



**UNIVERSIDADE CATÓLICA DE PELOTAS
ESCOLA DE ENGENHARIA E ARQUITETURA
CURSO DE ENGENHARIA ELÉTRICA/ELETRÔNICA**



**IDENTIFICAÇÃO DA RESPOSTA HUMANA AO
RASTREAMENTO UTILIZANDO TÉCNICAS ADAPTATIVAS**

RELATÓRIO FINAL DO BOLSISTA

PERÍODO: 2003

Aluno: Márcio da Silva Goulart

Orientadores: Márcio Holsbach Costa

Antônio Carlos da Rocha Costa

SUMÁRIO

1. Dados de Identificação	1
1.1 Nome da Universidade.....	1
1.2 Escola / Curso	1
1.3 Título.....	1
1.4 Período das Atividades	1
1.5 Nome do Bolsista.....	1
1.6 Nome dos Orientadores	1
2. Avaliação	2
3. Introdução	3
3.1 Objetivo	3
3.2 Justificativa	3
4. Fundamentos Teóricos	4
4.1 Identificação de Sistemas: Resposta Humana ao Rastreamento	4
4.2 Algoritmo Adaptativo <i>Least Mean Square</i> (LMS).....	5
4.3 Algoritmo NLMS	6
4.4 Desempenho de Filtros Adaptativos	7
5. Procedimentos Metodológicos	9
5.1 Diagrama em Blocos do Projeto	9
5.2 Diagrama em Blocos do Sistema de Identificação	11
5.3 Estrutura do Programa Desenvolvido	12
6. Resultados	14
6.1 Indivíduo 1	14
6.2 Indivíduo 2.....	17
6.3 Indivíduo 3.....	18
7. Discussão e Conclusão	21
8. Bibliografia.....	22
9. Apreciação Crítica.....	23
10. Anexos.....	24

1 DADOS DE IDENTIFICAÇÃO

1.1 Nome da Universidade: Universidade Católica de Pelotas

1.2 Escola/Curso: Curso de Engenharia Elétrica – Escola de Engenharia e Arquitetura

1.3 Título: Identificação da Resposta Humana ao Rastreamento Utilizando Técnicas Adaptativas

1.4 Período das Atividades: março à dezembro de 2003

1.5 Nome do Bolsista: Márcio da Silva Goulart

1.6 Nomes dos Orientadores: Márcio Holsbach Costa e Antônio Carlos da Rocha Costa

2 AVALIAÇÃO

Neste período pude avançar e finalizar meus estudos sobre o trabalho *Identificação da Resposta Humana ao Rastreamento*. Em decorrência disto, este período de pesquisa foi muito importante, pois foi onde conseguimos implementar o sistema de identificação e verificamos a sua eficiência.

Durante o trabalho de pesquisa pude aplicar e enriquecer os conhecimentos adquiridos durante o curso, bem como conhecer novos conceitos não vistos.

A conclusão a que chego ao fim deste período de pesquisa, é que estas atividades extracurriculares são de suma importância, à medida que propiciam ao aluno bolsista desenvolver trabalho de bancada, trabalhar em busca de soluções para problemas que ocorrem no decorrer do projeto, estudar e colocar em prática o que foi estudado, etc, ou seja, oportunizam ao aluno vivenciar situações que poderão ocorrer na trajetória de sua vida profissional.

Pelotas, 15 de dezembro de 2003.

Márcio da Silva Goulart
(bolsista)

Márcio Holsbach Costa
(orientador)

Antônio Carlos da Rocha Costa
(co-orientador)

3 INTRODUÇÃO

Este relatório apresenta os resultados do desenvolvimento de um sistema de identificação da resposta humana ao rastreamento. Suas principais aplicações são em testes psicotécnicos e na avaliação de desempenho motor.

3.1 Objetivos

O objetivo deste trabalho é estimar parâmetros representativos da habilidade/capacidade do usuário relacionados ao seu desempenho durante tarefas relacionadas ao processo de rastreamento.

3.2 Justificativa

Este trabalho justifica-se pelas suas aplicações na área da medicina, a qual está cada vez mais informatizada. Com isso, este projeto pode ser válido em vários tratamentos e exames médicos como em testes psicotécnicos, treinamento de pilotos de aeronaves, acompanhamento clínico de processos fisioterápicos, avaliação da capacidade motora em portadores de necessidades especiais, etc.

