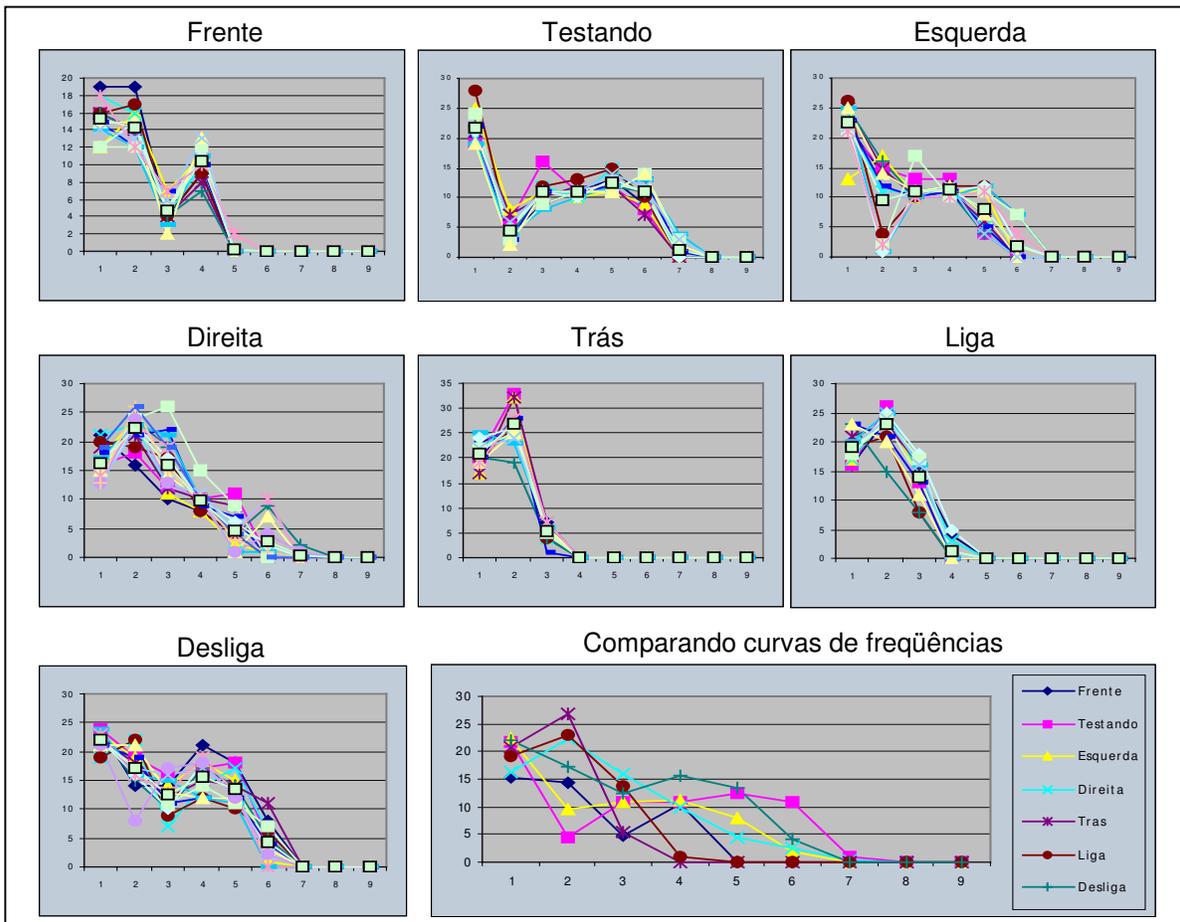


Fig. 6.5 - Curvas de frequência para diferentes palavras.

Legenda:

- Eixo Y: Bordas de subida, Eixo X: número da amostragem
- Linha indicada por quadrados: Média

O levantamento destas curvas amostradas permitiu gerar uma curva de frequências média para cada palavra ou som vocalizado. No sistema implementado utilizaram-se as sete curvas de frequência ilustradas na figura 6.3 para constituir uma biblioteca de palavras adequada ao usuário. A cada início de vocalização, representado pela segunda borda de subida do sinal amplificado, inicia-se uma amostragem em tempo real da curva de frequências vocalizada pelo usuário. Esta amostragem é armazenada para posterior comparação com a biblioteca de curvas de frequências já conhecidas.

A comparação da curva amostrada é realizada somando-se, em módulo, as diferenças de sua frequência com os respectivos intervalos de tempo de cada uma das curvas de frequência da biblioteca. Como exemplo, a figura 6.6 ilustra uma situação de comparação em que a amostra apresenta casamento com a palavra “frente”.

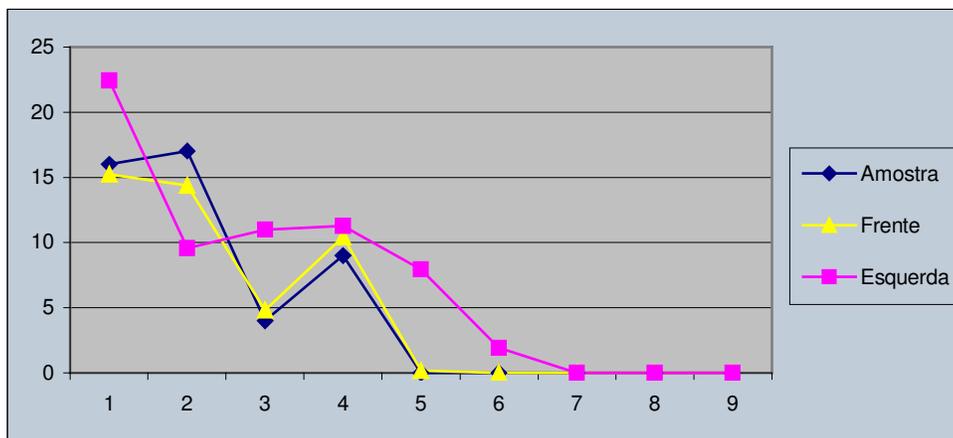


Fig. 6.6 - Comparando uma amostragem com duas curvas de frequências da biblioteca.

As aquisições realizadas em intervalos de 100 milissegundos para a obtenção das curvas de frequência demonstram que, para uma mesma palavra, observam-se desvios inferiores a 15 bordas de subida entre as curvas vocalizadas e a curva média de frequências que a representa, como demonstrado nas tabelas do apêndice II. Em outras palavras é possível definir como casamento quando a palavra vocalizada acompanha a curva média (representação da palavra) em desvios inferiores a 15 unidades. É necessário observar que, para manter uma consistência entre as leituras, após o levantamento da média para as curvas frequências, os valores de amplificação e de período de trabalho do PWM não devem ser alterados.

Por meio da comparação destes desvios instantâneos de frequência é possível obter o desvio entre a curva de frequência amostrada e as curvas de frequência da biblioteca, resultando em três situações:

- *Desvio maior que 15*: Os testes realizados demonstram que apenas desvios iguais ou menores que 15 correspondem a um casamento com uma palavra da biblioteca. Caso este desvio seja maior que 15, a palavra será descartada.
- *Desvio igual ou menor que 15 em uma palavra da biblioteca apenas*: representa um casamento com uma palavra e esta é assumida como o comando desejado.
- *Desvio igual ou menor que 15 em mais de uma palavra da biblioteca*: neste caso o comando será associado a palavra com menor desvio.

O algoritmo utilizado apresenta como vantagem uma simplicidade que o permite ser programado em um microcontrolador (PIC16F877), que possui características muito limitadas em termos de memória e capacidade de processamento, se comparado a um microcomputador. Como exemplo a tabela 6.1 ilustra um microcomputador pessoal de porte médio comparado com um microcontrolador PIC 16F877 de 40 pinos.

Tabela 6.1 - Comparação entre recursos de microcontrolador e computador (fontes: Microchip, 2007; e software CPU-Z, 2007).

Característica	Microcontrolador PIC16F877	Microcomputador pessoal*
Memória de programa	8K Linhas de código	100 Gbytes
RAM	368 Bytes	516M Bytes
Frequência do processador	20 MHz	2000 MHz
Barramento do processador	8 bits	32 ou 64 bits

* Valores aproximados para uma configuração de médio porte

Esta nova arquitetura para reconhecimento de voz requer o uso do microcomputador em apenas duas situações:

- Durante a fase da gravação do programação do microcontrolador 1, que ocorre uma única vez, durante a construção da interface;
- Ao inserir ou alterar uma curva de frequência associada a um comando em EEPROM, que dispensa a programação do software presente no microcontrolador 1. Neste caso o computador opera como interface para a troca de dados na porta serial, permitindo o uso de qualquer software capaz de realizar comunicação pela porta serial.

A utilização do computador externo apenas em fases em que não é necessário o controle da cadeira de rodas permitiu considerar esta Segunda arquitetura como embarcada, associando uma significativa redução de custos e maior mobilidade.

6.2. Filtragem de Joystick por Fuzzy

Usuários com doenças como a esclerose múltipla e a LER (lesão por esforço repetitivo), podem apresentar limitações nos membros superiores como mãos trêmulas ou movimentos involuntários, como descrito por Keates *et al.* (1999) na figura 2.4, que representa a resposta de um usuário com contrações involuntárias na mão. Estas limitações podem incapacitar o uso de interfaces contínuas, como *joysticks* ou o movimento do mouse de um computador. Caso um usuário que apresente este distúrbio utilize uma cadeira de rodas elétrica convencional, controlada por *joystick*, esta se comportará de forma indesejada, ainda que se acompanhe em tempo real os movimentos solicitados pelo usuário.

A abordagem inicial considerou uma implementação direta de lógica fuzzy, recebendo os valores do joystick e interpretando-se em cinco situações de posição para cada eixo: 0-10%, 20-30%, 40-60%, 70-80%, 90-100%, como ilustrado na figura 6.7.

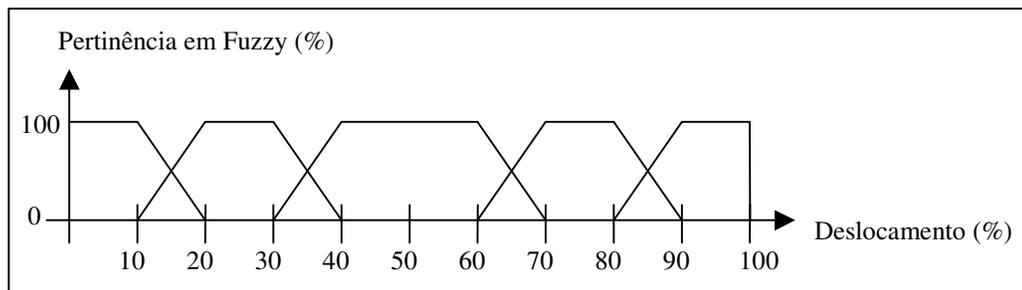


Fig. 6.7 – Aplicação direta de lógica fuzzy para um eixo do joystick.

Contudo, as simulações realizadas em Labview® demonstram que, para fontes de ruídos superiores a amplitudes de 20%, a saída em fuzzy é incerta a respeito de qual é a posição desejada pelo usuário. Esta incerteza ocorre como resultado de uma saída fuzzy que acompanha em tempo real a entrada do usuário: é necessário avaliar mais pontos em um determinado intervalo de tempo por meio de uma amostragem de valores em função do tempo. Desta forma, propõe-se um algoritmo de filtragem por fuzzy em dois estágios: atribuição grau de pertinência e lógica crisp.

O primeiro estágio parte do princípio que a mão trêmula oscila em torno da posição desejada, observado também por Keates *et al.* (1999), formulou-se a hipótese de que os movimentos indesejados são mais rápidos se comparados aos movimentos que definem o comando desejado. Em outras palavras, os movimentos indesejados situam-se em faixas de frequência mais elevadas que o deslocamento para a posição desejada. Por este motivo, o algoritmo inicia com um filtro do tipo “passa-baixa” digital, obtido pela média dos valores amostrados diretamente no *joystick*. Observa-se que esta opção de filtro também é sugerida por Keates *et al.* (1999).

É necessário observar que, a medida em que o número de amostragens aumenta, permite-se uma maior dispersão de valores na entrada e menores variações na saída, relacionando-se a graus de pertinência menores. Assim, como demonstrado na equação 6.1, determina-se que o grau de

pertinência da filtragem fuzzy é o inverso do número de amostras: para uma amostragem por período de tempo ($A = 1$), a saída será igual a entrada (Pertinência = 100%). De forma semelhante, para uma amostragem de 10 valores ($A = 10$), a entrada permitirá valores com maior dispersão agrupados em torno de sua média na saída (pertinência = 10%).

$$Pertinência = 100 \times \frac{1}{A} \quad \text{Eq. 6.1}$$

A utilização do grau de pertinência fuzzy com a filtragem passa baixa digital permite utilizar um algoritmo suficientemente simples, e posterior emprego em microcontroladores destinados a eletrônica embarcada. Deste modo, o filtro passa baixa elimina os picos e vales que se dispersam da posição desejada pelo usuário, permitindo obter um sinal com menores dispersões da posição desejada.

Para o emprego deste filtro passa baixa, ajustado pela pertinência fuzzy, deve ser observada a relação entre oscilação tolerável e tempo de resposta, semelhante a abordagem de Adachi *et al.* (1998) para a obtenção do posicionamento da cabeça. Caso seja utilizada uma grande amostragem por intervalo de tempo, o sinal possuirá menores variações em torno da posição desejada. Ou seja, uma estabilidade maior ao custo de um período de resposta maior. A utilização de maiores amostragens por intervalo de tempo pode ser empregada caso o usuário não necessite de uma resposta rápida ou possua maiores dificuldades em manter-se na posição desejada.

A recíproca também é verdadeira: caso haja uma pequena amostragem por intervalo de tempo (grau de pertinência alto), este sinal de saída apresentará um tempo de resposta menor ao custo de maiores oscilações em torno da posição desejada.

Em testes realizados, utilizando apenas este filtro passa-baixa, observou-se que as oscilações em torno do ponto desejado não podem ser eliminadas por completo. Deste modo, no segundo estágio utilizou-se lógica crisp na forma de uma *deadzone* com uma posição central móvel e amplitude ajustável. Um procedimento semelhante também é empregado em *joysticks*

para jogos, contudo não variando a posição central, mas permitindo apenas um ajuste de amplitude.

No procedimento empregado, a posição central desta *deadzone* altera-se a cada instante em que o sinal de saída do filtro passa-baixa ultrapassa os extremos de sua amplitude, comportando-se como uma histerese. Este acompanhamento da média, que é obtida pelo filtro passa baixa, por meio do valor intermediário entre os limites superior e inferior permite obter a posição desejada pelo usuário.

Para a calibrar a amplitude desta *deadzone* deve ser observada a relação entre oscilação aceitável e sensibilidade: grandes amplitudes para a *deadzone* reduzem praticamente toda a oscilação ao custo de uma insensibilidade quase total aos movimentos do usuário. De forma recíproca, amplitudes pequenas permitem uma grande sensibilidade, mas não são capazes de filtrar as oscilações não eliminadas por meio do filtro passa baixa.

Esta combinação de filtro passa baixa e lógica crisp foi simulada em Labview® (ver apêndice III) demonstrando ser confiável quando os valores de número de amostragens (para o filtro passa-baixa), e de amplitude para *deadzone* são ajustados às características do usuário simulado. Nestes experimentos simulou-se um *joystick* para obter a posição desejada em função do tempo, adicionou-se sinais de ruído e filtrou-se o sinal pelo método descrito. Os resultados obtidos são descritos em mais detalhes no tópico resultados e discussões.

Na figura 6.8 é possível observar a simulação de uma entrada tipo rampa (linha roxa), na qual se adiciona um erro aleatório de +4 a -4 unidades (linha amarela) e a média resultante (linha azul claro). Em função da média (linha azul claro), o centro da *deadzone* móvel (limitada pelas linhas verde e vermelha) resulta na saída (linha azul escuro), que coincide com a entrada do usuário de forma a filtrar os ruídos inseridos.



Fig. 6.8 - Software para obtenção da posição desejada pelo usuário.

Para o desenvolvimento deste algoritmo procurou-se utilizar recursos de processamento que possibilitem a sua implementação futura em eletrônica embarcada, por intermédio de microcontroladores. Esta eletrônica embarcada permitirá adaptar *joysticks* de cadeiras de rodas para usuários que possuem contrações involuntárias nas mãos, bem como demais aplicações que relacionem este tipo de usuário e interfaces de características contínua como alavancas ou volantes.

6.3. Implementação do HaWCoS em Microcontroladores PIC

As arquiteturas de interface HaWCoS, desenvolvidas por Felzer e Freisleben (2002) permitem que usuários realizem interface com computadores ou cadeiras de rodas por meio de uma única entrada, como ilustrado na figura 6.9.

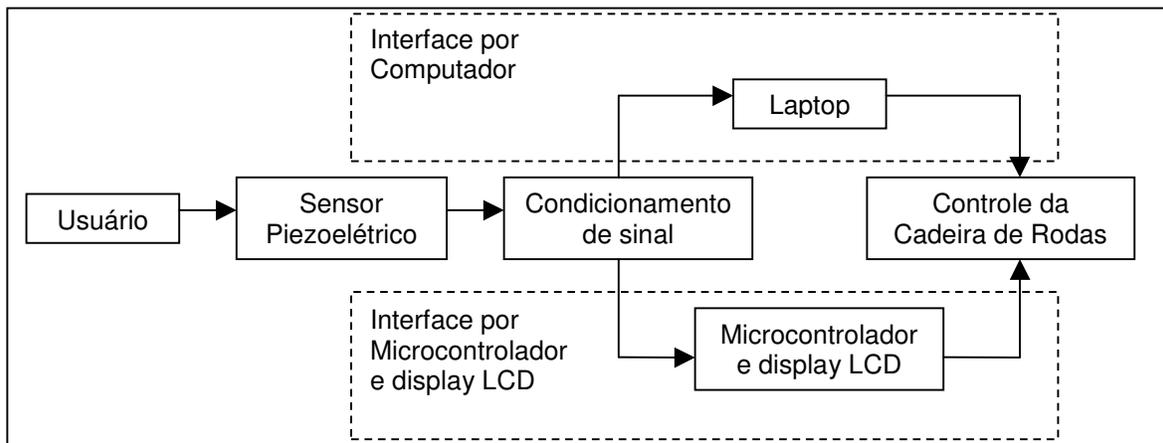


Fig. 6.9 - Arquiteturas do Sistema HaWCoS (fonte: Felzer e Freisleben, 2002).

Esta técnica pode ser implementada por meio de duas arquiteturas: utilizando um computador ou por meio de microcontrolador e display de LCD. Ambas as arquiteturas já se encontram implementadas para controle de uma cadeira de rodas elétrica e testadas por Torsten Felzer que é um dos desenvolvedores desta interface.

Observando a necessidade de viabilizar o acesso desta interface para usuários com menor poder aquisitivo, optou-se por desenvolver um novo protótipo desta interface para um kit de motorização de baixo custo. Este novo protótipo de interface foi desenvolvido em parceria com o pesquisador Torsten Felzer, de modo a combinar os conceitos já utilizados na interface HaWCoS com os necessários ao kit de motorização. Desta forma foi possível observar três pontos a serem adaptados para o kit de motorização: controle de motores, diversidade de comandos (e custo associado a respectivas funções), e protocolo de controle.

O kit de motorização possui um controle discreto de motores, ou seja, ligar ou desligar e sentido de rotação sem permitir o uso de velocidades intermediárias. Este controle permite uma significativa redução de custos associados a interface e *drivers* utilizados. Diversamente, o sistema HaWCoS permite um controle contínuo de velocidades à medida em que acelera os motores de modo progressivo para a direção selecionada, permitindo uma movimentação suave.

Outro ponto a ser observado é que a interface HaWCoS possui uma grande diversidade de funções, além do controle de movimentos de uma cadeira de rodas, como utilizar aplicativos de um laptop (caso este esteja presente), definir curva de aceleração e ligar faróis auxiliares. No kit de motorização estas funções não estão presentes, o que permite delimitar apenas o controle de movimentos da cadeira como função. Esta redução de comandos permite simplificar a interface com o usuário ao utilizar um painel de 14 LED's, associado diretamente a um microcontrolador de baixo custo, como o PIC 16F877. É necessário ressaltar que estas simplificações, embora permitam reduzir o custo da interface, também eliminam diversas funções presentes no sistema HaWCoS.