4 FUNDAMENTOS TEÓRICOS

Nesta seção serão introduzidos os conceitos teóricos necessários à compreensão do trabalho. Ao longo deste relatório a seguinte notação será utilizada: letras minúsculas em itálico representam variáveis escalares, letras minúsculas em negrito são vetores coluna e letras maiúsculas em negrito são matrizes quadradas. A variável n representa o tempo discreto.

4.1 Identificação de Sistemas: Resposta Humana ao Rastreamento

Diversos trabalhos na literatura mundial têm tentado determinar funções transferência para os Seres Humanos. A característica única de cada um deles é o método de identificação de sistemas utilizado [1].

Diversos experimentos têm demonstrado que o Ser Humano pode ser modelado por uma função transferência do tipo passa-baixas e que seu comportamento possui características que variam no tempo com determinado atraso.

O homem é um sistema de controle altamente adaptativo capaz de ajustar suas características dentro de uma ampla faixa de elementos dinâmicos e capaz de aprender e prever ações futuras de um sinal [1].

Funções transferência têm sido determinadas para quantificar a capacidade de rastreamento de Seres Humanos desde 1940. A maioria delas é similar à equação (1) determinada por McRuer e Krendel em 1957 [1]. Neste modelo D representa o atraso de transporte, T_a a constante de antecipação, T_l o erro de alisamento constante e T_n o atraso neuromuscular. O ganho K e as constantes de tempo variam de acordo com a tarefa de controle. D e T_n são assumidos fixos para cada operador, mas variações entre operadores são aceitas dentro de certos limites [1].

$$G(s) = \frac{K (1 + T_a s) e^{-Ds}}{(1 + T_l s)(1 + T_n s)} \quad (1)$$

onde s representa o domínio da transformada de Laplace.

Alguns valores representativos são os seguintes (em segundos):

$$\begin{cases} D = 0,2 \text{ s} \pm 20\% \\ T_a = 0 \text{ à } 2,5 \text{ s (variável)} \\ T_l = 0 \text{ à } 20 \text{ s (variável)} \end{cases}$$

4.2 Algoritmo Adaptativo *Least Mean Square* (LMS)

Há vários sinais de interesse na natureza. Alguns destes se caracterizam por uma certa regularidade em suas propriedades, o que lembra um sinal periódico. Entretanto, não podem ser definidos como tal porque existe uma pequena variação estatística a cada momento. Em geral, esses sinais são medidos conjuntamente com interferências indesejáveis, que devem ser eliminadas.

Usualmente, os filtros clássicos baseados em frequências de corte constantes não são eficientes para remoção dessas interferências, porque as características estatísticas tanto do sinal quanto da interferência variam no tempo. Portanto é necessário a utilização dos filtros adaptativos. De uma forma geral, os diversos algoritmos da família LMS (*Least Mean Square*), também chamados de algoritmos de gradiente estocástico, são determinados a partir da seguinte equação [2]

$$\mathbf{c}(n+1) = \mathbf{c}(n) + \mu \left[-\nabla E \{ e^2(n) \} \right] \quad (2)$$

Esta equação pode ser interpretada como sendo um procedimento recursivo, onde novas atualizações do vetor $\mathbf{c}(n) = [c_0(n) \ c_1(n) \ \dots \ c_{N-1}(n)]^T$ são determinadas a partir de seu valor no instante de tempo anterior mais uma parcela proporcional (onde o fator de proporcionalidade é dado pelo parâmetro μ) ao gradiente da superfície do sinal de erro médio quadrático.

O algoritmo LMS é importante pela sua simplicidade e facilidade de computação. Se a estrutura do sistema adaptativo é composta por um combinador linear e uma linha de atraso, como mostra a figura 1, e o sinal de entrada $x(n)$ e a resposta desejada $d(n)$ estão disponíveis a cada iteração, o algoritmo LMS é geralmente a melhor escolha para muitas aplicações de processamento adaptativo de sinais [3].

Na figura 1, o sinal de erro é dado por:

$$e(n) = d(n) - y(n) \quad (3)$$

onde $y(n) = \mathbf{c}^T(n) \mathbf{x}(n)$ é uma estimativa do sinal desejado e o vetor de amostras do sinal de excitação $\mathbf{x}(n)$ é definido como $\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]^T$.

Muitos algoritmos adaptativos estimam o gradiente da superfície do erro médio quadrático $\xi = E \{ e^2(n) \}$ através de diferenças entre pequenos termos de $e^2(n)$. Já o algoritmo LMS utiliza o próprio $e^2(n)$, como uma estimativa do MSE [3].

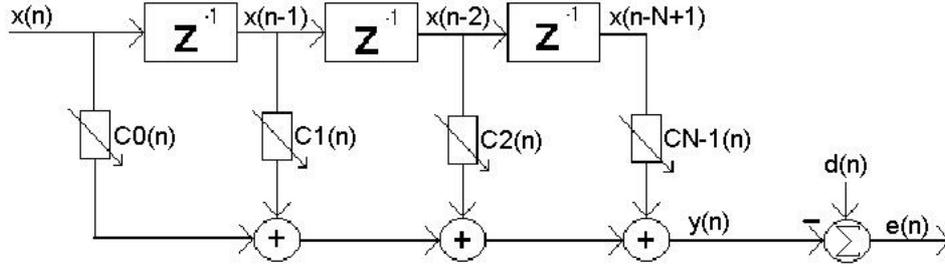


Figura 1. Combinador linear adaptativo e linha de atraso.

Então a cada iteração do processo adaptativo, temos uma estimativa do gradiente. Através da equação (3) tem-se que:

$$\nabla [E\{e^2(n)\}] \cong \nabla [e^2(n)] = \begin{bmatrix} \frac{\partial e^2(n)}{\partial c_0} \\ \dots \\ \frac{\partial e^2(n)}{\partial c_{N-1}} \end{bmatrix} = 2e(n) \begin{bmatrix} \frac{\partial e(n)}{\partial c_0} \\ \dots \\ \frac{\partial e(n)}{\partial c_{N-1}} \end{bmatrix} = -2e(n)\mathbf{x}(n) \quad (4)$$

Substituindo (4) em (2) pode-se obter a equação de atualização do algoritmo LMS:

$$\mathbf{c}(n+1) = \mathbf{c}(n) + \mu e(n)\mathbf{x}(n) \quad (5)$$

onde μ é a constante de ganho que regula a velocidade e a estabilidade da adaptação.

O algoritmo LMS pode ser implementado sem operações de média, divisão, raiz, ou diferenciação e é elegante em sua simplicidade e eficiência [3]. Entre os diversos membros da família LMS encontra-se o algoritmo NLMS, o qual é um dos mais utilizados na prática.

4.3 Algoritmo NLMS

O algoritmo NLMS ou LMS normalizado é uma variação do LMS. Ele aplica em sua fórmula uma normalização do sinal de excitação. Essa normalização provoca a utilização de passos de convergência variáveis, de acordo com a potência do sinal de entrada $x(n)$. O passo de convergência $\mu = \beta / [\mathbf{x}^T(n)\mathbf{x}(n) + \delta]$ será inversamente proporcional à potência do sinal de entrada $x(n)$. A equação de atualização do NLMS é dada por [3].

$$\mathbf{c}(n+1) = \mathbf{c}(n) + \frac{\beta}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta} e(n)\mathbf{x}(n) \quad (6)$$

Onde δ é uma constante pequena e positiva que evita divisões por zero quando o produto $\mathbf{x}^T(n)\mathbf{x}(n)$ for nulo.

4.4 Desempenho de Filtros Adaptativos

Algumas características do processo de adaptação são de grande utilidade para a análise e o entendimento do comportamento dos filtros adaptativos. A mais importante delas é a curva de aprendizado que caracteriza o desempenho de um filtro adaptativo e é amplamente utilizada em estudos teóricos e experimentais [3].

A curva de aprendizado relaciona o erro médio quadrático (EMQ) em função do tempo. Quando o filtro adaptativo inicia sua operação, os seus coeficientes podem possuir valores bem diferentes do vetor de coeficientes ótimo (\mathbf{c}^o - conjunto de coeficientes que minimiza o erro médio quadrático, ou sob outro ponto de vista, que produz a melhor estimativa de $d(n)$ através de $y(n)$ em termos médios quadráticos), causando um EMQ grande. Durante a execução do algoritmo, os coeficientes vão se aproximando de \mathbf{c}^o , causando por consequência uma diminuição do EMQ, esta fase é chamada de adaptação. Após esta fase o filtro entra no modo de regime permanente.

As figuras 2 e 3 mostram exemplos da operação de um filtro adaptativo, baseado no algoritmo LMS.

Na figura 2 foi utilizado um sinal de excitação não-correlacionado com características gaussianas, média zero e variância unitária, passo de convergência de 0,01, 2000 iterações, 10 realizações (promediações) e número de coeficientes da planta igual a 10. Na figura 3 foi utilizado um passo de convergência de 0,006, 2000 iterações, 20 realizações (promediações) e número de coeficientes da planta igual a 10, sob as mesmas condições.