O terceiro ponto a se adaptar associa-se ao protocolo de interface utilizado, pois o sistema HaWCoS possui uma interface desenvolvida para as características da cadeira de rodas utilizada, como tipo de *joystick* e níveis de tensão. O protocolo utilizado no kit de permite alternar entre o sistema HaWCoS desenvolvido e um *joystick* já utilizado no kit de motorização.

Estes três requisitos permitiram delinear uma interface que mantém o conceito utilizado pelo sistema HaWCoS, contudo controlando discretamente os motores, utilizando um *display* de 14 LED's e um protocolo dedicado ao padrão utilizado no kit de motorização. A figura 6.10 representa a arquitetura desenvolvida com base no sistema HaWCoS e mais detalhes sobre a programação do microcontrolador estão presentes nos apêndice IV e apêndice VI.

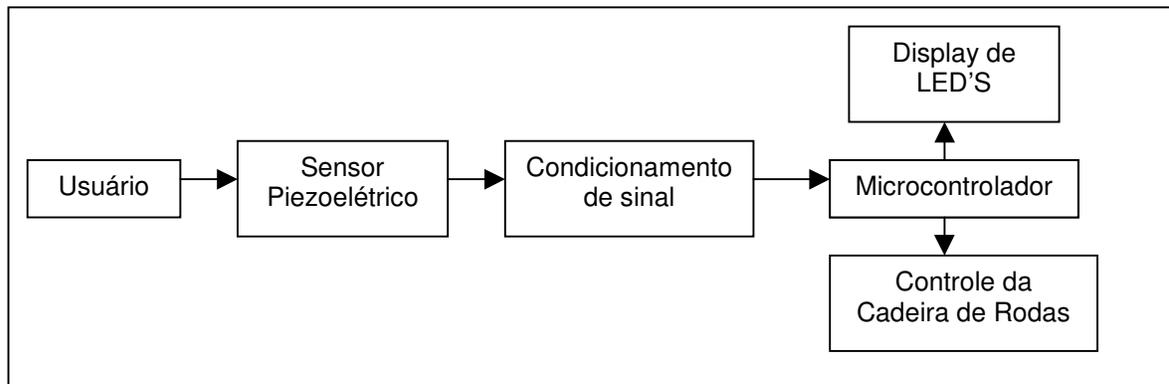


Fig. 6.10 - Arquitetura utilizada para interface

Desta forma é possível detalhar esta interface em três sub-sistemas: condicionamento de sinal, processamento (microcontrolador) e interface de saída (*display* de LED's).

6.3.1. Condicionamento de sinal

Para a interação com a interface HaWCoS, o usuário utiliza como interface de entrada o sensor piezoelétrico. Este dispositivo, ilustrado na figura 6.11, normalmente é utilizado como fonte sonora ao ser estimulado por uma fonte de tensão elétrica alterna (AC), permitindo seu uso para toque de celulares, alarmes e brinquedos.

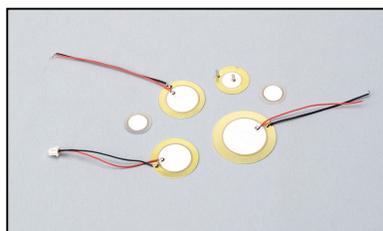


Fig. 6.11 - Piezoelétrico (fonte: International Components Cooperation, 2007).

Assim, a tensão elétrica AC é convertida em ondas sonoras por meio das contrações e dilatações do piezoelétrico, de forma análoga caso este seja submetido a uma compressão ou dilatação, o mesmo retorna uma tensão elétrica. Esta tensão elétrica é proporcional a suas características construtivas e a compressão ou dilatação aplicada. Desta forma, realizaram-se testes ao comprimir manualmente piezoelétricos e obter diretamente a sua tensão em um osciloscópio.

Os resultados demonstraram pulsos de tensões entre 200mV e 2V, significativamente acima das tensões presentes em técnicas EMG e EOG, como no projeto EagleEyes apresentado por Gips (1998) com tensões em torno de 100 microvolts. Considerando que as tensões do piezoelétrico permitiram uma relação de sinal e ruído que dispensa a necessidade de filtros, implementou-se o piezoelétrico por meio de uma ponte H. Esta ponte H, por sua vez, é associada a um amplificador operacional em configuração de ganho infinito.

Implementou-se esta configuração de ganho infinito pois é reconhecido que amplificadores operacionais reais não apresentam ganho infinito, mas uma condição de ganho máximo, que permite ajustar a sensibilidade nos resistores da ponte H. Desta forma, os pulsos de tensão amostrados no piezoelétrico são amplificados para valores entre 0 e 5 volts. Esta amplificação permite a amostragem do sinal por meio de bordas de subida (2 a 2,5 volts) em um microcontrolador PIC 16F877A, responsável pelo processamento e controle.

Observa-se que, caso o usuário possua dificuldades em se adaptar a este tipo de sensor piezoelétrico, pode ser utilizado outro tipo de sensor. O único requisito necessário é a saída em borda de subida de 2 a 2,5 volts a cada interação do usuário, mantendo os sistemas de interface e controle descritos a seguir.

6.3.2. Processamento e Controle

O segundo sub-sistema é implementado em um microcontrolador PIC 16F877A e possui quatro finalidades: receber os dados da interface de entrada, processá-los, exibi-los na interface de saída e controlar os *drivers*.

A recepção dos dados do usuário ocorre por meio de bordas de subida obtidas após a amplificação do sinal do piezoelétrico, resultando em um clique. Caso sejam detectadas duas bordas de subida, em um curto intervalo de tempo, é reconhecido como duplo clique. Esta forma de interpretação (um ou dois cliques), em função do intervalo de tempo entre interações, é derivada do sistema HaWCoS de Felzer e Freisleben (2002). A interpretação de um ou dois cliques permite controlar e configurar todo o funcionamento da interface por meio de uma única entrada associada ao piezoelétrico.

Desta forma, também é possível selecionar entre dois métodos de movimentação: com pausas e sem pausas.

- *Com pausas*: os comandos de direção se mantêm, de modo a serem alternados a cada evento de um clique. Um evento de dois cliques retorna ao modo de configuração. Neste modo simplificado, após cada evento de um clique, o movimento da cadeira de rodas é interrompido, e retorna após um período pré-definido de 1 a 2 segundos. Este período permite que o usuário escolha a direção de movimento com a cadeira imobilizada
- *Sem pausas*: apresenta duas diferenças ao modo anterior: não apresenta pausas entre os movimentos, e permite alterar a disposição dos comandos de movimento. Este modo sem pausas se assemelha ao método de movimentação HaWCoS de Felzer e Freisleben (2002), no qual não há pausas quando se alterna entre as direções de movimento.

Ainda que o microcontrolador utilizado possua duas saídas PWM, capazes de controlar a velocidade de modo independente para cada um dos dois motores, o driver implementado no protótipo não permite este controle de velocidade. Desta forma, não serão implementadas as funções de aceleração, presentes no sistema HaWCoS de Felzer e Freisleben (2002). Assim, o controle dos motores se resume a cinco pinos, quatro de controle de movimentos e um referencial, permitindo utilizar o mesmo padrão de *joysticks* Atari e Master System.

O *layout* de disposição dos controles frente, trás, esquerda e direita pode ser alterado na movimentação sem pausas, permitindo alternar entre quatro disposições pré-definidas semelhantes ao sistema HaWCoS de Felzer e Freisleben (2002). Para o modo de configuração, e nos dois modos de controle, é necessário um retorno para que o usuário acompanhe como esta

interação ocorre. Este retorno é exibido por meio de uma interface de saída na forma de um painel de 14 LED's multiplexados e controlados pelo microcontrolador.

6.3.3. Interface de Saída e Controle de Movimentos

A interface de saída, representada na figura 6.12, é responsável por demonstrar em tempo real ao usuário como a interface está operando e em qual modo o usuário se situa (configuração, movimento com pausa ou movimento sem pausa).

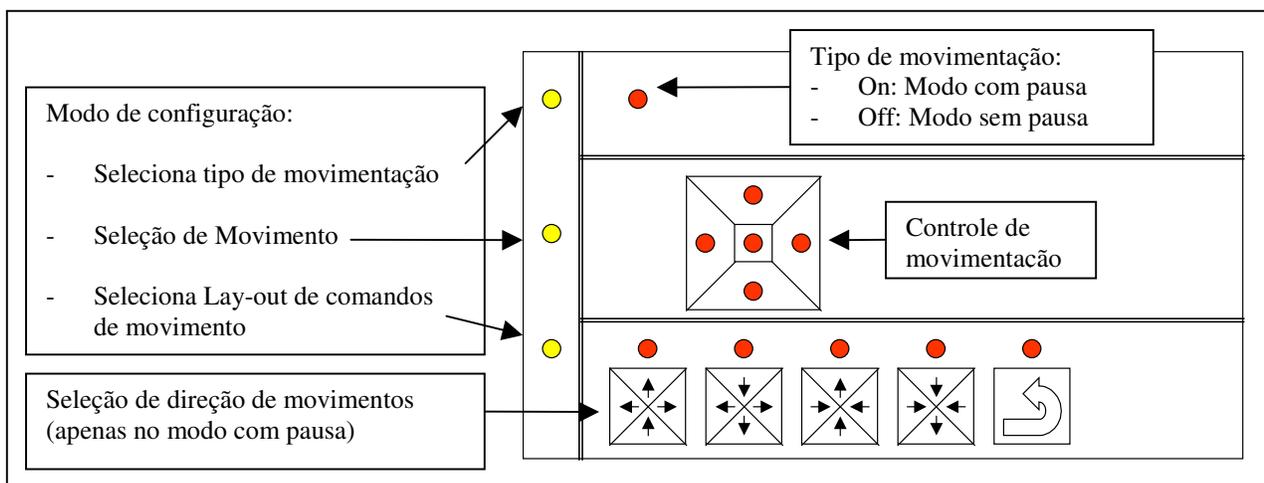


Fig. 6.12 - Interface de saída.

Seguindo o conceito da interface HaWCoS de Felzer e Freisleben (2002), toda a seleção de comandos é realizada por meio de comandos de um ou dois cliques. Deste modo, manteve-se como referência o método de navegação interface HaWCoS e retiraram-se os comandos não implementados no kit de motorização, como a aceleração, controle de velocidade máxima e ligar faróis.

Outras funções como período máximo entre dois cliques e tempo de espera para acionar movimento (modo com pausa) são definidas em tempo de programação do microcontrolador. Deste modo, evita-se que o usuário altere configurações que lhe impeçam de utilizar a interface de modo adequado. Esta iniciativa, de retirar comandos desnecessários ao kit de motorização e eliminar o acesso a configurações que possam prejudicar o sistema, facilita a operação e aumenta a robustez da interface.

Desta forma, com um número reduzido de comandos, definiu-se um padrão para a navegação em função de um ou dois cliques, como ilustrado na figura 6.13:

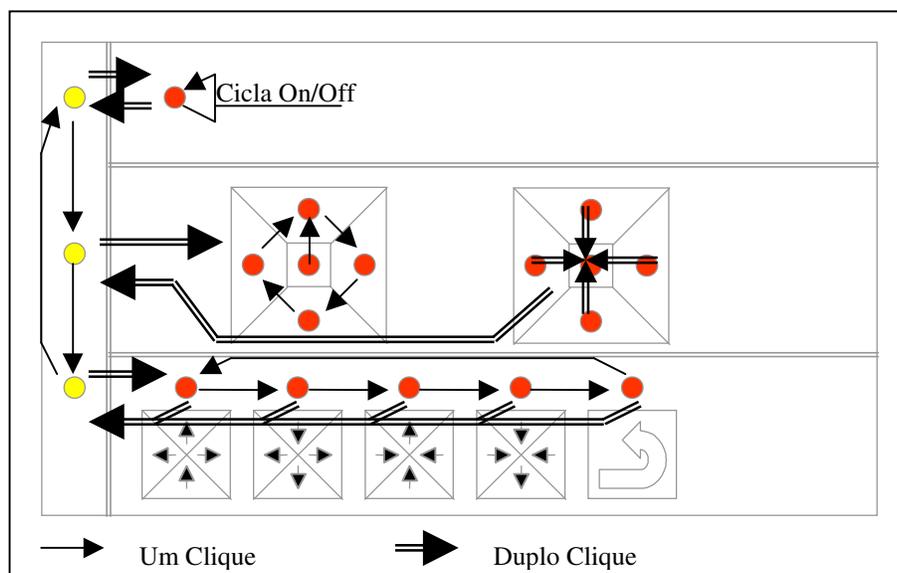


Fig. 6.13 - Navegação na interface por cliques.

Este padrão de um ou dois cliques utiliza: dois cliques para alternar entre o modo de configuração (esquerda, com LED's amarelos) e a configuração selecionada (direita, com LED's vermelhos); e um clique dentro de cada configuração para a seleção. Desta forma, é possível manter a consistência: para sair das caixas demarcadas com duas linhas, ver figura 6.12 e figura 6.13, são utilizados dois cliques; e para ciclar entre suas opções por meio de um clique.

Como exemplo, para movimentar a cadeira, quando o sistema é ligado apenas o primeiro LED amarelo (posição superior) e o LED vermelho central (controle de movimentação) encontram-se acesos: o usuário encontra-se no modo de configuração. Após um clique, este primeiro LED se apaga e o segundo LED amarelo se acende. Quando o usuário realiza dois cliques, a interface responde piscando todos os LED's acesos e inicia o modo de controle de movimentos. Neste modo, quando realiza-se o primeiro clique é selecionado o movimento para frente e a cada clique alterna-se a direção de movimentos: frente, direita, trás, esquerda e frente novamente. O movimento da cadeira inicia-se após selecionar uma direção por um intervalo de tempo, pré-definido como 1 segundo. Quando o usuário deseja parar a movimentação, este

realiza dois cliques e retorna-se ao modo de configuração, indicado por um piscar de todos os LED's.

Retornando ao modo de configuração, é possível ciclar entre as opções a cada clique, que alternam entre os LED's amarelos do superior ao inferior. Quando o terceiro LED encontra-se selecionado (aceso) é possível realizar um duplo clique para alterar o *layout* de controle. No modo de seleção de *layout*, de forma semelhante ao modo de movimentação, é possível alternar entre as opções a cada clique e selecionar a disposição dos comandos frente, trás, esquerda e direita presentes no modo de controle de movimentação. A escolha de layout permite, por exemplo, otimizar a movimentação para frente (eliminando a direção trás) e desta forma reduzir o número de interações do usuário. Observa-se também uma quinta opção: retorno, nesta opção mantém-se o *layout* anteriormente selecionado. Para confirmar a seleção de um determinado layout realiza-se duplo clique no layout selecionado, retornando-se ao modo de configuração.

Considerando que este processo de seleção de opções pode dificultar o uso desta interface para usuários iniciantes, duas opções de interface simplificada são propostas. Em ambas as soluções não se utiliza a seleção de movimentos com ou sem pausas, define-se como *default* a utilização da movimentação com pausas. Assim o primeiro modelo de interface simplificada consiste em utilizar apenas o controle de movimentação, sem permitir alterações de *layout*, como ilustrado na figura 6.14. Nesta solução o comando de duplo clique para a movimentação e os comandos de clique simples alternam a direção de movimentos.

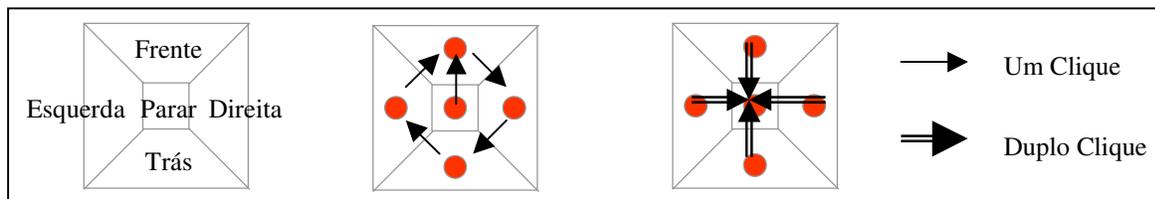


Fig. 6.14 - Interface simplificada modelo 1.

Reconhecendo a necessidade em otimizar o controle para determinadas situações propõe-se um segundo modelo para uma interface simplificada, acrescentando a seleção de layout na interface anterior. Assim a segundo interface simplificada é uma alternativa intermediária entre a proposta na figura 6.14 (apenas movimentação) e a proposta nas figuras 6.12 e 6.13 (movimentação e seleção de layout). Este segundo modelo de interface, ilustrado na figura 6.15, acrescenta um layout com as direções de movimento padrão e elimina a seleção de movimentação com ou sem pausas, ilustrada na figura 6.12.

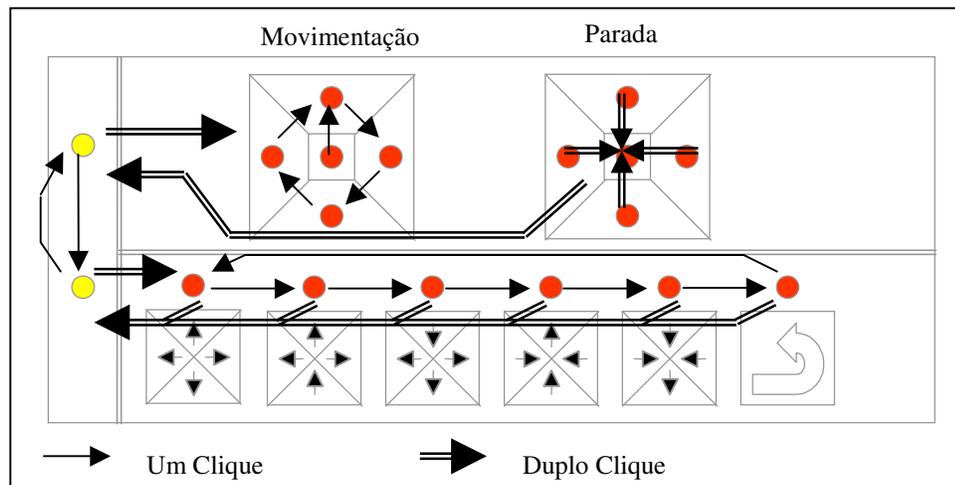


Fig. 6.15 - Interface simplificada modelo 2.

Desta forma, o *layout* padrão utilizado é o primeiro, que permite movimentar-se para na sequência frente, direita, trás, esquerda e frente novamente. Caso o usuário deseje alternar o layout, a movimentação deve ser parada com duplo clique e a opção de selecionar layout é iniciada. Após confirmar o layout desejado retorna-se ao modo de movimentação.

6.4. Integração entre Interfaces e Sistemas Auxiliares

Segundo a proposta de Yanco (1998b), o desenvolvimento de protótipos para interfaces e sistemas de auxílio à navegação é seguido da sua modularização. Este tópico descreve um protocolo para modularizar interfaces, e sistemas auxiliares de navegação, em um kit de motorização de cadeira de rodas. Este kit de motorização foi desenvolvido pelo Laboratório de Projeto Mecânico da Unicamp e descrito por Alvarenga (2006), integrante da equipe de desenvolvimento deste kit de motorização.

Contudo, em casos de limitações severas, como usuários que apresentam movimentos involuntários nas mãos, é necessário utilizar outro tipo de interface, como voz ou contrações musculares. Em casos mais severos, em função das características da interface ou do usuário, pode ainda ser necessária a adição de um sistema de auxílio à navegação. Este sistema de auxílio à navegação tem por objetivo suprir as necessidades ainda presentes após a utilização de uma determinada interface, como detectar obstáculos e evitar colisões.

Seguindo a relação usuário e tarefa, delimitou-se dois tipos de interface a serem utilizadas: HaWCoS embarcado em microcontrolador e *joystick*; uma função evitar colisões por sonar para auxílio a navegação e levantaram-se as características de entrada do *driver* de controle para os motores. Observou-se que neste *driver* de controle há um par de entradas para cada motor. Nestes pares utilizam-se comandos discretos (0V ou 24V) para representar os sentidos de rotação, como ilustrado na tabela 6.2:

Tabela 6.2 - Lógica de controle para o driver.