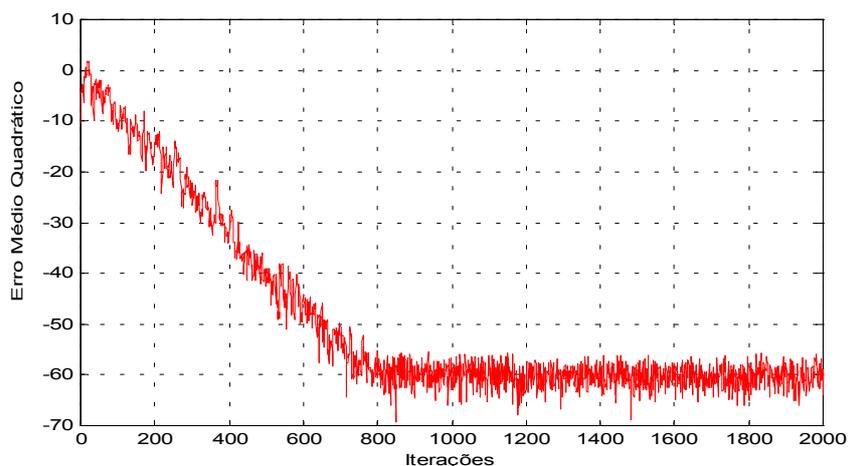


Figura 2. Desempenho do filtro adaptativo para um passo de convergência de 0,01, 2000 iterações, 10 realizações (promediações) e número de coeficientes da planta 10.

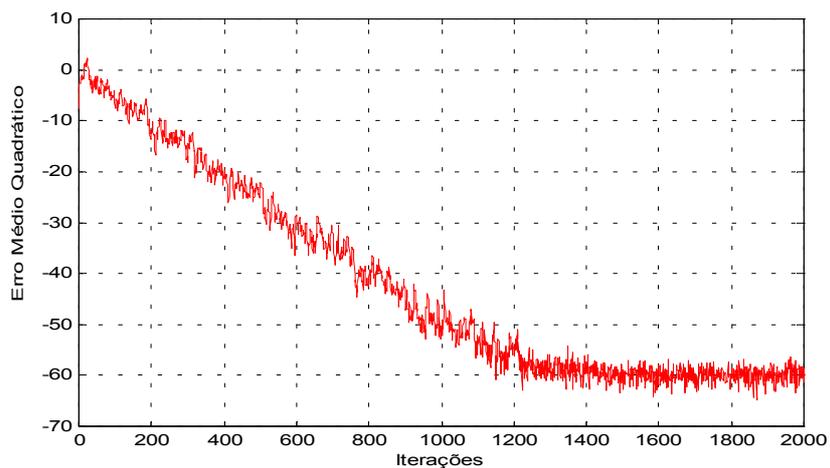


Figura 3. Desempenho do filtro adaptativo para um passo de convergência de 0,006, 2000 iterações, 20 realizações (promediações) e número de coeficientes da planta 10.

A escolha do passo convergência determina a duração do tempo de transição e o nível de cancelamento em regime permanente. Quanto menor for o erro em regime permanente melhor terá sido a estimação de $d(n)$ por $y(n)$.

5 PROCEDIMENTOS METODOLÓGICOS

Para a implementação do sistema proposto foram estudadas e desenvolvidas diversas partes do *software* necessário para o seu funcionamento.

5.1 Diagrama em Blocos do Projeto

Na figura 4 é apresentado o diagrama em blocos do projeto.