Movimento	Entrada A (V)	Entrada B (V)
Parado	0v	0v
Movimento Horário	0v	24v
Movimento Anti-horário	24v	0v
Parado	24v	24v

Observando que o microcontrolador e o sonar utilizam em suas valores entre 0V e 5V, também compatíveis ao joystick já presente, foi necessário desenvolver um módulo adicional para adaptar-se os níveis de tensão (MAANT). O desenvolvimento deste módulo ocorreu por intermédio da abordagem usuário e tarefa, na qual foi possível localizar as interfaces (*joystick* e sistema HaWCoS embarcado em microcontrolador), sistemas de auxílio a navegação (sonar), *driver* de controle (tarefa) e o fluxo de dados entre estas (setas).

Foi possível relacionar como fluxo de dados três comandos de movimento por *driver*, pois há duas direções de movimento e o comando parar. Combinando os dois *drivers*, observam-se 6 comandos necessários para a interface com a cadeira de rodas e mais dois fios para a alimentação dos sistemas envolvidos. Deste modo, a técnica M3S, apresentada por Nelisse (1998), demonstra ser possível modularizar e integrar estes componentes por meio de um protocolo de comunicação serial, gerenciado por uma central de controle e configuração.

Assim, a meta de integrar e modularizar os sistemas de interface, auxílio a navegação e *drivers* para motores é atendida pelo protocolo M3S. Contudo é necessário um controlador central, e que todos os dispositivos envolvidos possuam capacidade de realizar uma comunicação serial. Deste modo, observou-se que para a implementação de um protocolo de comunicação serial e controlador central, é necessário o emprego de unidades de processamento (microcontroladores ou microcomputadores) em cada módulo.

Considerando que o sistema a ser controlado demanda apenas quatro direções ou comandos, optou-se por utilizar um protocolo simplificado de comunicação, ao evitar a comunicação serial, mas mantendo a modularidade. Observando que o número de comandos distintos são 6, uma alternativa que se demonstrou viável em relação à interface serial é a comunicação paralela entre os módulos. Neste caso, associa-se uma via de comunicação para cada tipo de informação, resultando também em 6 vias. Foi possível reduzir o número de vias 6 para 4, ao observar que o comando parar ocorre na ausência dos dois comandos de movimentar-se, como demonstrado na tabela 6.2.

Desta forma, foi possível delimitar comunicação com o driver por meio de quatro linhas de comunicação (para a movimentação) e mais duas para alimentação e referencial de fonte de energia, como ilustrado na figura 6.16. É necessário observar que os pinos de 1 a 4, relacionados a movimentação, já estão representando os contatos do joystick.

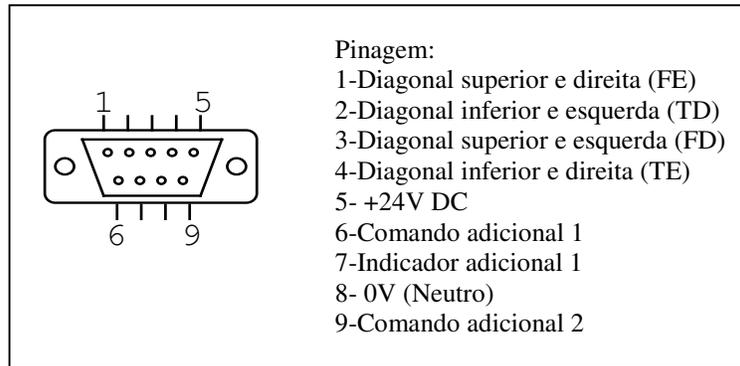


Fig. 6.16 - Configuração de pinos no protocolo utilizado.

Por meio desta padronização das entradas do *driver*, que é semelhante a utilizada em *joysticks* Master System, foi possível construir um MAANT (módulo adicional para adaptar de níveis de tensão). O MAANT reconhece 0V como nível lógico 0, e permite receber tensões entre 5 a 24 volts para nível lógico 1 (equivalente a 24V para o driver). Assim, mantém-se a mesma configuração de pinos apresentada na figura 6.16 para o módulo seguinte e permite-se controlar os movimentos da cadeira de rodas ao combinar os comandos de direção utilizados, como ilustrado na tabela 6.3.

Tabela 6.3 - Relação entre comandos de direção e movimentos

Códigos de movimentos (figura 6.16)				
FE	FD	TE	TD	Direção
5 a 24V	0V	0V	0V	Avanço à direita
5 a 24V	5 a 24v	0V	0V	Frente
0V	5 a 24V	0V	0V	Avanço à esquerda
0V	5 a 24V	5 a 24V	0V	Rotação anti-horária sobre eixo
0V	0V	5 a 24V	0V	Recoo à direita
0V	0V	5 a 24V	5 a 24V	Trás
0V	0V	0V	5 a 24V	Recoo à esquerda
5 a 24V	0V	0V	5 a 24V	Rotação horária sobre eixo

O par de fios associados à alimentação é diretamente conectado às baterias como uma fonte de 24 volts. Caso estejam presentes circuitos integrados, como microcontroladores, é possível utilizar reguladores de tensão LM7805 associados a capacitores para obter uma fonte de 5 volts sem quedas, como ilustrado na figura 6.17.

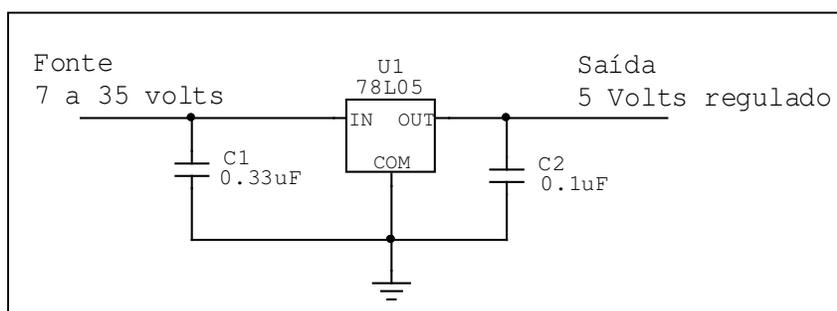


Fig. 6.17 - Regulador de tensão (fonte: STMicroelectronics, 2007).

Após padronizar os níveis de entrada do *driver* por meio do MAANT, e o protocolo demonstrado na figura 6.16, foi necessário integrar as duas interfaces e o sistema de navegação no mesmo protocolo. Uma abordagem tradicional sugere um sistema de processamento cartesiano, como observado por Gomi e Griffith (1998) e implementado por Madarasz *et al.* (1986). Nesta abordagem cartesiana, é necessária a presença de uma unidade central, responsável

por processar a informação proveniente dos sensores e interfaces com o usuário, e agregando uma maior complexidade ao sistema.

Procurou-se então descentralizar o processamento, seguindo o conceito de uma arquitetura *subsumption*, para permitir que cada módulo seja inserido ou retirado sem a necessidade de reprogramação do sistema pelo usuário. Desta forma, o conceito da modularidade presente nas arquiteturas comportamental de Gomi e Griffith (1998), e *subsumption* de Brooks (1986), permite um processamento descentralizado e paralelo entre os módulos de interface e auxílio à navegação. Este conceito de integração presente nas arquiteturas comportamental e *subsumption* dispensa a necessidade de uma central de controle e configuração, presente na técnica M3S.

Seguindo este conceito de atribuir comportamentos para cada módulo foi possível organizá-los em uma hierarquia de controle, como ilustrado no diagrama da figura 6.18.

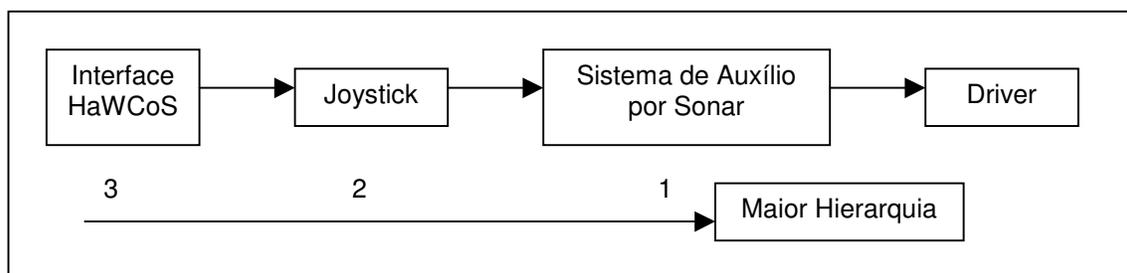


Fig. 6.18 - Exemplo de integração hierárquica entre módulos.

Assim foi possível atribuir comportamentos para cada elemento da módulo:

- *Interface HaWCoS e Joystick*: recebem dados do usuário e controlam os quatro comandos de direção;
- *Sistema de Auxílio por Sonar*: quando detecta obstáculos desabilita determinadas direções de movimento;
- *Driver*: recebe informações dos elementos anteriores e controla os motores;

Observa-se que o sistema de auxílio à navegação por sonar, que recebe informações das interfaces, sobrepõe-se a comandos de hierarquias anteriores quando ocorre um evento de detecção de objeto. Nestes eventos o módulo associado ao sonar realiza seu comportamento de inibir movimentos na direção do obstáculo detectado. Desta forma, a arquitetura representada na figura 6.14 demonstra como os módulos são organizados na arquitetura utilizada:

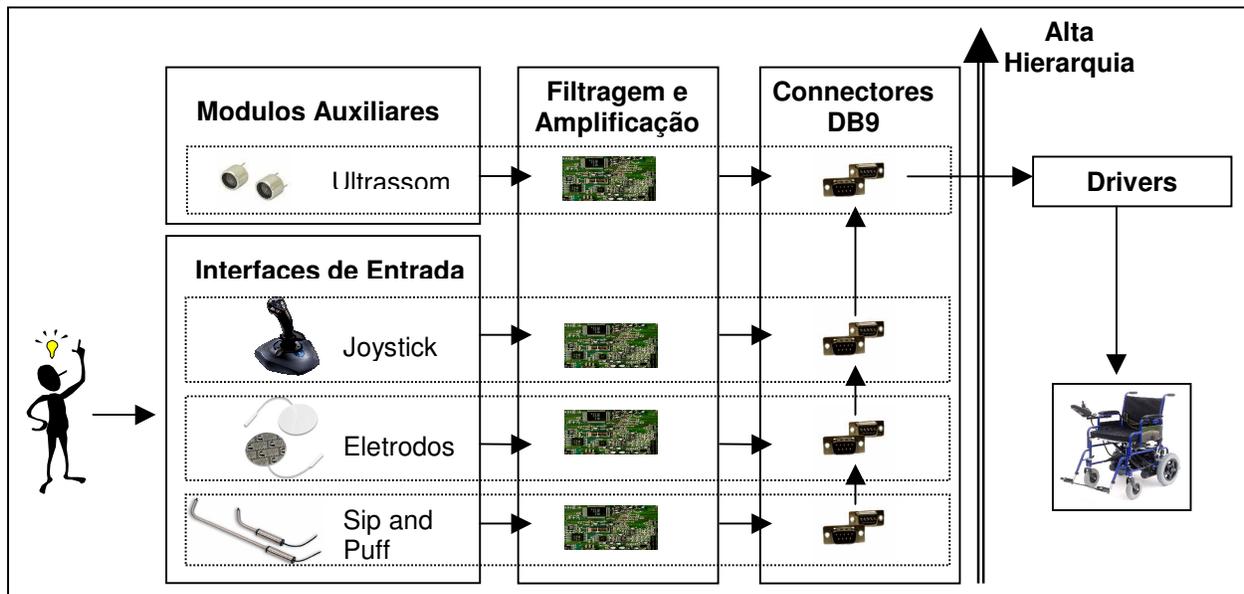


Fig. 6.19 - Arquitetura modular utilizada.

Nesta arquitetura foi possível manter o protocolo de interface paralela, apresentada na figura 6.12, ao incorporar os circuitos de filtragem e amplificação as suas respectivas interfaces e sistemas de auxílio à navegação. Para isto, o detector de sonar (ou ultrassom) possui todo o processamento e condicionamento de sinal necessários embutidos em seu módulo. De forma análoga, a interface HaWCoS embutida em microcontrolador também possui todo o hardware necessário contido em seu módulo. Esta modularização permite retornar para este protocolo *subsumption* apenas o seu comportamento: receber dados do usuário e controlar direções.

Capítulo 7

Resultados e Discussões

Este tópico realiza uma revisão a respeito de experimentos realizados com os sistemas desenvolvidos, apresentando os resultados positivos e algumas dificuldades encontradas.

7.1. Reconhecimento de Voz

Os resultados obtidos com o software comercial de reconhecimento de voz demonstraram ser possível o controle de uma cadeira de rodas por voz, além de outras alternativas, como a automação residencial. Contudo esta arquitetura pode ser inadequada para a implementação em cadeiras de rodas, devido ao seu consumo energético, dimensões e custos de hardware.

Uma segunda alternativa de implementação em microcontrolador permitiu uma significativa redução em seu consumo energético, dimensões e custos de hardware. Deste modo a utilização do computador externo ocorre apenas em fases em que não é necessário o controle da cadeira de rodas, permitindo considerar esta segunda arquitetura como embarcada, e associando uma significativa redução de custos e maior mobilidade.

A estimativa de custos, levantada em lojas de eletrônica em setembro de 2007, para os principais componentes desta segunda arquitetura é apresentada na tabela 7.1:

Tabela 7.1 - Estimativa de custos da segunda arquitetura

Componente	Preço (R\$)
Microcontrolador 1 (PIC 16F877A)	17,00
Componentes para módulo de interface serial	20,00
Placa para microcontrolador	20,00
Amplificador para microfone	10,00
Microfone tipo IBM PC	25,00
Caixas e conectores	15,00
Custo total	107,00

Observa-se que esta estimativa não considera o microcontrolador 2, que também está presente na primeira arquitetura, que utiliza reconhecimento de voz, pois o objetivo é comparar os preços de ambas as arquiteturas. Desta forma é possível observar uma significativa diferença entre custos:

- *Primeira arquitetura*: custo estimado em R\$ 2050,00: software de reconhecimento de voz (aproximadamente R\$ 350) e um computador laptop (aproximadamente R\$1700,00);
- *Segunda arquitetura*: custo estimado para componentes em R\$ 107,00, desprezando-se a mão de obra.

Contudo, dada a simplicidade do algoritmo de reconhecimento de voz implementado em microcontrolador, há uma maior dificuldade no reconhecimento palavras ou sons vocalizados, especialmente quando a pronuncia do usuário demanda um tempo maior que o correspondente na biblioteca. Em outras palavras, pronunciar rapidamente gera um mau casamento entre a curva de frequência amostrada e a correspondente na biblioteca, pronunciada em velocidade normal. De forma análoga, a pronuncia em um período menor que o da biblioteca também dificulta a identificação do comando solicitado.

Assim, observa-se que há uma necessidade em escolher qual arquitetura melhor se adapta para cada sistema embarcado: as reduções em custo e em dimensão de um microcontrolador, ou a grande biblioteca e maior confiabilidade presentes no reconhecimento de voz de um computador.

7.2. Filtragem de *Joystick* por Fuzzy

Para obtenção dos resultados modelou-se um usuário, em LabView, por meio de uma simulação de um *joystick* de um eixo. Esta simulação de joystick demonstra a posição desejada na qual adiciona-se o ruído (fontes senoidal ou randômica) de amplitude regulável. A figura 7.1 ilustra estas duas fontes de ruído (amarelo) adicionadas ao sinal de entrada desejado (roxo) com uma amplitude, de pico a pico, de 5 unidades.

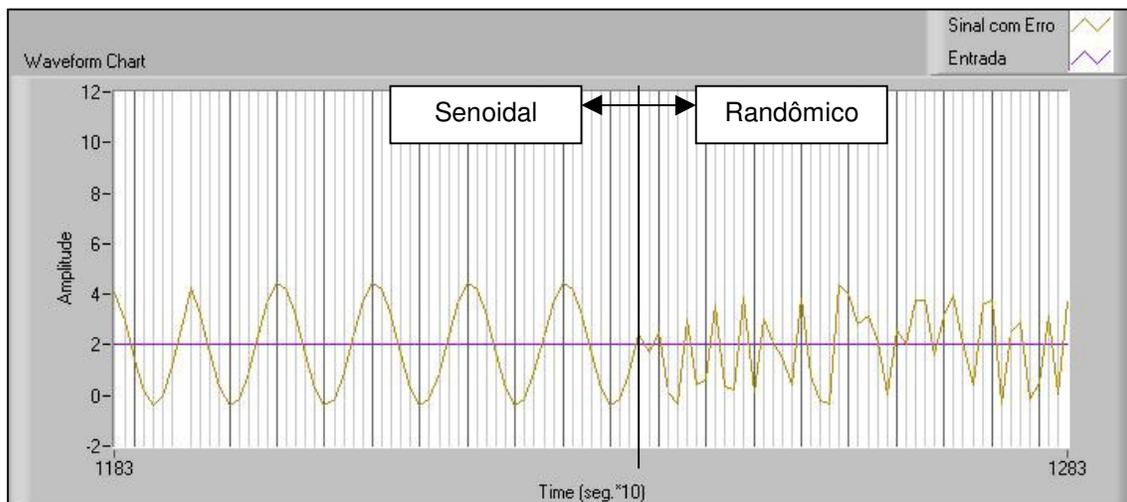


Fig. 7.1 - Fontes de ruído senoidal (1Hz) e randômico adicionada a uma entrada.

As escalas de tempo do gráfico (0,1 segundo/divisão) e a taxa de amostragem do sinal de entrada (10Hz) foram mantidas em todos os testes realizados. Este sinal é amostrado a 10Hz (100ms entre cada aquisição) para posterior filtragem por meio do filtro passa baixa e fuzzy já descritos. Para observar a estabilidade e tempo de resposta, avaliaram-se os regimes estacionário (mantém-se o valor da entrada) e transiente (entrada degrau de 8 unidades), ambos considerando a adição de ruído. Deste modo, foi possível obter o comportamento para as entradas senoidal e randômica, bem como posteriormente alterar dois parâmetros: número de amostras para o filtro passa baixa (mantendo a taxa de amostragem em 10Hz) e a amplitude para o filtro fuzzy.

Testes realizados demonstraram que, para uma mesma frequência de amostragem, o aumento no número de amostras resulta em um tempo maior de acomodação de forma proporcional. Desta forma, em uma entrada degrau de 8 unidades sem a adição de distúrbios, para cada 10 amostras a acomodação demanda 1 segundo: 20 amostras equívalem a 2 segundos e 50 amostras a 5 segundos. Um exemplo da obtenção deste tempo de resposta pode ser observado na figura 7.2, que simula uma entrada degrau sem ruído (neste caso a legenda “sinal com erro” equívale a entrada).

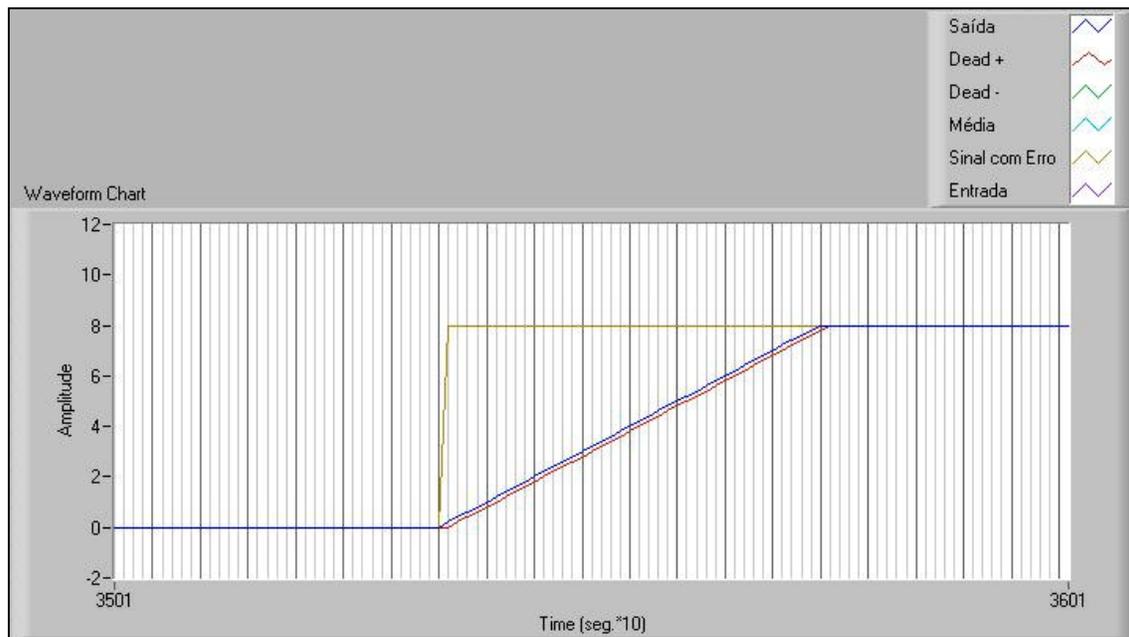


Fig. 7.2 - Acomodação de 4 segundos para 40 amostras.