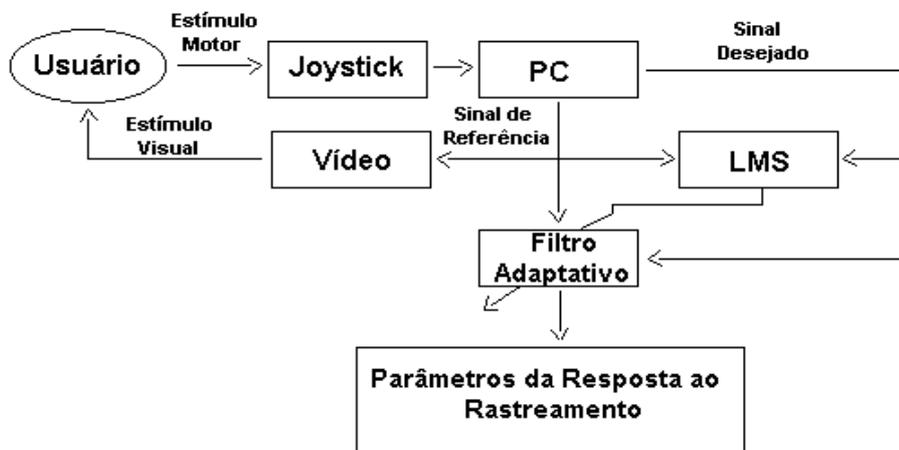


Figura 4. Diagrama em Blocos

O sistema a ser desenvolvido consiste em um estímulo visual produzido na tela de um computador pessoal, sob a forma de um caminho a ser percorrido por um cursor. Este cursor é controlado por um comando manual (*joystick*). A figura 5 apresenta uma visão da interface gráfica desenvolvida.

Na figura 5 pode-se identificar uma área gráfica que definirá a região de estímulo ao usuário (retângulo). Dentro desta área uma linha, definida pela palavra “trilha” determina o caminho pelo qual o usuário deverá guiar o cursor através do comando manual. A aparência final do sistema será muito semelhante a um jogo de computador, possibilitando que o mesmo seja aplicado em indivíduos de pouca idade sob a forma de atividade lúdica. Efeitos visuais e sonoros podem ser incorporados posteriormente.

O funcionamento do sistema apresentado pode ser descrito da seguinte maneira: o estímulo visual da trilha, produzido na tela do computador, é processado no cérebro do usuário e um estímulo motor é criado com a intenção de acompanhar as variações deste caminho (trilha). O sinal visual é portanto tratado como um sinal de referência (excitação) enviado a um sistema adaptativo. O sinal desejado $d(n)$ nada mais é que o estímulo visual processado pelo conjunto de sistemas visual-cognitivo-motor do

usuário, tendo como resposta um estímulo motor. Supondo-se que este sistema possa ser considerado linear dentro de certos limites que dependerão das frequências de excitação utilizadas e das condições ambientais, pode-se supor que este sinal seja aproximado pela seguinte equação:

$$d(n) = \mathbf{c}^{oT} \mathbf{x}(n) \quad (7)$$

onde $\mathbf{c}^{oT} = [c_0^o \ c_1^o \ \dots \ c_{N-1}^o]$ são os coeficientes que representam o sistema visual-cognitivo-motor.

A cada instante de tempo, cuja duração será definida ao longo do trabalho, a região de estímulo (parte interna do retângulo) será deslocada para a parte inferior da tela e uma nova atualização será realizada na parte superior dando a impressão de um comportamento dinâmico e variação contínua do estímulo. A sensação visual equivale ao de uma janela de visualização fixa sobre um caminho dinâmico que se move da parte superior para a inferior da tela.

De forma a possibilitar a identificação dos parâmetros de interesse (c^o), o sinal de estímulo será considerado como sinal de excitação ($x(n)$) para um sistema adaptativo. O sinal de erro é calculado da mesma forma que na equação (3) e novas atualizações dos coeficientes adaptativos serão obtidas através da equação (6).

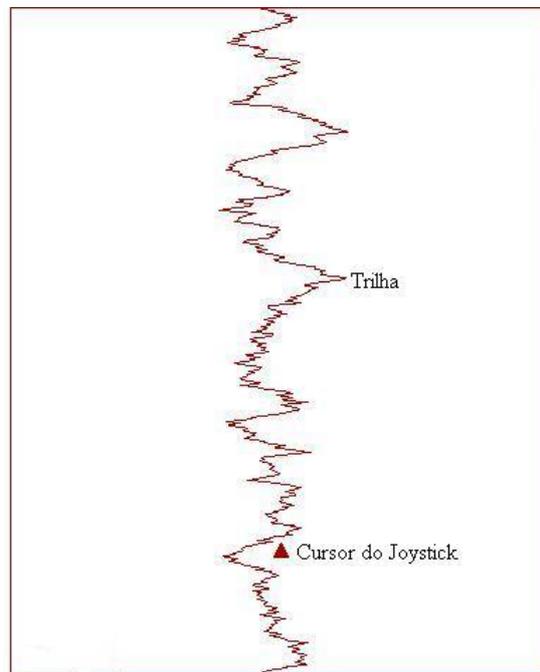


Figura 5. Interface gráfica

5.2 Diagrama em Blocos do Sistema de Identificação

Na figura 6 é apresentado o diagrama em blocos do sistema de identificação baseado no algoritmo LMS.

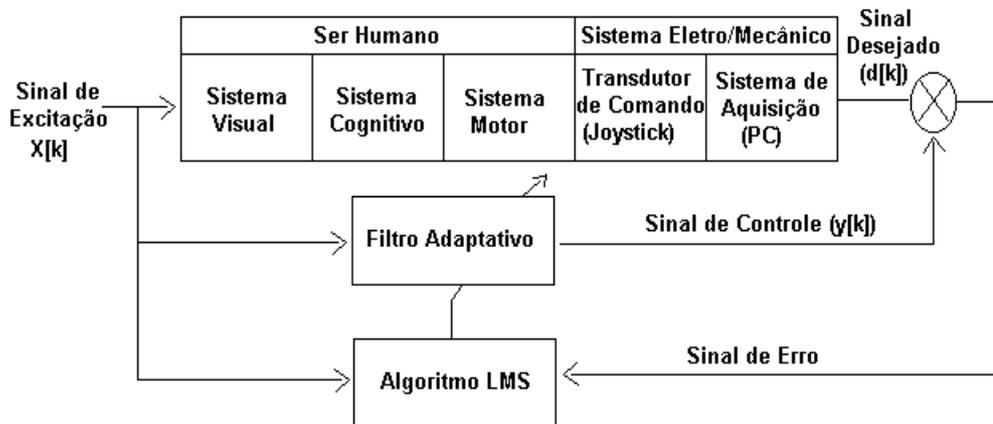


Figura 6: Diagrama em blocos do sistema de identificação baseado no algoritmo LMS.

O diagrama em blocos do sistema de identificação baseado no algoritmo LMS possui quatro partes principais, as quais são: Ser Humano, Sistema Eletro/Mecânico, Algoritmo NLMS e Filtro Adaptativo.

- **Ser Humano**

O sinal de excitação $x(n)$, localizado na parte superior esquerda da figura 6, ativa o sistema visual do ser humano, o qual ativa o sistema cognitivo que, por sua vez, ativa o sistema motor com o intuito de corrigir a porção do cursor.

- **Sistema Eletro/Mecânico**

O Sistema Eletro/Mecânico é constituído pelo comando manual e sistema de aquisição deste sinal. Embora sua influência seja identificada conjuntamente com os parâmetros humanos, suas características são invariáveis e portanto espera-se que sejam separáveis das informações referentes a cada indivíduo.

- **Algoritmo NLMS**

O algoritmo LMS tem a função de adaptar os coeficientes do filtro adaptativo de forma a produzir uma estimativa dos parâmetros da resposta motora humana.

- **Filtro Adaptativo**

Tem como objetivo produzir o sinal de controle $y(n)$ para a minimização do erro médio quadrático. Sua estrutura é a apresentada na figura 1 (combinador linear adaptativo) e seus parâmetros modificados a cada instante pelo algoritmo NLMS.

5.3 Estrutura do Programa Desenvolvido

O programa de gerenciamento do sistema foi implementado em linguagem C [4] e possui sete partes principais: interface gráfica, leitura do comando manual, algoritmo NLMS, determinação dos parâmetros de rastreamento, rotina de interrupção, rotina de gravação de dados e a integração do sistema.

- **Interface Gráfica**

Na interface gráfica do projeto foi implementada a rotina de desenho para delimitar a área onde atua o *joystick*, a rotina de desenho da trilha e a rotina de desenho do cursor do *joystick*.

O desenho da trilha foi implementado através de uma rotina de números aleatórios gaussianos correlacionados. A cada instante de tempo (determinado pela rotina de temporização), uma nova amostra aleatória é gerada, com média igual à metade da largura da tela (em pixels) e uma correlação determinada de forma que o sinal não varie muito abruptamente. Esta amostra é armazenada em um *buffer* de tamanho igual à altura da tela (em pixels) que funciona como uma linha de atraso. A cada amostra gerada, a mais antiga é descartada, as amostras antigas são atrasadas por uma unidade de tempo (índice = índice +1) e a nova amostra ocupa a primeira posição. Este *buffer* é utilizado para traçar a trilha (sinal de excitação para o usuário) (anexo 1).