Os testes com uma fonte de ruído senoidal visam representar um movimento oscilatório em torno de um ponto. Foi possível observar que, quando o período de amostragem do filtro passa baixa é igual ao período de um ciclo completo da entrada senoidal, é possível eliminar esta fonte de distúrbio ao custo de um atraso de resposta. Como exemplos, mantendo a frequência de amostragem em 10Hz, para um ruído senoidal de 1Hz (período de 1 segundo/ciclo) são necessárias 10 amostras a cada 0,1 segundo (período de 1 segundo/ciclo). Da mesma forma e para 0,5Hz (período de 2 segundo/ciclo), é possível utilizar 20 amostras a cada 0,1 segundos (período de 2 segundo/ciclo).

Esta relação permite reduzir o número necessário de amostragens para a obtenção de uma maior velocidade de resposta, como ilustrado na figura 7.3, ilustrando um ruído de 1Hz e amplitude 5 unidades, filtrado por 10 amostras no filtro passa baixa.

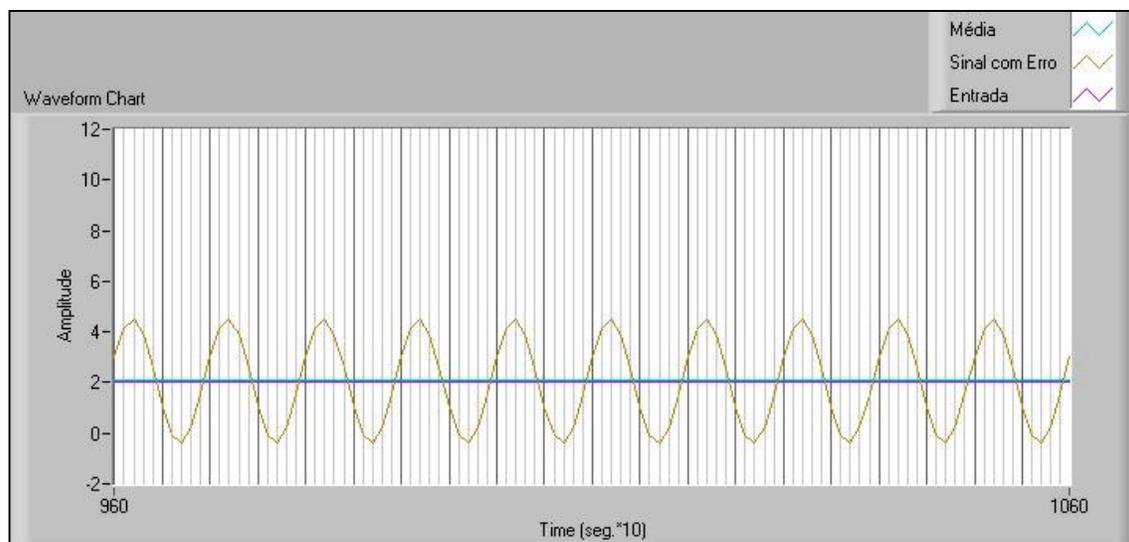


Fig. 7.3 - Supressão do ruído senoidal por filtragem.

Observa-se que amostragens múltiplas de uma determinada frequência apresentam o mesmo efeito. Desta forma, para uma entrada de 2Hz, taxas de amostragens de 2Hz, 4Hz, 6Hz e 8Hz também eliminarão esta fonte de ruído. Estas taxas múltiplas para amostragem podem ser utilizadas quando um ciclo do sinal de entrada resulta em amostragens inferiores a 5. Desta forma, utiliza-se um múltiplo de amostragem do sinal de entrada igual ou superior a 5, evitando problemas relacionados a baixas amostragens. Desta forma observou-se que, quando a frequência

de oscilação é conhecida, não é necessário o filtro fuzzy (relacionado a incerteza), pois a sua amplitude pode ser definida como zero.

Ao submeter estes mesmos parâmetros de entrada e filtragem a um degrau de 8 unidades observa-se um *delay* de aproximadamente 0,5 segundos entre o sinal aplicado sem ruído e a média obtida. Este *delay* de 0,5 segundos pode ser observado na figura 7.4 e está diretamente relacionado ao número de amostragens. Nesta figura as linhas leves representam 0,1 segundos e linhas grossas representam 0,5 segundos, mantendo a mesma escala dos gráficos anteriores.

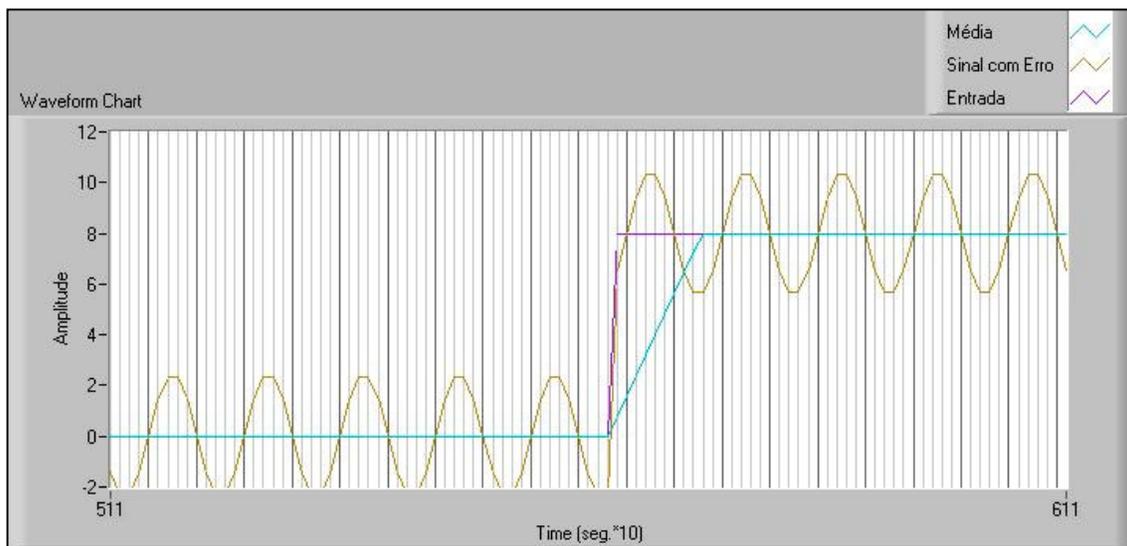


Fig. 7.4 - Resposta em degrau para ruído senoidal de 1Hz.

Após os testes para distúrbios senoidais, realizaram-se os testes para uma entrada somada a um ruído randômico. É necessário observar que não é mais possível definir o número de amostras em função da frequência do ruído pois, posto que este é randômico, não possui uma ou mais frequências equivalentes definidas. Neste caso é necessário combinar os valores de número de amostragens (filtro passa baixa), com a amplitude da incerteza (filtro fuzzy) para um tratamento adequado. Estes parâmetros são definidos em função do tempo de resposta e da amplitude do sinal resultante após a filtragem passa baixa.

Desta forma, adicionou-se um ruído randômico de amplitude máxima 5 unidades, de pico a pico, ao sinal de entrada desejada. O número de amostragens do filtro passa baixa, que se relaciona se relaciona ao tempo de resposta, foi mantido em 10 amostragens posto que na

filtragem anterior foi possível obter uma resposta satisfatória de 0,9 segundo para 8 unidades em entrada degrau. A demonstração de uma entrada degrau de 8 unidades nestes parâmetros é observada na figura 7.5:

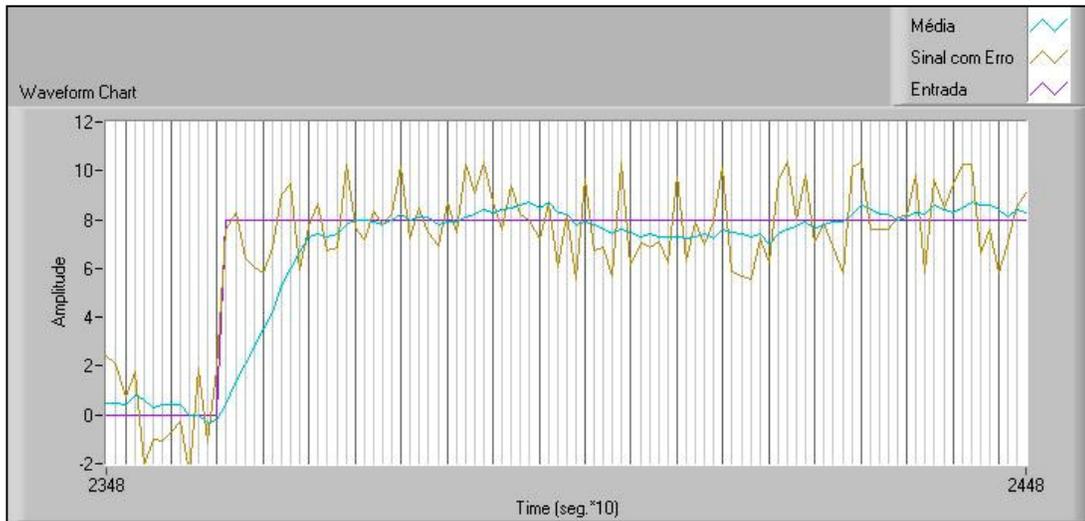


Fig. 7.5 - Filtragem passa baixa de um ruído randômico.

Posteriormente a esta filtragem passa baixa, utilizou-se o filtro fuzzy com amplitude de 2 unidades, como demonstrado na figura 7.6.

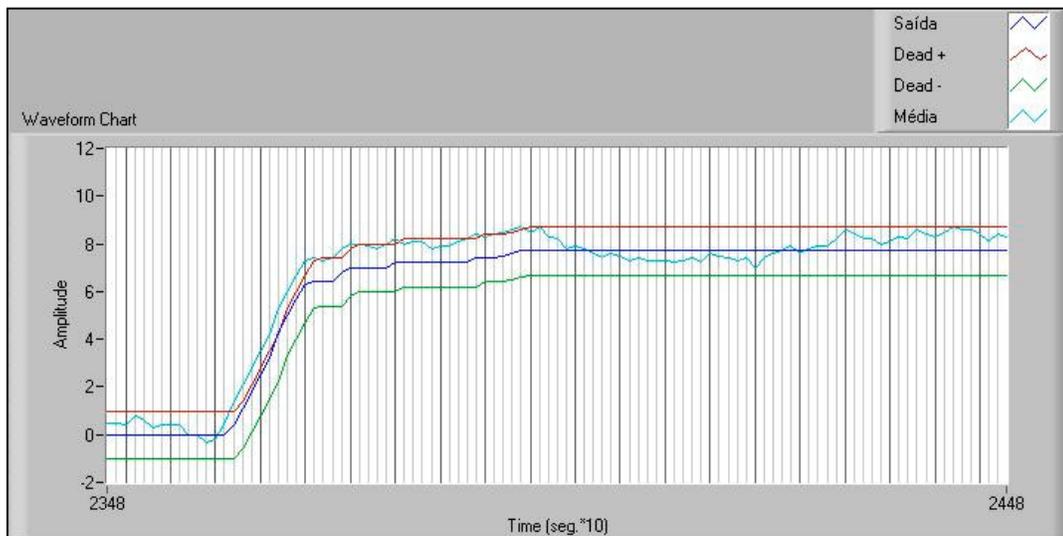


Fig. 7.6 - Filtragem fuzzy.

Na comparação entre as figuras 7.5 e 7.6 é necessário ressaltar que a curva “média” representa o mesmo intervalo de tempo em ambos os casos. Desta forma, é possível relacionar o sinal de entrada (magenta), a soma de um ruído (amarelo), a filtragem passa baixa (azul claro), e a filtragem *fuzzy* com seus limites superior (vermelho), inferior (verde) e valor central (azul escuro). Observa-se que a saída é representada pelo valor central do filtro *fuzzy* (azul escuro). Uma representação destas curvas neste mesmo período de tempo é demonstrada na figura 7.7:

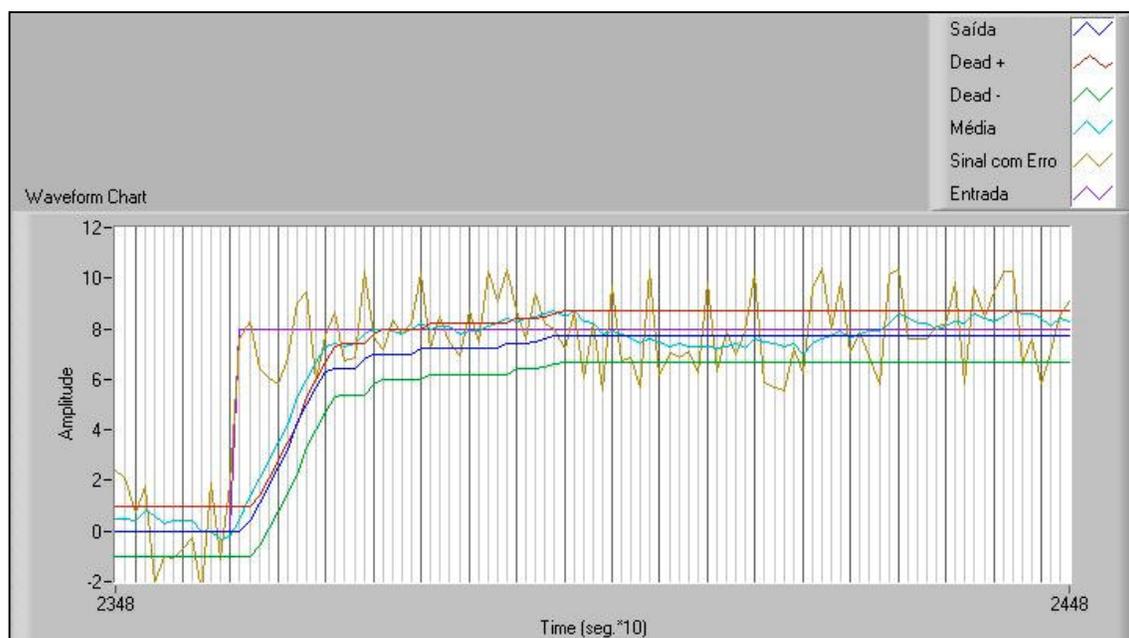


Fig. 7.7 - Curvas relacionadas a filtragem.

Inicialmente as curvas de entrada (magenta) e saída (azul escuro) são coincidentes. Após o degrau há um transiente de 1 segundo para a resposta do filtro passa baixa, seguido de 2,5 segundos para uma acomodação final relacionada ao filtro *fuzzy*. Esta segunda acomodação, relacionada ao filtro *fuzzy* não pode ser definida como uma constante de tempo, posto que não é dependente apenas de valores previsíveis, como a amplitude *fuzzy* (definida pelo usuário). O tempo gasto para a acomodação *fuzzy* também é dependente da curva resultante do filtro passa baixa (sua amplitude) e da incerteza do valor de entrada.

Os testes para a implementação desta interface em *joysticks* reais observaram uma não linearidades e fenômenos de interferências cruzadas entre seus eixos, como ilustrado na figura 7.8:

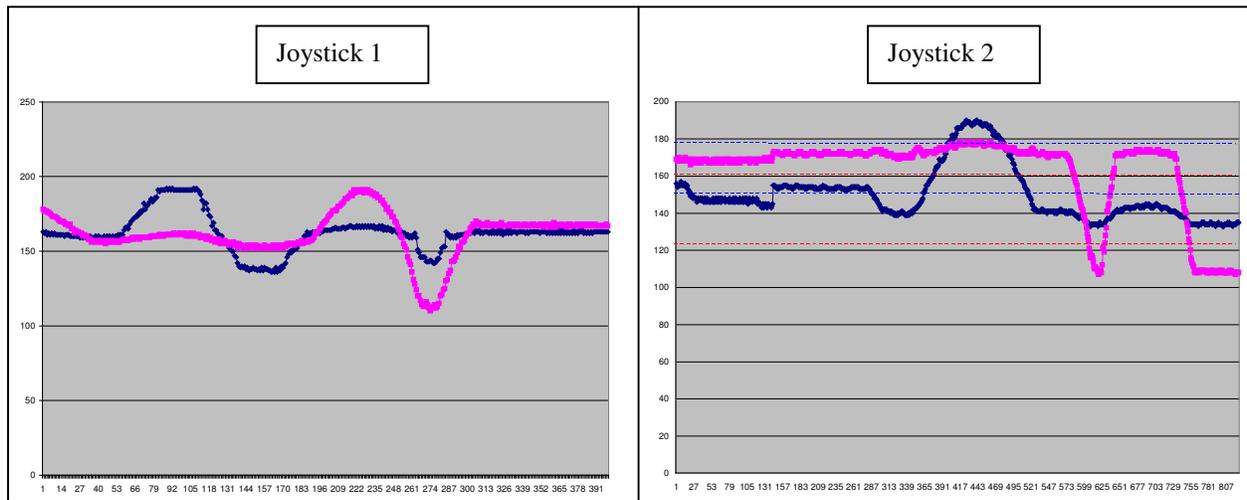


Fig. 7.8 - Testes com dois *joysticks* reais.

Legenda: Azul: eixo “x”
Magenta: Eixo “y”

Nestes testes não se utilizou nenhum processo de filtragem e moveu-se um eixo do *joystick* por vez, de uma posição extrema à outra, enquanto os dados são amostrados pelo microcontrolador em tempo real, e enviados pela porta serial do computador para plotagem dos gráficos. Em ambos os casos (testes dos dois *joysticks*) observou-se interferências entre os eixos, principalmente nas posições de deslocamento máximo.

Desta forma, na presente data deste trabalho, procura-se reconhecer e isolar as fontes destas interferências. Outra opção pode ser avaliar outros tipos de sensores para a leitura dos eixos de *joystick*, como encoders; ou alterar o circuito elétrico que obtém a leitura dos potenciômetros.

7.3. Implementação do HaWCoS em Microcontroladores PIC

Os experimentos realizados com esta interface já demonstraram ser possível controlar um kit de motorização, sem a necessidade de alterações em *hardware* ou *firmware*. Para os testes, interpretou-se como duplo clique a ocorrência de dois cliques no intervalo de 0,8 segundo. No movimento com pausas utilizou-se uma espera de 2 segundos antes de realizar o último movimento selecionado. Observou-se que o tempo entre dois cliques (0,8 segundo) difere em função do usuário, de modo que testes preliminares demonstraram que alguns usuários demandam um tempo maior para realizar o duplo clique.

Para a otimização destes tempos (espera para movimentos e duplo clique), pode ser possível utilizar a técnica de modelamento dos tempos de resposta MHP, proposta por Keates *et al.* (1999) e Card *et al.* (1983). As colaborações com o pesquisador Torsten Felzer demonstraram que a técnica de controle com pausas evita alterar o *layout* de comandos, embora possa resultar em um tempo de espera indesejado entre selecionar uma direção e movimentar-se. Desta forma, ambas as implementações (com ou sem pausas) estão presentes e podem ser selecionadas pelo usuário quando necessário.

7.4. Integração entre Interfaces e Sistemas Auxiliares

Para a validação da integração hierárquica de interfaces, testaram-se dois protótipos funcionais: um em escala reduzida e um segundo por meio do kit de motorização.

O protótipo em escala reduzida, também utilizado para simular os movimentos na interface de reconhecimento de voz, permitiu avaliar a confiabilidade de uma integração *subsumption* por hierarquia. Neste protótipo toda a lógica foi programada em um único microcontrolador, de modo que se associa um conjunto de sub-rotinas a cada *hardware* externo (interface ou sensor).

Assim, a retirada ou inserção de um módulo externo consiste em habilitar ou desabilitar uma determinada sub-rotina e atribuir um comportamento e uma hierarquia correspondentes. Observou-se uma grande facilidade para a programação e integração de novos módulos, contudo nesta implementação a confiabilidade deste sistema de controle é diretamente relacionada aos módulos de maior hierarquia.

Nos testes realizados com o sensor de sonar, reconhecido como uma alta hierarquia, observou-se um grande número de falsos positivos (detectar objetos inexistentes) quando fontes externas de som, quando os motores ou relês são acionados. Estes falsos positivos podem interromper a movimentação por curtos intervalos de tempo, ou nos piores casos, impossibilitar a movimentação. Deste modo, observou-se a necessidade de utilizar uma outra técnica para detecção de obstáculos, ou acrescentar um módulo comportamental para negar a detecção de objetos em situações de falsos positivos.

Os testes realizados com o kit de motorização permitiram avaliar os aspectos modulares desta arquitetura *subsumption* integrada por hierarquia. Nestes testes utilizaram-se os módulos de HaWCoS embarcado, *joystick*, sonar e adaptador de nível de tensão, ilustrados na figura 7.9.



Fig. 7.9 - Módulos utilizados no kit de motorização.

A inserção ou retirada de módulos não necessita a reprogramação de uma central ou de nenhum outro sistema envolvido, e a conectividade entre estes elementos é demonstrada na figura 7.10, Para favorecer a compreensão destes módulos, é possível observar a disposição dos circuitos no apêndice V.

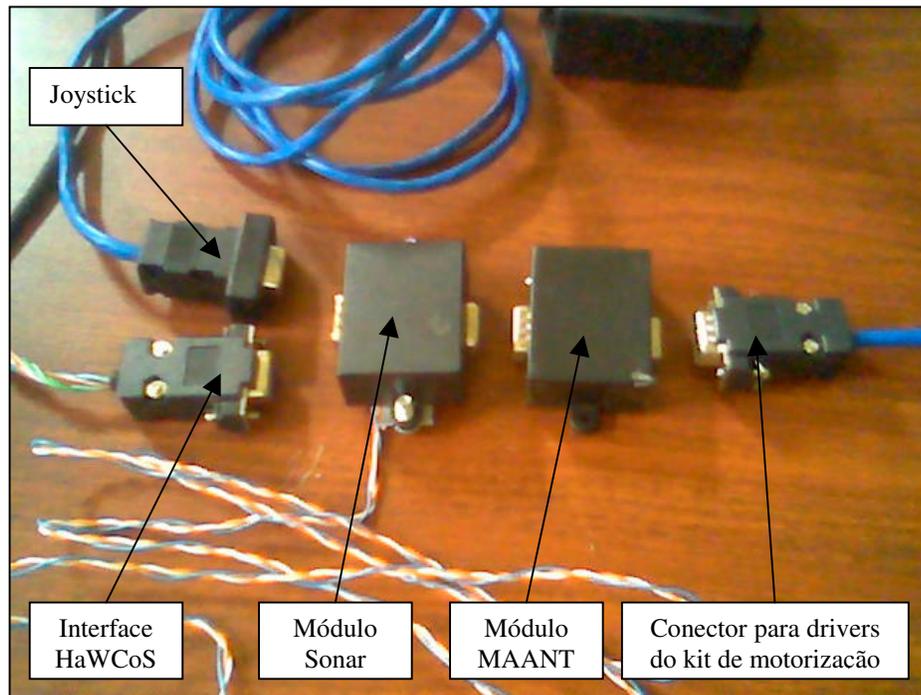


Fig. 7.10 - Conectividade entre módulos.

Os testes de conectividade relacionaram 5 opções:

- *Joystick* → *Driver*: Configuração original do kit de motorização
- *Joystick* → *MAANT** → *Driver*: Para testar MAANT
- *HaWCoS* → *MAANT** → *Driver*: Para usuários com limitações nas mãos
- *Joystick* → *Sonar (para trás)* → *MAANT** → *Driver*: Para usuários com dificuldade em manobrar
- *HaWCoS* → *Sonar (para trás)* *MAANT** → *Driver*: Para usuários com dificuldade em manobrar

Obs: * Módulo Adicional para Adaptar de Níveis de Tensão

Hierarquia menor → Hierarquia maior

A intervenção do módulo sonar ocorre durante todo o intervalo de tempo em que há um objeto detectado.

Os testes demonstraram ser possível alternar entre um *joystick* e uma interface HaWCoS sem a necessidade de mudança nos circuitos de *driver*, e mantendo a consistência nos comandos (não altera-se a direção de movimentos). Também foi possível testar a inserção de um sonar em uma hierarquia superior a da interface, e observar a capacidade de desabilitar direções de movimento quando objetos são detectados.

Capítulo 8

Conclusões e Planos Futuros

O sistema de reconhecimento de voz por microcontrolador utilizado já permite a vocalização de comandos por meio de palavras ou sons, contudo ainda não possui uma boa margem de acerto. Implementações futuras devem estudar métodos mais sofisticados para a análise do sinal de voz, respeitando as limitações do microcontrolador, para aumentar o índice de acerto.

Após a identificação de soluções para os problemas relacionados a não linearidade e interferências cruzadas nos eixos do *joystick*, será possível implementar o algoritmo para filtragem de movimentos involuntários em *joysticks* (descrito nos tópicos 6.2 e 7.2) por meio de eletrônica embarcada em microcontroladores.

Os testes realizados com o sistema HaWCoS desenvolvido em microcontroladores PIC, já demonstram ser possível controlar um pequeno protótipo e um kit de motorização de uma cadeira de rodas por meio de sensores piezoelétricos. As pesquisas futuras podem abordar os tempos de resposta dos usuários, permitindo definir melhor o período para dupla contração e para a espera antes de se movimentar. Estes períodos se associam diretamente as capacidades motora, cognitiva e de percepção de cada usuário.

A técnica de integração modular implementada no kit de motorização demonstrou-se amigável, pois não é necessário reprogramar nenhum dos módulos em sua inserção ou retirada. Contudo foi observado que, para uma integração *subsumption* por hierarquia, é fundamental que as hierarquias superiores possuam alta confiabilidade, garantindo a estabilidade do sistema como um todo.

Abordando o sistema de integração desenvolvido como uma arquitetura *subsumption*, observa-se que os níveis de maior autonomia, seguindo a classificação de Rasmussen (1983), estão associados ao usuário. Pesquisas futuras nesta integração entre módulos deverão incluir comportamentos relacionados a “fugir” de obstáculos, além do comportamentos já existente de evitar colisão. Estes módulos comportamentais permitirão camadas mais elevadas de autonomia para o sistema de auxílio a navegação.

O desenvolvimento de novas soluções em interfaces, sistemas de apoio e a sua integração modular permitirão um ganho significativo para a mobilidade e independência de portadores de necessidades especiais. Estes ganhos também afetam de forma indireta as pessoas com quem estes portadores de necessidades especiais convivem, garantindo um melhor convívio social e qualidade de vida.

Capítulo 9

Bibliografia

- Abellard, A., Khelifa, M. M. B., A Petri Net Modelling of a Neural Human Machine Interface, *IEEE International Conference on Industrial Technology*, pp. 949 - 954, 2004.
- Adachi, Y., Kuno, Y., Shimada, N., Shirai, Y., Intelligent Wheelchair Using Visual Information on Human Faces, Canada, *Proceedings of IEEE/RSJ*, 1998.
- Agostini, Y., Bourhis, G., Human-oriented architecture for an intelligent powered wheelchair, Vienna, Austria, *IFAC Worhhop DARS'95*, Sept. 19-20, 1995, pp. 127-132.
- Alhmana, J., *Modelisation par Réseaux de Petri à Flux de Données. Application à la synthèse de l'opérateur de Riccati rapide*, Université de Marseille Tese, 1983, 111p. Tese (Doutorado).
- Alvarenga, Flávia Bonilha, *Desenvolvimento de Sistemas de Motorização Alternativa para Cadeira de Rodas Convencionais*, UNICAMP Universidade Estadual de Campinas, 2002, Tese (Mestrado).
- Alvarenga, Flávia Bonilha, *Uma Abordagem Metodológica para o Projeto de Produtos Inclusivos*, UNICAMP Universidade Estadual de Campinas, 2006, Tese (Doutorado).
- Aubrey, E., The Lost History Of The Wheelchair, *on-line*, <http://www.63s.org>, acesso em 6 julho 2007.
- Becker, Marcelo, *Estudo sobre robôs de locomoção: formas construtivas, dirigibilidade e controle*, UNICAMP Universidade Estadual de Campinas, 1997, Tese (Mestrado).
- Becker, Marcelo, *Aplicação de tecnologias assistivas e técnicas de Controle em Cadeiras de Rodas Inteligentes*, UNICAMP Universidade Estadual de Campinas, 2000, Tese (Doutorado).
- Becker, M., Dantas, C. M., Macedo, W. P., Obstacle Avoidance Procedure for Mobile Robots, *Proceedings of Cobem 2005*, pp. 250 - 257, 2005.

- Bilmes, J. A., Li, X., Malkin, J., Kilanski, K., Wright, R., Kirchhoff, K., Subramanya, A., Harada, S., Landay, J. A., Dowden, P., Chizeck, H., The Vocal Joystick Demo at UIST05: A Voice-Based Human-Computer Interface, *National Science Foundation*, Grant No. IIS-0326382, 2006.
- Blessing, L.T.M., Chakrabarti, A., Wallace, K.M., A Design Research Methodology, Prague, Czech Republic, *Proceedings of the International Conference on Engineering Design (ICED)*, 1995, pág. 502-507.
- Bourhis, G., Agostini, Y., Man-machine Cooperation for the Control of an Intelligent Powered Wheelchair, *Journal of Intelligent and Robotic Systems*, pp. 269 - 287, 1998a.
- Bourhis, G., Agostini, Y., The Vahm Robotized Wheelchair: System Architecture and Human-Machine Interaction, *Journal of Intelligent and Robotic Systems*, pp. 39 – 50, 1998b.
- Braitenberg, V., *Vehicles: experiments in synthetic psychology*. MIT Press, Cambridge, MA. 1984.
- Britannica Concise, Photograph: Genghis, *on line*, <http://concise.britannica.com/ebc/art-56152/Genghis-the-robot-Genghis-was-built-at-MIT-in-the>, Acesso em 12 jan. 2008.
- Brooks, R. A., A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2, 1986.
- Buhler, C., Robotics for Rehabilitation - a European Perspective, Bath, UK, *Proceedings of the International Conference on Rehabilitation Robotics (ICORR) '97*, 1997, pp. 5-11.
- Card, S. K., Moran, T. P., Newell A., *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, 1983.
- Chapin, J. K., Moxon, K. A., Markowitz, R. S., Nicolelis, M. A. L., Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex, *Nature America*, Articles vol. 2 n°7, pp. 664 - 670, 25 maio 1999.
- CPU-Z, Software para diagnóstico de hardware, *on line*: www.cpuz.com. Acesso em jun. 2007
- Cunningham, S., Medical "Sip and Puff" Device, *catálogo on-line parallax*, disponível em http://www.parallax.com/html_pages/resources/custapps/app_medical_sip_puff_device.asp. Acesso 23 jan. 2007.
- De Mauro, C., Gori, M., Maggini, M., Martinelli, E., A voice device with an application-adapted protocol for microsoft windows, Firenze, Italy, *Proceedings IEEE Int. Conf. on Multimedia Comp. and Systems*, 1999, volume II, pp. 1015 - 1016.
- Dix, A. J., Finlay, J. E., Abowd, G. D., Russel, B., *Human-Computer Interaction*, Prentice Hall Europe, 1998.

- Engelbart, D., Did you know?, MIT, *on-line*, disponível em http://web.mit.edu/invent/www/ima/engelbart_draw.html. Acesso em 7 jan. 2007.
- Felzer, T., Freisleben, B., HaWCoS: The "Hands-free" Wheelchair Control System, Edinburgh, Scotland, *Assets 2002*, July 8-10.
- Felzer, T., Nordmann, R., Alternative Wheelchair Control, *Darmstadt University of Technology*, Germany, 2007.
- Fiorioni, P., Shiller, Z., Motion Planning in Dynamic Environments using the Relative Velocity Paradigm, *Proceedings of the 1993 IEEE International Conference on Robotics & Automation*, Atlanta, 1993.
- Gasson, M., Hutt, B., Goodhew, I., Kyberd, P., Warwick, K., Bidirectional Human Machine Interface Via Direct Neural Connection, Berlin, Alemanha, *Proceedings of IEEE*, 2002.
- Gips, J., On Building Intelligence into EagleEyes, *Computer Science Department Boston College Chestnut Hill*, pp. 50 - 58, 1998.
- Gips, J., Olivieri, P., EagleEyes: An Eye Control System for Persons with Disabilities, *The Eleventh International Conference on Technology and Persons with Disabilities*, Los Angeles, California, Março 1996.
- Gomi, T., Griffith, A., Developing Intelligent Wheelchairs for the Handicapped, *Assistive Technology and AI*, LNAI 1458, pp. 150-178, 1998.
- Graham, E. D., MacKenzie, C. L., Physical Versus Virtual Pointing, Vancouver, Canada, *Proceedings of CHI '96*, 1996, pp. 292-299.
- Graham-Rowe, D., Half fish, half robot, *New Scientist*, 10 Junho 2000.
- Harter, M. R., Excitability and Cortical Scanning: A Review of two Hypotheses of Central Intermittency in Perception, *Psychological Bulletin*, 68, pp. 47-58, 1967.
- Harvey, I., Evolutionary robotics and SAGA: the case for hill crawling and tournament selection. C. Langton, *Artificial Life 3 Proceedings*, ed., Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol XVI, 1992, pp. 299-326.
- Holland, J. M., *Designing Autonomous Mobile Robots*, Newnes, 2004.
- Igarashi, T., Hughes, J. F., Voice as sound: Using nonverbal voice input for interactive control. *ACM UIST 2001*, nov. 2001.
- International Component Corporation, Linha "Intervox", *catálogo on-line* http://www.icc107.com/inter_productover.cfm, Acesso em 22 out 2007.

- Jacob, R.J.K., The Use of Eye Movements in Human-Computer Interaction Techniques:What You Look at Is What You Get. *ACM Transactions on Information*, pp. 152-169, 1991.
- Jácomo, Justo Emilio Alvarez, *Desenvolvimento de um Robô Autônomo Móvel Versátil utilizando Arquitetura Subsumption*, UNICAMP Universidade Estadual de Campinas, 2001, 121p. Tese (Mestrado).
- James, Information About Deafblindness - Refreshable Braille Displays, *on-line* <http://www.deafblind.com/display.html>. Acesso 23 out. 2007.
- Karshmer, A.I., Interdisciplinary Efforts to Facilitate the Production of Tools to Support the Disabled and Elderly in the Information Society, San Francisco, USA, *Proceedings of HCI International*, 1997, pp. 409-412.
- Keates, S., Clarkson, P. J., Robinson, P., Developing a Methodology for the Design of Accessible Interfaces, *University of Cambridge*, mai. 1999.
- Kennedy, P. R., Bakay, R. A. E., Moore, M. M., Adams, K. Goldwaithe, J., Direct Control of a Computer from the Human Central Nervous System, *Proceedings of IEEE*, VOL. 8, NO. 2, Junho 2000, pp. 198-202.
- Kirkup, L., Searle, A., Craig, A., McIsaac, P., Moses, P., EEG-Based system for rapid on-off switching without prior learning, *Medical & Biological Engineering & Computing*, 35, 504-509, 1997.
- Laurel, B., *The Art of Human Computer Interface Design*, Addison-Wesley Publishing Company, 1994.
- Law, C. K. H., Leung, M. Y. Y., Xu, Y., Tso, S. K., A cap as interface for wheelchair control, *IEEE/RSJ* Out. 2002, pp. 1439 - 1444, on-line, disponível em <http://ieeexplore.ieee.org/iel5/8071/22356/01043957.pdf>. Acesso 15 Abr. 2007.
- LC Technologies, Eyegaze Analysis System, on-line, disponível em http://www.interactive-ind.de/de/eye_tracker/eyegaze_system_allgemein.htm. Acesso em 23 jan. 2007.
- Lombardi Junior, Arley de Barros, *Desenvolvimento e Modelagem de uma Cadeira de Rodas Servo-Assistida para Crianças*, UNICAMP Universidade Estadual de Campinas, 2002, Tese (Mestrado).
- Madarasz, R. L., Heiny, L. C., Cromp, R. F., Mazur, N. M., The Design of an Autonomous Vehicle for the Disabled, *IEEE Journal of Robotics and Automation*, pp. 117-127, Set. 1986.
- Maes, P., Brooks, R. A., Learning to Coordinate Behaviors, *National Conference on Artificial Intelligence*, 1990.
- Mahadevan, S., Connell, J., Automatic Programming of Behavior-based Robots using Reinforcement Learning, *Proceedings of AAAI-91*, 1991.

- Mahoney, R., Robotic Products for Rehabilitation: Status and Strategy, Bath, UK, *Proceedings of ICORR '97*, 1997, pp. 12-22.
- Microchip, Data Sheet do microcontrolador PIC16F877, *on line* www.microchip.com. Acesso em jun. 2007.
- Microsoft, Types of Assistive Technology Products, *on-line* <http://www.microsoft.com/enable/at/types.aspx>. Acesso em 23 jan. 2007.
- Min, J. W., Kyoobin, L., Lim, S. C., Kwon, D. S., Human-friendly Interfaces of Wheelchair Robotic System for Handicapped Persons, *IEEE/RSJ, Out.* 2002, pp. 1505 – 1510.
- Nicolelis, M. A. L., Actions from thoughts, *Nature*, vol. 409, pp. 403 - 407, 2001
- Olwal, A., Feiner, S. Interaction techniques using prosodic features of speech and audio localization. New York, USA, *IUI '05: Proc. 10th Int. Conf. on Intelligent User Interfaces*, 2005, pp. 284 a 286.
- Poole, A., Ball, L. J., Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects, Psychology Department, *Lancaster University*, UK, on-line, disponível em <http://www.alexpoole.info/academic/Poole&Ball%20EyeTracking.pdf>. Acesso em 16 jan. 2007.
- Prassler, E., Scholz, J., Fiorini, P., MAid: A Robotic Wheelchair Roaming in a Railway Station, *Parte do projeto INSERVUM*, 1999.
- Prassler, E., Scholz, J., Strobel, M., MAid: Mobility Assistance for Elderly and Disabled People, *IEEE*, 1998.
- Preece, J., Rogers, Y., Sharp, H., *Interaction Design: Beyond Human-Computer Interaction*, Wiley, 2002.
- Rasmussen, J., Skills, rules and knowledge; signals, signs and symbols, and other distinctions in human performance models, *IEEE Transactions on Systems, Man and Cybernetics*, 1983, pp. 257 – 266.
- Ross, D., Personal freedom: a wearable interactive universal access device, *Proceedings of the RESNA '96 Annual Conference*, Salt Lake City, Utah, junho de 1996, pp. 450-452.
- Ross, D. A., Sanford, J. A., Wearable Computer as a Remote Interface for People with Disabilities, *IEEE* 1997.
- Shaw, R., Crisman, E., Loomis, A., Laszewski, Z., The eye wink control interface: Using the Computer to Provide the Severely Disabled with Increased Flexibility and Comfort, *IEEE*, 1990, pp. 105 – 111.