Para que o usuário consiga percorrer a trilha, é preciso que o *joystick* seja mostrado na forma de um cursor. Para isso foi implementada uma rotina que salva a imagem na posição em que o cursor está e redesenha a mesma imagem em outra posição da tela, ou seja, na próxima posição em que o usuário vai colocá-lo [5]. A posição vertical é fixa e a horizontal determinada pelo *joystick*. O cursor foi desenhado na forma de um triângulo (anexo 1).

- **Leitura do Comando Manual**

Como o programa foi implementado no sistema operacional DOS, foi necessário uma rotina para fazer a leitura do *joystick*. Esta lê as informações existentes na porta serial do computador, a qual o *joystick* está conectado [5] (anexo1).

- **Algoritmo NLMS**

O algoritmo NLMS foi utilizado para a adaptação da estrutura FIR (*finite impulse response*) utilizada em conjunto com a linha de atraso. Ao final do processo de adaptação, quando da convergência do filtro adaptativo, os coeficientes representarão a

resposta ao impulso do sistema visual-cognitivo-motor do usuário. A partir deste resultado, diversas informações podem ser extraídas.

O algoritmo NLMS foi implementado no programa MatLab [6] que lê o arquivo de saída produzido pelo programa em C. O sinal de excitação é a coordenada x (variações horizontais) da trilha e o sinal desejado $d(n)$ é a coordenada x do *joystick* (anexo 2).

- **Determinação dos Parâmetros de Rastreamento**

Neste trabalho o objetivo é obter-se a resposta ao impulso do sistema visual-cognitivo-motor do usuário. Outras possibilidades podem ser exploradas futuramente.

- **Rotina de Interrupção**

Como o algoritmo adaptativo necessita de uma base de tempo constante foi preciso implementar uma rotina de interrupção relacionada ao *timer* do computador. Esta forma de temporização possui como grande fator de limitação a característica de que a frequência de amostragem máxima é de 18,2 Hz. Entretanto, como grande vantagem, não é necessária a utilização de *hardware* externo.

Em [1] foi desenvolvido um sistema de avaliação baseado em métodos paramétricos que utilizou uma frequência de amostragem de 21 Hz. Desta forma, acreditamos que a frequência de amostragem utilizada é suficiente para uma primeira abordagem do problema. Frequências maiores podem ser obtidas através da construção de um pequeno *hardware* externo para a temporização.

- **Rotina de Gravação de Dados**

Como o algoritmo NLMS foi implementado em Matlab (por questões de simplicidade), foi necessária uma rotina que fizesse a gravação dos dados do sistema. Neste arquivo é gravado o sinal de excitação (coordenada x da trilha) e do sinal desejado (coordenada x do *joystick*). Estes sinais são armazenados em *buffer* para depois, através de uma função que permite fazer escrita de blocos de informações, serem escritos no arquivo em que vão ser armazenados (anexo 1).

O método explicado acima realiza a gravação dos dados em formato binário. Em decorrência disto, foi também preciso implementar uma rotina em Matlab para fazer a leitura deste arquivo (anexo 2).

- **Integração do Sistema**

Após a conclusão das rotinas explicadas acima, partimos para a integração do sistema.

6 RESULTADOS

A partir do sistema descrito na seção 5.3 foram realizados cinco procedimentos de identificação para cada uma de três pessoas consideradas normais, de forma a averiguar a capacidade e velocidade de resposta ao processo de rastreamento de cada uma. O algoritmo NLMS foi implementado para um tamanho de filtro ($N = 500$ coeficientes) e um passo de convergência ($\mu = 0,1$), da mesma forma para as três pessoas (anexo 2).

6.1 Indivíduo 1

O indivíduo 1 é do sexo masculino, idade 23 anos, aparentemente normal em termos de capacidade física e mental e consentiu em participar do experimento como voluntário. A figura 7 apresenta um recorte do processo de rastreamento, onde se pode verificar que dentro de um certo grau de precisão o usuário está conseguindo seguir o sinal de excitação com um certo atraso.