- Siegwart, R., Nourbakhsh, I. R., *Introduction to Autonomous Mobile Robots*, Bradford Book, 2004.
- Simon, H. A., *Sciences of the Artificial*. Cambridge, MA: *M.I.T.*, 1969.
- Simpson, R. C., Levine, S. P., Bell, D. A., Jaros, L. A., Koren, Y., Borenstein, J., NavChair: An Assistive Wheelchair Navigation System with Automatic Adaptation, *Assistive Technology and AI*, LNAI 1458, pp. 235-255, 1998.
- Speropoulos, C. A., Townson, A. F., Assistive Technology Assessments in a Tertiary-care Rehabilitation Facility, *BC Medical Journal*, Volume 43, Number 2, pp. 82-85, março 2001.
- STMicroelectronics, Data sheet do LM7805, *on line*, <http://www.st.com>. Acesso em jul. 2007.
- Talwar, S. K., Xu S., Hawley, E. S., Weiss, S. A., Moxon, K. A., Chapin, J. K., Rat Navigation Guided by Remote Control, *Nature*, vol. 417, pp. 37-38, 2 maio 2002.
- Taylor, P. B., Nguyen, H. T., Performance of a Head-Movement Interface for Wheelchair Control, *IEEE EMBS*, pp. 1590 – 1593, Set. 2003.
- Tellex, S., Relational Interface for a Voice Controlled Wheelchair, *on-line*, <http://ocw.mit.edu/NR/rdonlyres/Media-Arts-and-Sciences/MAS-965Spring-2005/2801B076-BCAF-4BDC-8722-F020DA6D8683/0/tellexfinalproj.pdf> Acesso em 30 jan. 2007.
- University of Pittsburgh, Wheeling in the New Millennium: The history of the wheelchair and the driving forces in wheelchair design today, *Department of Rehabilitation Science and Technology*, 2002, *on-line*, http://www.wheelchairnet.org/WCN_WCU/SlideLectures/Sawatzky/WC_history.html, Acesso 6 Julho 2007.
- USP, Implante Coclear, *on-line* http://www.otorrinousp.org.br/imageBank/seminarios/seminario_1.pdf. Acesso em 23 out. 2007.
- Wolpaw, J. R., McFarland, D. J., Neat, G. W. & Forneris, C. A., Electroencephalography, *Europhysiol.*, vol. 78, pp.252-259 1991.
- Yanco, H. A. , Wheellesley: A Robotic Wheelchair System: Indoor Navigation and User Interface, *Assistive Technology and AI*, pp. 256 - 268, 1998a, *on-line* <http://www.cs.uml.edu/~holly/papers/ai-at-yanco.pdf>.
- Yanco, H. A., Integrating Robotic Research: A Survey of Robotic Wheelchair Development, California, *AAAI Spring Symposium on Integraing Robotic Research*, 1998b.
- Yanco, H. A., Gips, J., Preliminary Investigation of a Semi-Autonomous Robotic Wheelchair Directed Through Electrodes, *Proceedings of the Rehabilitation Engineering Society of North America Annual Conference*, Jun. 1997, pp. 414 – 416.

Apêndice I

Histórico de Cadeiras de Rodas

A cadeira de rodas é um sistema de mobilidade para usuários sentados em caso de dificuldades ou incapacidade para locomover-se. Esta mobilidade pode ser realizada de modo manual ou por intermédio de diversas formas de locomoção automatizada.

Segundo Aubrey (2007), seu surgimento provém da diversidade de aplicações da roda como o seu emprego para transporte de objetos em (4000 A.C.) e em carruagens (1300 A.C.), segundo Wikipedia (2007). As primeiras documentações de equipamentos com características semelhantes a uma cadeira de rodas datam de 530 a.C. em um vaso Grego, e em 525 d.C. em uma gravura chinesa. Estes equipamentos tratam-se de uma maca com rodas para crianças (gregos) e uma cadeira com rodas para carregada por servos (chineses) como ilustrado na figura I.1:



Fig. I.1 - Maca grega e cadeira com rodas chinesa (fonte: University of Pittsburgh, 2002).

Segundo Aubrey (2007), outro invento mais sofisticado era utilizado pelo rei espanhol Philip em 1595. Esta cadeira possuía rodas e ajustes de altura, sendo carregada pelos seus servos da para realizar caminhos domésticos e percorrer os jardins do palácio, como ilustrado na figura I.2:



Fig. I.2 - Cadeira de rodas utilizada pelo rei Philip II (fonte: University of Pittsburgh, 2002).

Segundo Aubrey (2007), o primeiro invento movido pela força do usuário é documentado de 1665 e possui como inventor um relojoeiro alemão, Stephan Farfler aos 22 anos de idade. Este invento assemelha-se muito a uma bicicleta movida pela mão, no qual um par de manivelas move a roda dianteira, também utilizada para a direção, como ilustrado na figura I.3:



Fig. I.3 - Cadeira de Stephan Farfler (fonte: University of Pittsburgh, 2002).

Em 1783, John Dawson também conhecido como “o construtor de cadeiras de rodas” (*Wheelchair Maker*), inventou um equipamento com ajustes para pernas e costas. Esta cadeira de rodas é movida ao puxar de rodas pelo usuário e dominou o mercado até o século 19, como ilustrado na figura I.4:

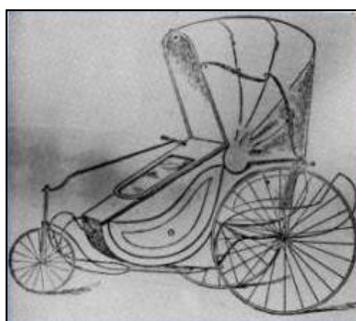


Fig. I.4 - Cadeira de John Dawson (fonte: University of Pittsburgh, 2002).

Segundo Aubrey (2007), possuindo um maior foco no conforto do usuário, desenvolveu-se a cadeira *seating wheelchair* no século 18. Este modelo permite ajustes de inclinação e se parece com uma cadeira residencial de luxo, mas com duas grandes rodas de madeira, como ilustrado na figura I.5:

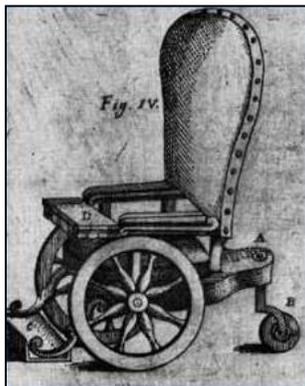


Fig. I.5 - Cadeira modelo *seating wheelchair* (fonte: University of Pittsburgh, 2002).

Um outro modelo, referenciado University of Pittsburgh (2002) é ilustrado na figura I.6, como cadeira leve utiliza bambu indiano na sua construção, pesando entre 22,68 kg e 26,31 kg.



Fig. I.6 - Cadeira com foco em redução de peso (fonte: University of Pittsburgh, 2002).

Segundo University of Pittsburgh (2006), em 1932 Herbert Everest (um engenheiro de minas deficiente) e Harold Jennings (engenheiro mecânico), desenvolveram uma cadeira de rodas dobrável. Esta cadeira, ilustrada na figura I.7, permitiu a fundação da companhia Everest & Jennings ou E&J, também responsável pela criação da primeira cadeira de roda motorizada.



Fig. I.7 - Cadeira de Everest & Jennings (fonte: University of Pittsburgh, 2002).

Em 1934, Samuel Duke desenvolveu o segundo modelo dobrável de cadeira de rodas. Nos anos seguintes até os dias atuais novas melhorias foram realizadas, permitindo um maior conforto, capacidades de dobrar, maiores velocidades, otimizações na suspensão e aparência.

Apêndice II

Tabelas com o levantamento das curvas de freqüência utilizadas

É possível observar os valores amostrados no processo de treinamento da biblioteca de palavras nas tabelas a seguir. Desta forma, obtém-se a média de bordas de subida em cada coluna para a curva representativa. Por meio da média de bordas de subida é possível obter o desvio de cada amostra ao subtrair-se o número de subdas da amostra em relação à curva representativa. É necessário observar que os desvios apresentados nestas tabelas representam uma curva inteira e não uma comparação entre os intervalos, esta segunda é realizada para teste de casamento.

Tabela II.1 – Amostragem da palavra Frente

Frente	Intervalo 1	Intervalo 2	Intervalo 3	Intervalo 4	Intervalo 5	Intervalo 6	Intervalo 7	Intervalo 8	Intervalo 9	Subidas	Desvio
Amostra 1	19	19	4	9	0	0	0	0	0	51	6,0714286
Amostra 2	16	14	5	12	0	0	0	0	0	47	2,0714286
Amostra 3	12	16	7	11	0	0	0	0	0	46	1,0714286
Amostra 4	18	16	4	10	0	0	0	0	0	48	3,0714286
Amostra 5	16	14	4	8	0	0	0	0	0	42	-2,9285714
Amostra 6	16	17	4	9	0	0	0	0	0	46	1,0714286
Amostra 7	16	14	4	7	0	0	0	0	0	41	-3,9285714
Amostra 8	15	12	7	10	0	0	0	0	0	44	-0,9285714
Amostra 9	14	12	3	11	0	0	0	0	0	40	-4,9285714
Amostra 10	12	15	5	11	0	0	0	0	0	43	-1,9285714
Amostra 11	12	12	5	12	0	0	0	0	0	41	-3,9285714
Amostra 12	15	15	2	13	0	0	0	0	0	45	0,0714286
Amostra 13	14	13	6	13	0	0	0	0	0	46	1,0714286
Amostra 14	18	12	7	10	2	0	0	0	0	49	4,0714286
Média	15,21429	14,35714	4,785714	10,42857	0,142857	0	0	0	0	44,92857	
Testando	Intervalo 1	Intervalo 2	Intervalo 3	Intervalo 4	Intervalo 5	Intervalo 6	Intervalo 7	Intervalo 8	Intervalo 9	Subidas	Desvio

Tabela II.2 – Amostragem da palavra Testando

Testando	Intervalo 1	Intervalo 2	Intervalo 3	Intervalo 4	Intervalo 5	Intervalo 6	Intervalo 7	Intervalo 8	Intervalo 9	Subidas	Desvio
Amostra 1	20	4	9	11	13	9	0	0	0	66	-6,3846154
Amostra 2	19	6	16	11	12	8	0	0	0	72	-0,3846154
Amostra 3	25	8	11	10	11	9	0	0	0	74	1,6153846
Amostra 4	20	2	12	10	15	10	1	0	0	70	-2,3846154
Amostra 5	21	7	10	12	12	7	0	0	0	69	-3,3846154
Amostra 6	28	4	12	13	15	10	0	0	0	82	9,6153846
Amostra 7	22	4	11	10	12	11	0	0	0	70	-2,3846154
Amostra 8	22	3	11	11	12	13	1	0	0	73	0,6153846
Amostra 9	19	4	8	10	12	13	4	0	0	70	-2,3846154
Amostra 10	21	5	11	10	13	11	0	0	0	71	-1,3846154
Amostra 11	24	3	9	11	11	14	3	0	0	75	2,6153846
Amostra 12	19	2	11	11	11	14	2	0	0	70	-2,3846154
Amostra 13	22	6	11	11	14	12	3	0	0	79	6,6153846

Tabela II.3 – Amostragem da palavra Esquerda

Esquerda	Intervalo 1	Intervalo 2	Intervalo 3	Intervalo 4	Intervalo 5	Intervalo 6	Intervalo 7	Intervalo 8	Intervalo 9	Subidas	Desvio
Amostra 1	22	14	11	12	7	0	0	0	0	66	1,8571429
Amostra 2	21	15	13	13	4	0	0	0	0	66	1,8571429
Amostra 3	13	17	10	11	11	0	0	0	0	62	-2,1428571
Amostra 4	23	11	11	12	5	0	0	0	0	62	-2,1428571
Amostra 5	25	15	10	11	6	0	0	0	0	67	2,8571429
Amostra 6	26	4	10	12	12	7	0	0	0	71	6,8571429
Amostra 7	25	16	10	11	4	0	0	0	0	66	1,8571429
Amostra 8	23	12	10	11	5	0	0	0	0	61	-3,1428571
Amostra 9	25	1	10	10	12	7	0	0	0	65	0,8571429
Amostra 10	22	1	11	11	12	2	0	0	0	59	-5,1428571
Amostra 11	21	2	17	10	11	7	0	0	0	68	3,8571429
Amostra 12	25	14	11	12	7	0	0	0	0	69	4,8571429
Amostra 13	22	10	10	12	4	0	0	0	0	58	-6,1428571
Amostra 14	21	2	10	10	11	4	0	0	0	58	-6,1428571
Média	22,42857	9,571429	11	11,28571	7,928571	1,928571	0	0	0	64,14286	

Tabela II.4 – Amostragem da palavra Direita

Direita	Intervalo 1	Intervalo 2	Intervalo 3	Intervalo 4	Intervalo 5	Intervalo 6	Intervalo 7	Intervalo 8	Intervalo 9	Subidas	Desvio
Amostra 1	21	16	10	8	4	0	0	0	0	59	-13
Amostra 2	16	18	12	10	11	0	0	0	0	67	-5
Amostra 3	17	24	11	8	2	1	0	0	0	63	-9
Amostra 4	21	23	12	10	1	1	0	0	0	68	-4
Amostra 5	19	21	12	10	9	0	0	0	0	71	-1
Amostra 6	20	19	17	8	4	0	0	0	0	68	-4
Amostra 7	13	24	19	10	4	9	2	0	0	81	9
Amostra 8	18	21	22	9	7	0	0	0	0	77	5
Amostra 9	17	23	21	10	1	1	0	0	0	73	1
Amostra 10	13	23	17	10	6	2	0	0	0	71	-1
Amostra 11	16	24	26	15	9	0	0	0	0	90	18
Amostra 12	15	23	14	10	1	7	0	0	0	70	-2
Amostra 13	13	22	13	11	5	10	0	0	0	74	2
Amostra 14	14	23	19	10	4	10	0	0	0	80	8
Amostra 15	13	24	13	10	1	4	1	0	0	66	-6
Amostra 16	13	26	15	10	3	1	0	0	0	68	-4
Amostra 17	19	26	19	10	4	0	0	0	0	78	6
Média	16,35294	22,35294	16	9,941176	4,470588	2,705882	0,176471	0	0	72	

Tabela II.5 – Amostragem da palavra Trás

Tras	Intervalo 1	Intervalo 2	Intervalo 3	Intervalo 4	Intervalo 5	Intervalo 6	Intervalo 7	Intervalo 8	Intervalo 9	Subidas	Desvio
Amostra 1	23	26	7	0	0	0	0	0	0	56	3,2857143
Amostra 2	20	33	5	0	0	0	0	0	0	58	5,2857143
Amostra 3	17	32	5	0	0	0	0	0	0	54	1,2857143
Amostra 4	22	27	4	0	0	0	0	0	0	53	0,2857143
Amostra 5	17	32	7	0	0	0	0	0	0	56	3,2857143
Amostra 6	20	26	4	0	0	0	0	0	0	50	-2,7142857
Amostra 7	20	19	4	0	0	0	0	0	0	43	-9,7142857
Amostra 8	20	28	1	0	0	0	0	0	0	49	-3,7142857
Amostra 9	25	23	6	0	0	0	0	0	0	54	1,2857143
Amostra 10	24	26	6	0	0	0	0	0	0	56	3,2857143
Amostra 11	22	25	5	0	0	0	0	0	0	52	-0,7142857
Amostra 12	19	25	6	0	0	0	0	0	0	50	-2,7142857
Amostra 13	22	24	7	0	0	0	0	0	0	53	0,2857143
Amostra 14	19	28	7	0	0	0	0	0	0	54	1,2857143
Média	20,71429	26,71429	5,285714	0	0	0	0	0	0	52,71429	

Tabela II.6 – Amostragem da palavra Liga

Liga	Intervalo 1	Intervalo 2	Intervalo 3	Intervalo 4	Intervalo 5	Intervalo 6	Intervalo 7	Intervalo 8	Intervalo 9	Subidas	Desvio
Amostra 1	16	24	15	4	0	0	0	0	0	59	1,9230769
Amostra 2	16	26	13	0	0	0	0	0	0	55	-2,0769231
Amostra 3	17	25	14	0	0	0	0	0	0	56	-1,0769231
Amostra 4	17	25	15	3	0	0	0	0	0	60	2,9230769
Amostra 5	21	23	16	0	0	0	0	0	0	60	2,9230769
Amostra 6	19	21	8	0	0	0	0	0	0	48	-9,0769231
Amostra 7	23	15	8	0	0	0	0	0	0	46	-11,076923
Amostra 8	23	21	13	0	0	0	0	0	0	57	-0,0769231
Amostra 9	20	25	16	0	0	0	0	0	0	61	3,9230769
Amostra 10	18	25	18	5	0	0	0	0	0	66	8,9230769
Amostra 11	18	24	17	1	0	0	0	0	0	60	2,9230769
Amostra 12	23	20	11	0	0	0	0	0	0	54	-3,0769231
Amostra 13	19	24	16	1	0	0	0	0	0	60	2,9230769
Média	19,23077	22,92308	13,84615	1,076923	0	0	0	0	0	57,07692	

Tabela II.7 – Amostragem da palavra Desliga

Desliga	Intervalo 1	Intervalo 2	Intervalo 3	Intervalo 4	Intervalo 5	Intervalo 6	Intervalo 7	Intervalo 8	Intervalo 9	Subidas	Desvio
Amostra 1	24	14	14	21	18	8	0	0	0	99	13,866667
Amostra 2	24	19	16	17	18	5	0	0	0	99	13,866667
Amostra 3	23	19	14	18	15	1	0	0	0	90	4,8666667
Amostra 4	19	22	7	14	17	6	0	0	0	85	-0,1333333
Amostra 5	22	18	12	15	14	11	0	0	0	92	6,8666667
Amostra 6	19	22	9	12	10	6	0	0	0	78	-7,1333333
Amostra 7	23	15	10	17	14	5	0	0	0	84	-1,1333333
Amostra 8	21	19	11	12	12	4	0	0	0	79	-6,1333333
Amostra 9	24	16	15	12	11	0	0	0	0	78	-7,1333333
Amostra 10	22	16	10	18	12	2	0	0	0	80	-5,1333333
Amostra 11	22	17	11	14	11	7	0	0	0	82	-3,1333333
Amostra 12	21	21	13	12	12	3	0	0	0	82	-3,1333333
Amostra 13	22	17	16	17	13	2	0	0	0	87	1,8666667
Amostra 14	23	16	13	19	13	0	0	0	0	84	-1,1333333
Amostra 15	21	8	17	18	12	2	0	0	0	78	-7,1333333
Média	22	17,26667	12,53333	15,73333	13,46667	4,133333	0	0	0	85,13333	

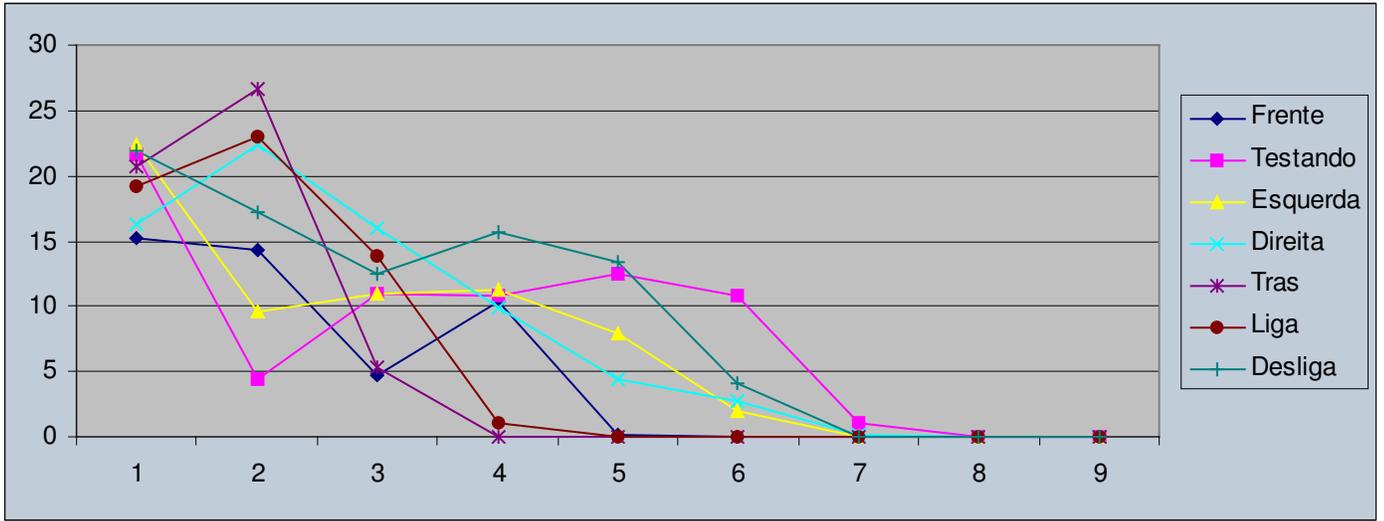


Fig. II.1 – Comparação entre curvas médias obtidas

Apêndice III

Software Labview

A figura III.1 descreve-se o código fonte do software utilizado na programação do filtro para o joystick:

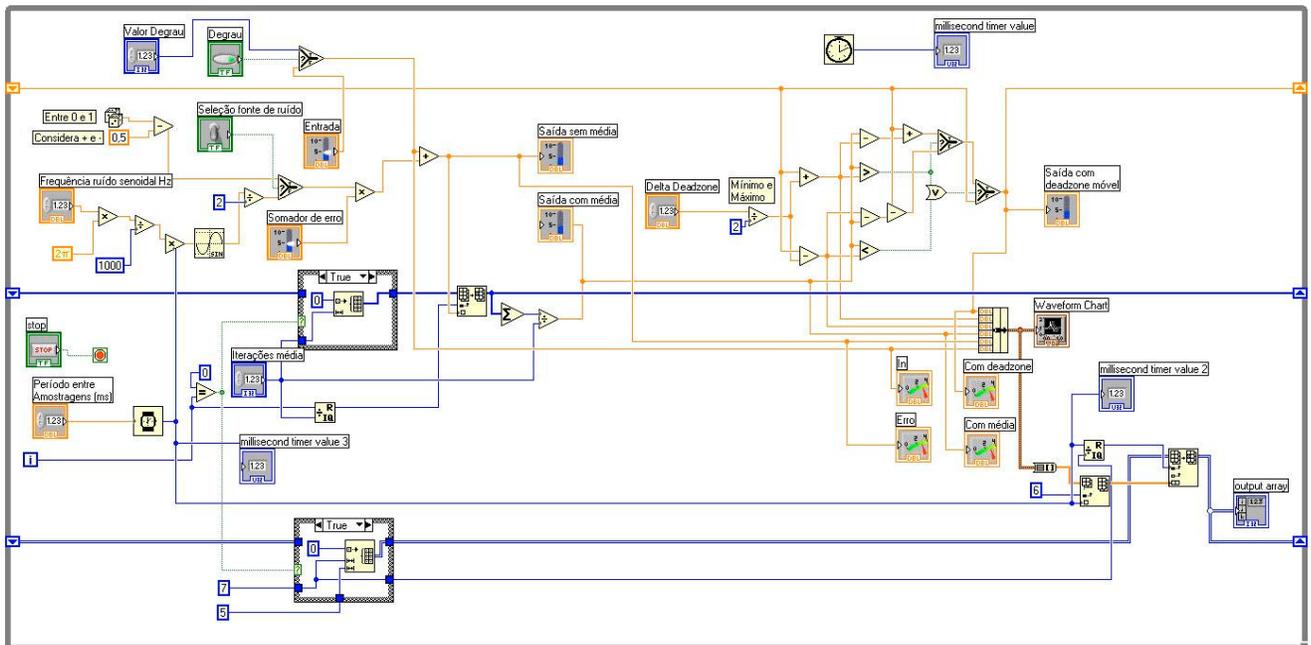


Fig. III.1 – Software em Labview

Apêndice IV

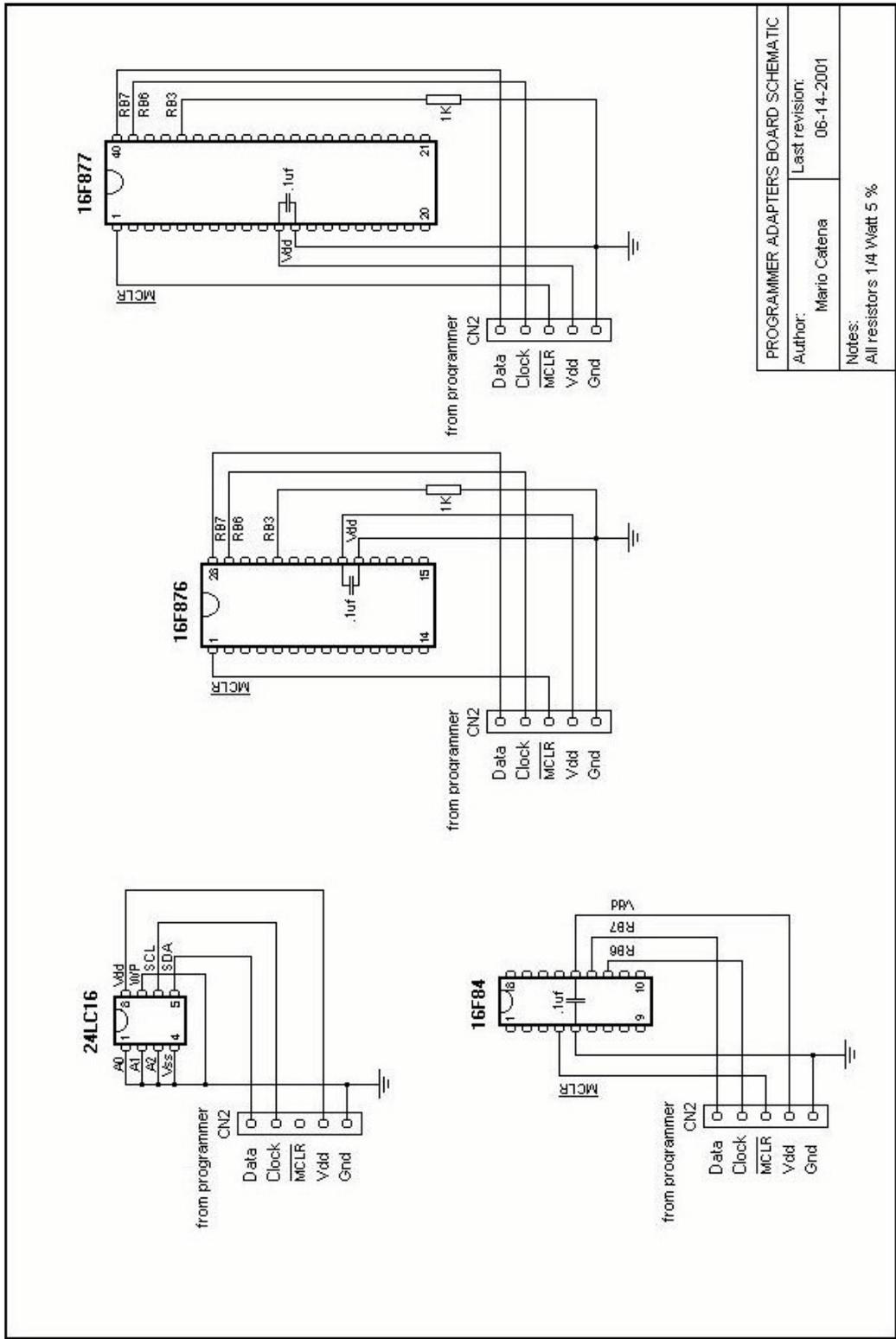
Programação do PIC 16F877A

Para a programação do microcontrolador PIC16F877A utilizaram-se quatro softwares:

- *Microchip MPLAB*: para gerar o código fonte (www.microchip.com)
- *Compilador CCS instalado no MPLAB*: para geração dos arquivos hexadecimais (www.ccsinfo.com)
- *Programador IC-Prog versão c*: para leitura dos arquivos hexadecimais e gravação no microcontrolador, permitindo utilizar a placa construída durante este projeto de mestrado (www.ic-prog.com)
- *Circuito integrado MAX RS-232*: para realizar Debug pela porta serial do microcontrolador, junto ao software Sion (incluído com o IC-Prog), que acompanha o IC-Prog. Mais detalhes sobre o MAX RS-232 podem ser encontrados no site: (www.maxim-ic.com)

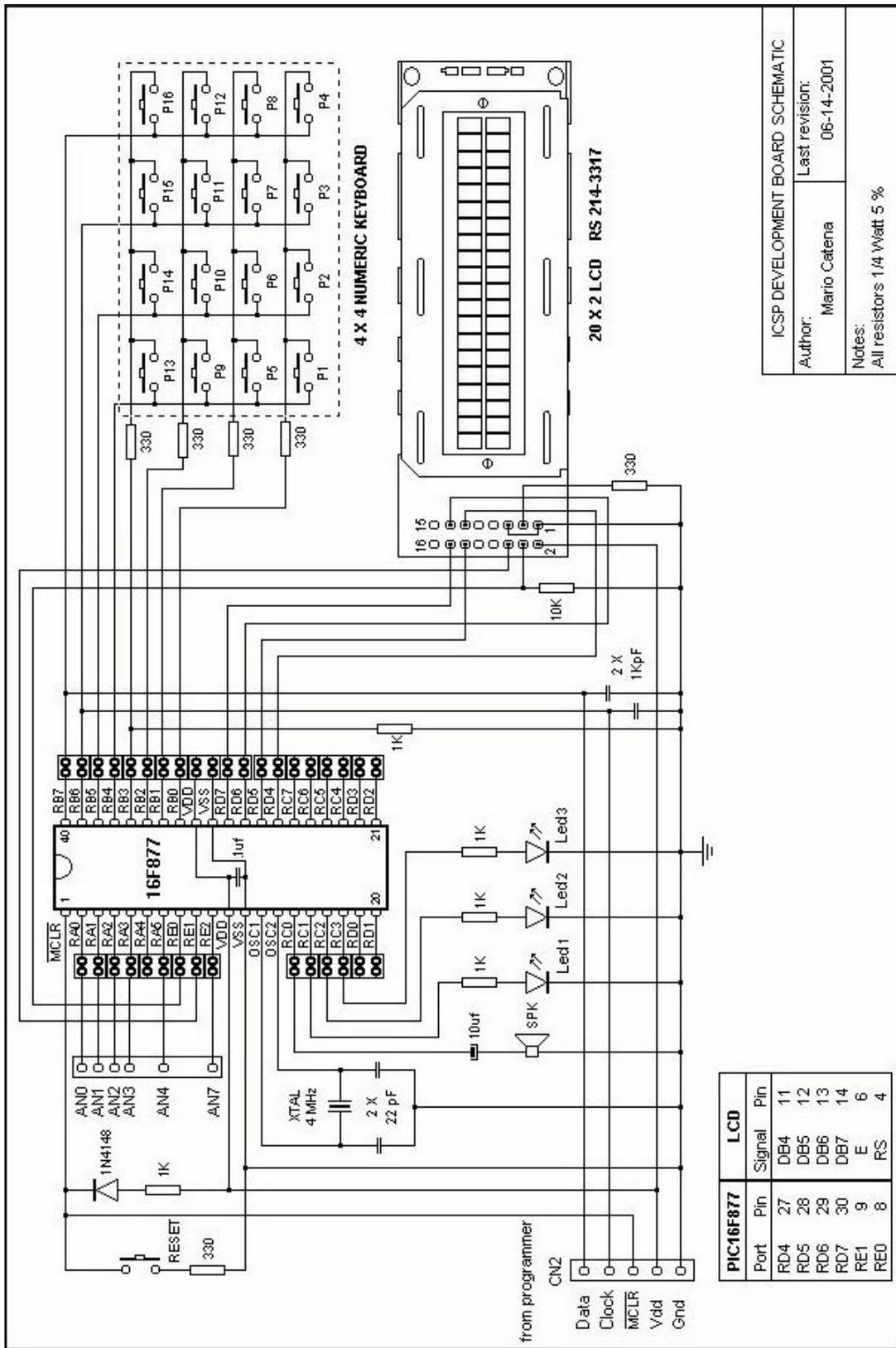
Desta forma seguem os anexos com os circuitos para confecção da placa programadora (“Taiti Serial Programmer”, utilizar com pino MCLR invertido, no IC-Prog) e da placa do microcontrolador.

Após estes circuito são apresentados em mais detalhes nas figuras D.1, D.2, D.3 e D.4 os módulos construídos para testar o protocolo de integração desenvolvido.



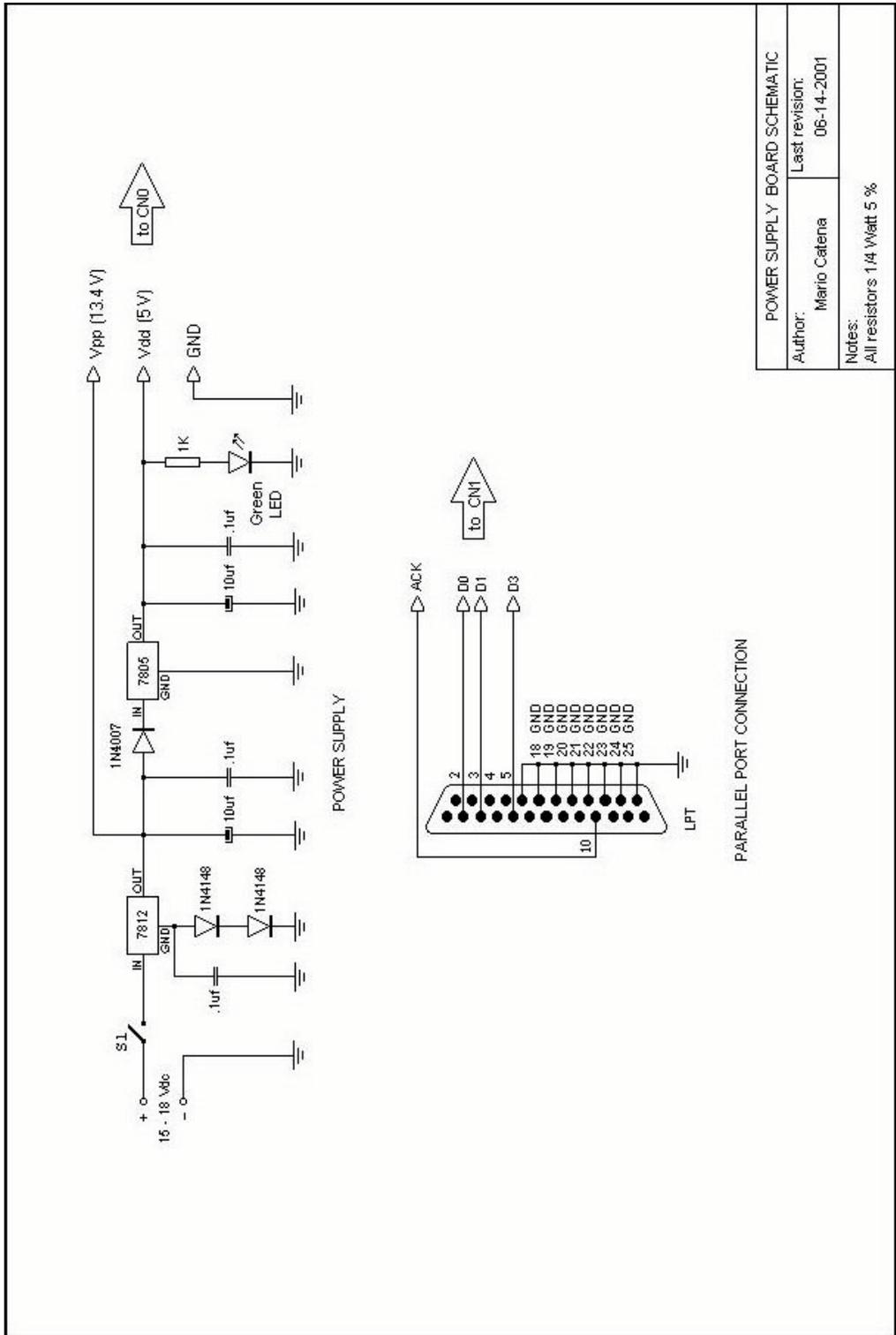
PROGRAMMER ADAPTERS BOARD SCHEMATIC	
Author:	Mario Catena
Last revision:	06-14-2001
Notes: All resistors 1/4 Watt 5 %	

Fig. IV.1 – Pinagens de microcontroladores da família 16F (www.ic-prog.com)



ICSP DEVELOPMENT BOARD SCHEMATIC
 Author: Mario Catena
 Last revision: 06-14-2001
 Notes: All resistors 1/4 Watt 5 %

Fig. IV.2 – Placa de testes para programas (www.ic-prog.com)



POWER SUPPLY BOARD SCHEMATIC	
Author:	Mario Cafena
Last revision:	06-14-2001
Notes: All resistors 1/4 Watt 5 %	

Fig. IV.3 – Fonte de alimentação (www.ic-prog.com)

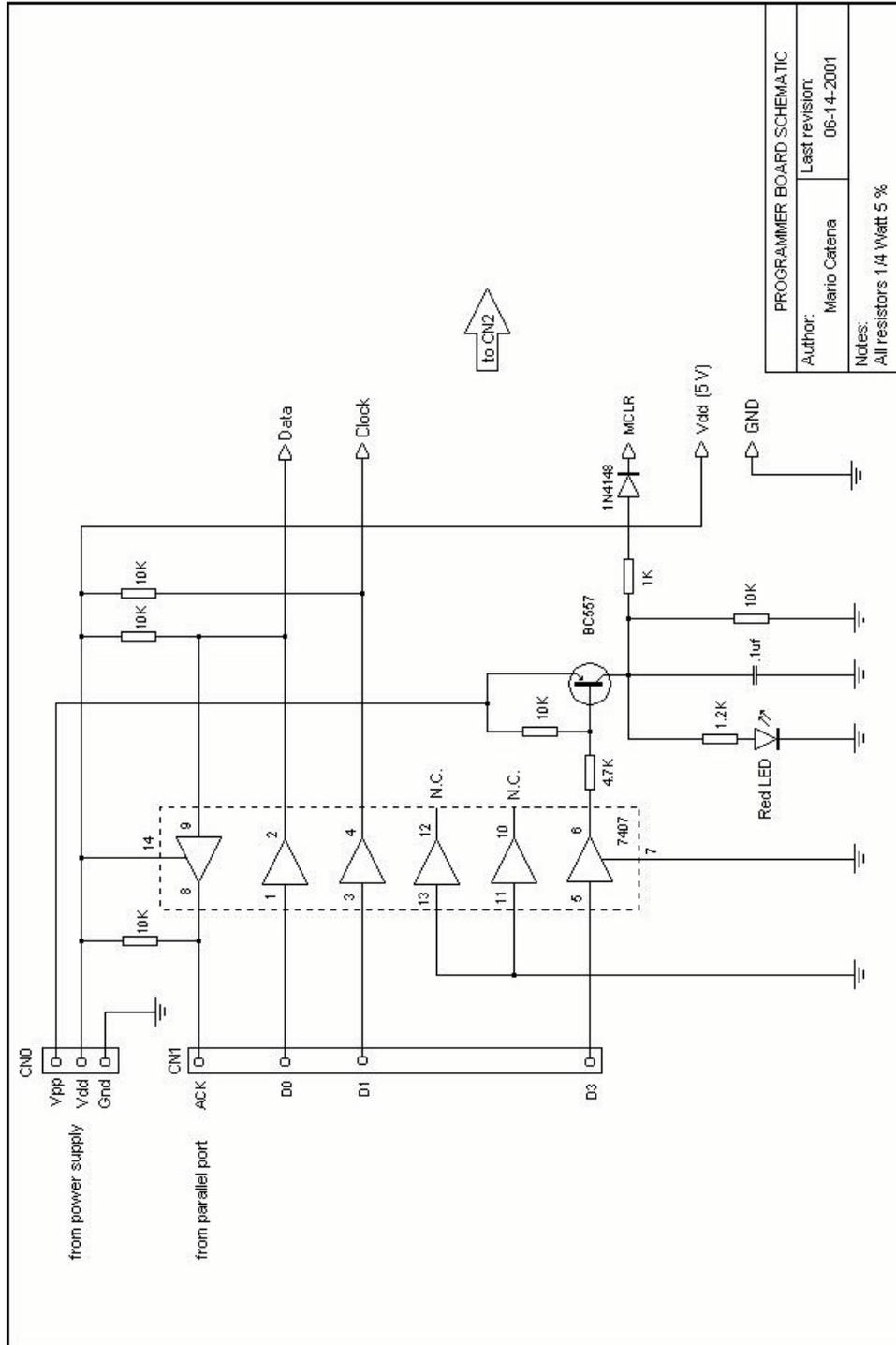


Fig. IV.4 – Progranador (www.ic-prog.com)

Apêndice V

Circuitos Envolvidos na Interface Modular

Para facilitar a compreensão, os módulos desenvolvidos são apresentados neste apêndice sem a caixa externa, demonstrando sucintamente a disposição dos seus componentes



Fig. V.1 – Amplificador Piezoelétrico



Fig. V.2 – Ajuste de Níveis de Tensão para o driver (MAANT)



Inibidor



Sonar

Fig. V.3 – Inibidor para adicionar módulo de Sonar

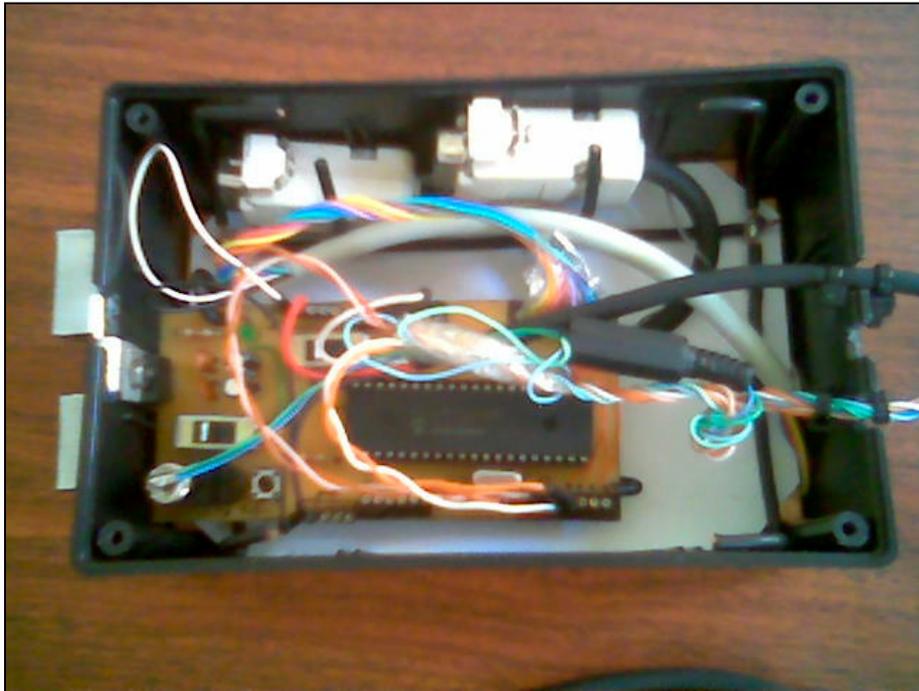


Fig. V.4 – Aplicação do microcontrolador 16F877A na Interface HaWCoS

Apêndice VI

Algoritmo do Firmware Utilizado na Interface HaWCoS

Para facilitar a compreensão, descreve-se sucintamente o algoritmo utilizado

(*): Subrotinas que ocorrem em paralelo às rotinas seguintes, por meio de interrupções de Timer 0

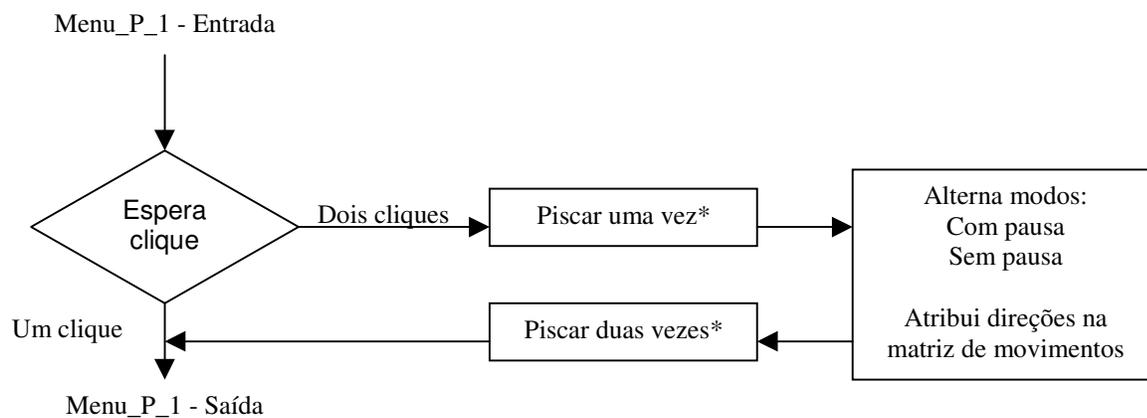


Fig. VI.1 – Subrotinas do menu 1

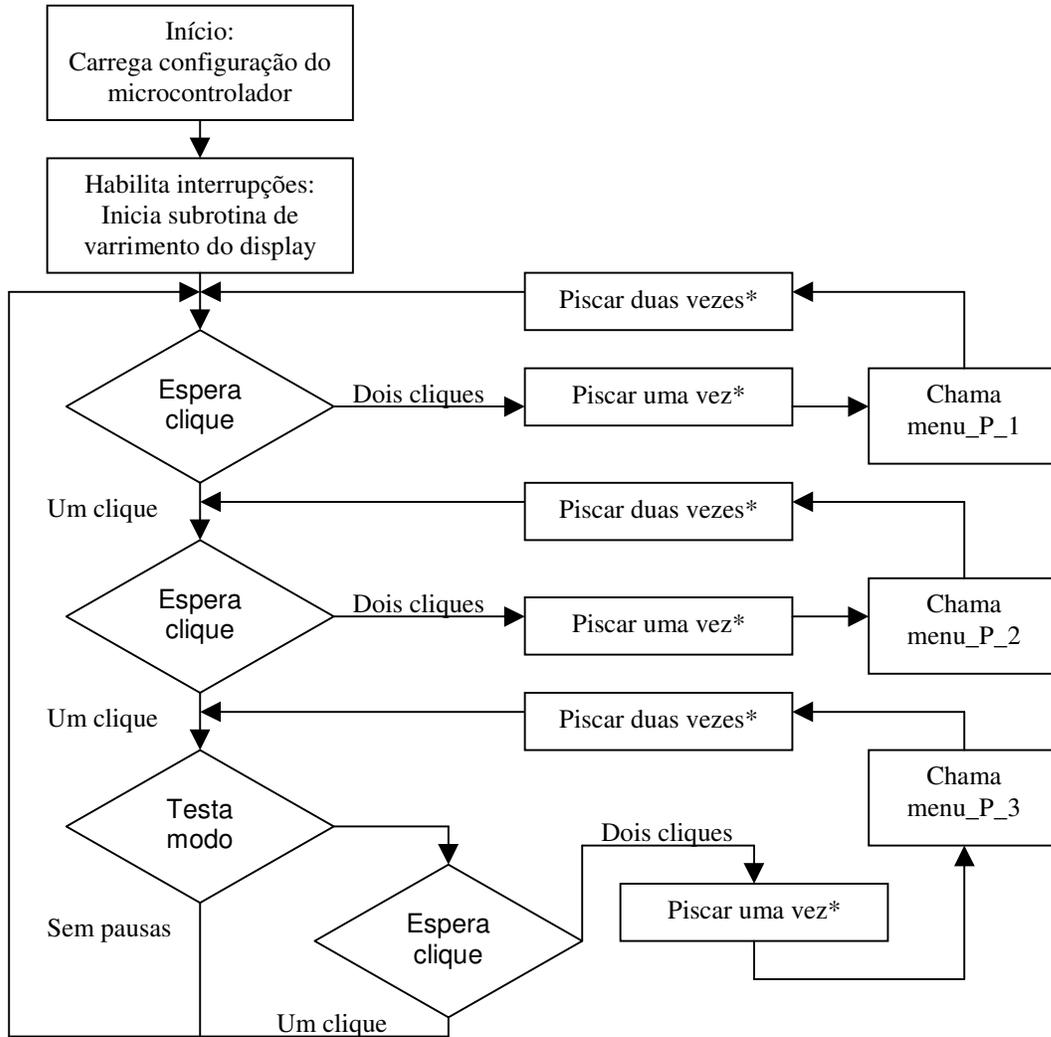


Fig. VI.2 – Rorina principal, responsável por organizar menus

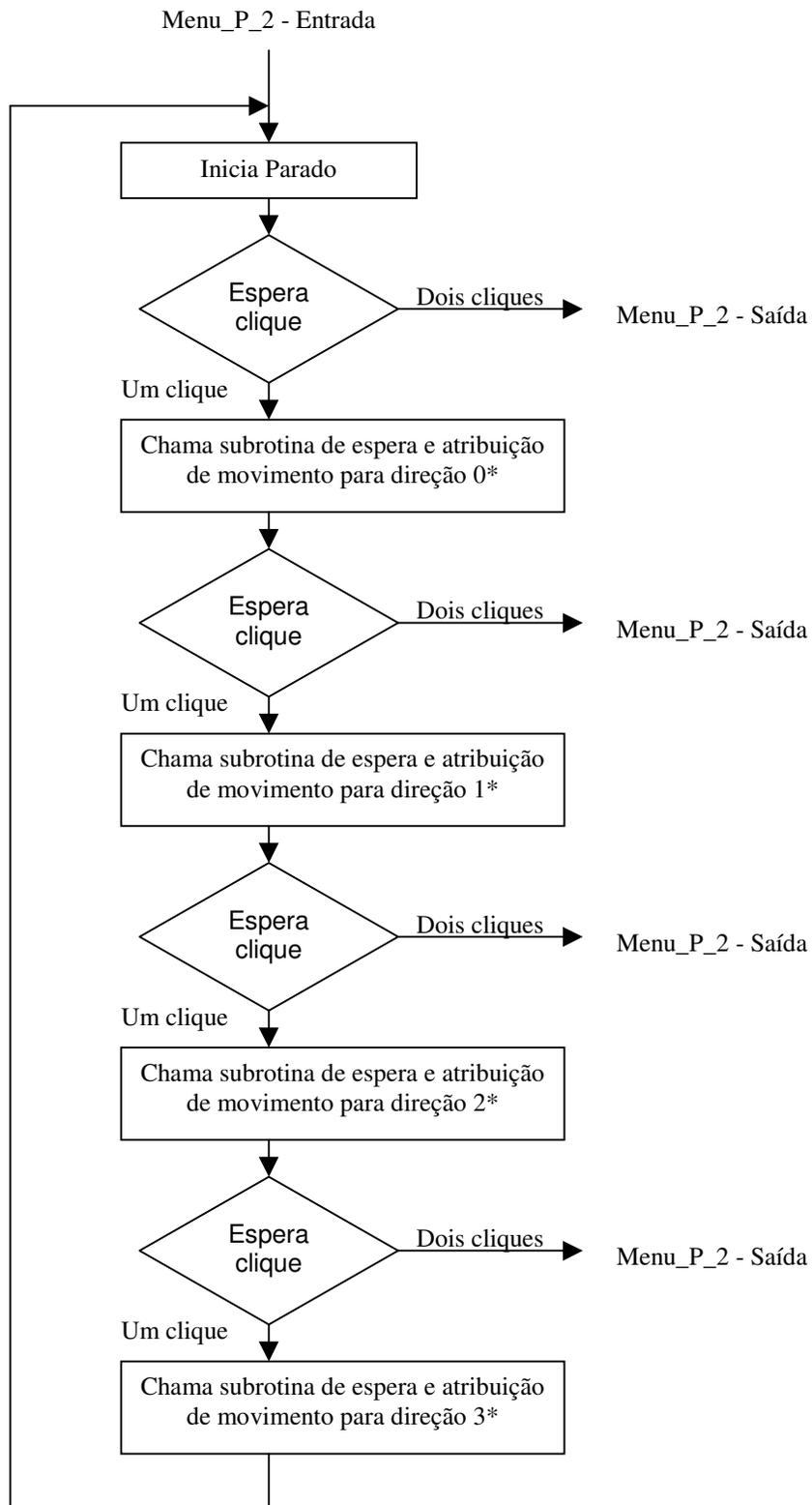


Fig. VI.3 – Subrotinas do menu 2

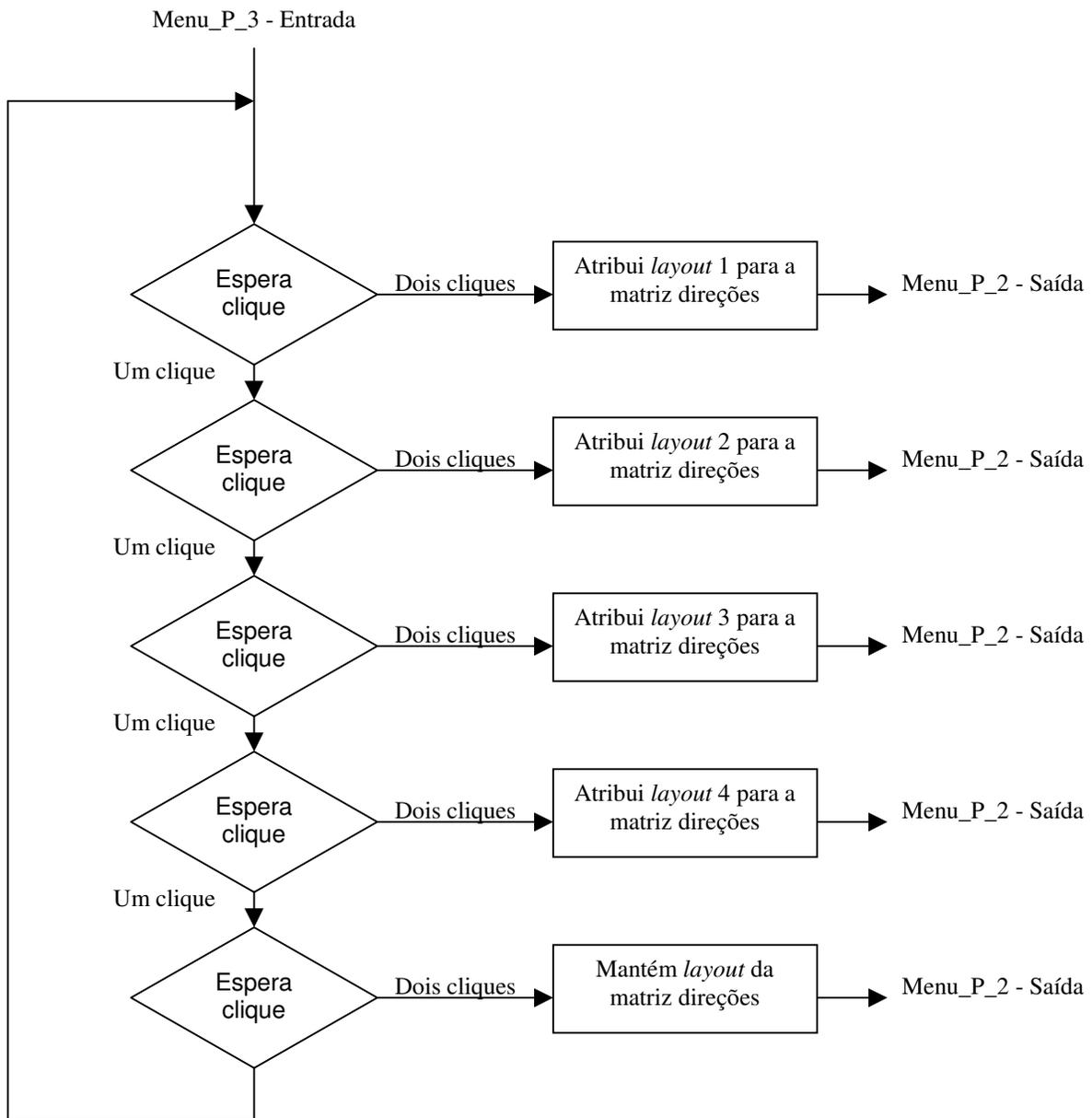


Fig. VI.4 – Subrotinas do menu 3

Ocorrência de borda de subida* - Entrada

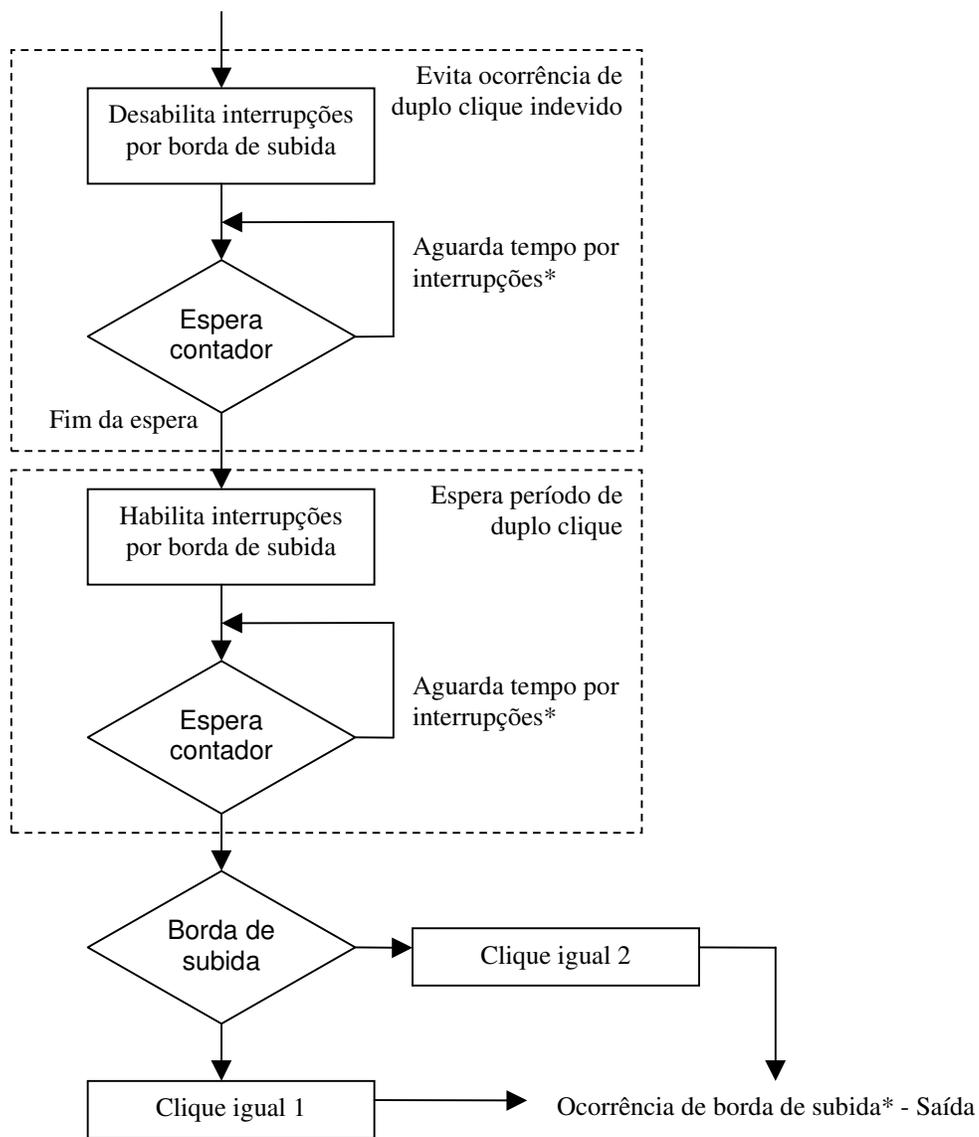


Fig. VI.5 – Subrotinas para obter interações do usuário por interrupção

Subrotina de atribuição de movimentos - Entrada

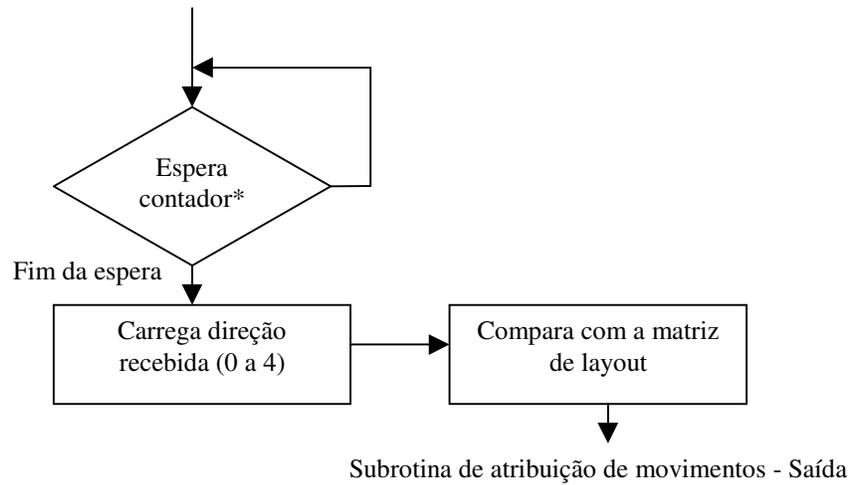


Fig. VI.6 – Atribuição de movimentos