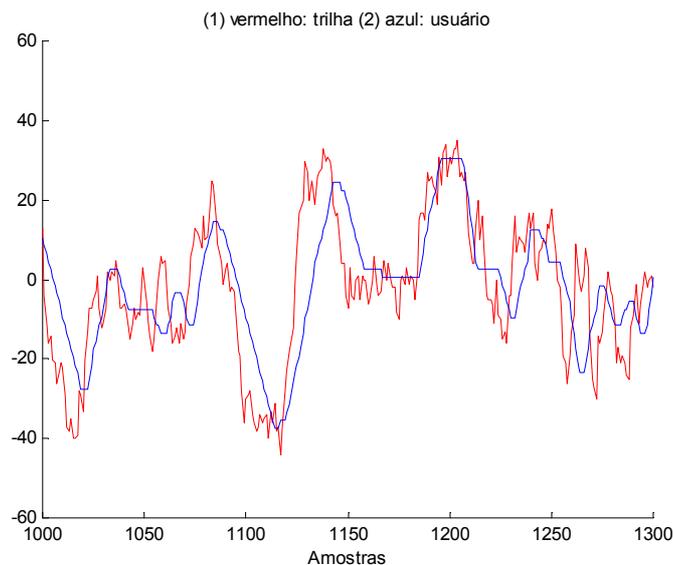


Figura 7. Processo de rastreamento. Em vermelho a trilha e em azul a resposta do usuário. No eixo horizontal encontram-se as amostras de cada processo e no vertical a sua amplitude.

A figura 8 apresenta o gráfico do erro quadrático (eixo vertical) pelo número de iterações (eixo horizontal) para apenas uma realização, representativa das demais, do processo de identificação. A partir desta figura, podemos notar que o erro quadrático diminui ao longo do número de iterações. Este comportamento indica a convergência do processo de adaptação.

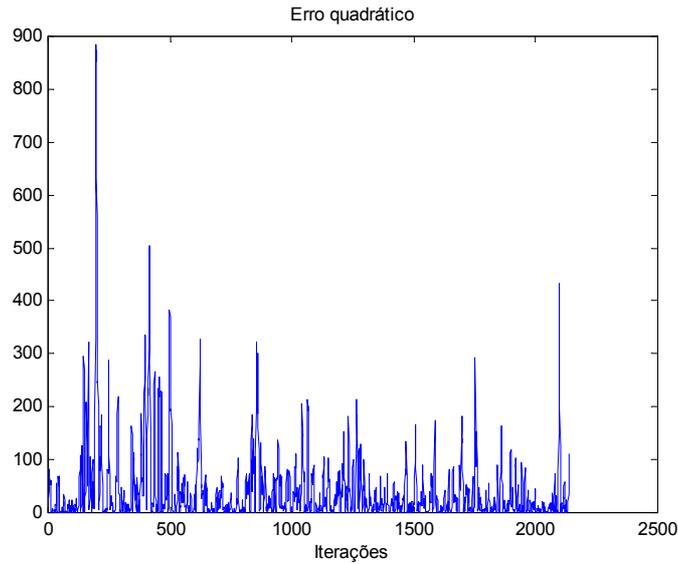


Figura 8. Erro quadrático de uma realização do processo de identificação para o indivíduo 1.

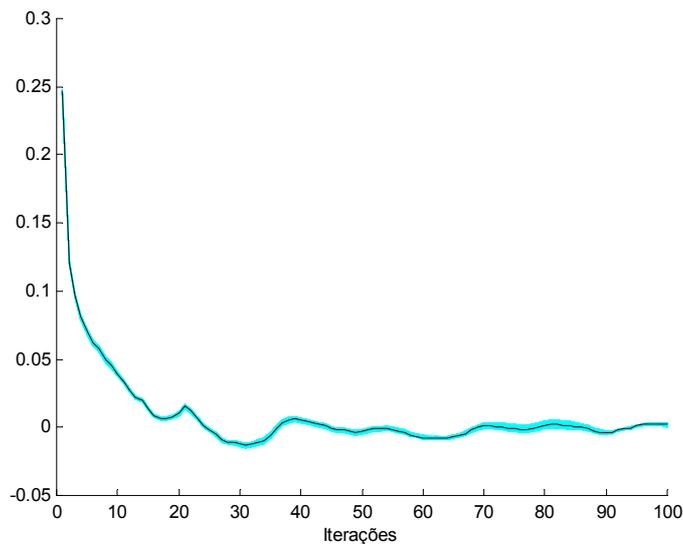


Figura 9. Coeficientes em regime permanente para indivíduo 1. Uma única realização. Em preto a média de 400 instantes de tempo e em ciano 400 instantes de tempo desenhados individualmente.

Para obter-se uma adequada estimativa dos coeficientes do filtro, foi realizada a média destes ao longo de 400 iterações. A figura 9 apresenta este resultado. Em preto visualiza-se a média das 400 amostras e em ciano as 400 individualmente, demonstrando a pequena variabilidade dos coeficientes neste trecho. O resultado apresentado representa a resposta ao impulso do sistema visual-cognitivo-motor humano para o indivíduo 1.

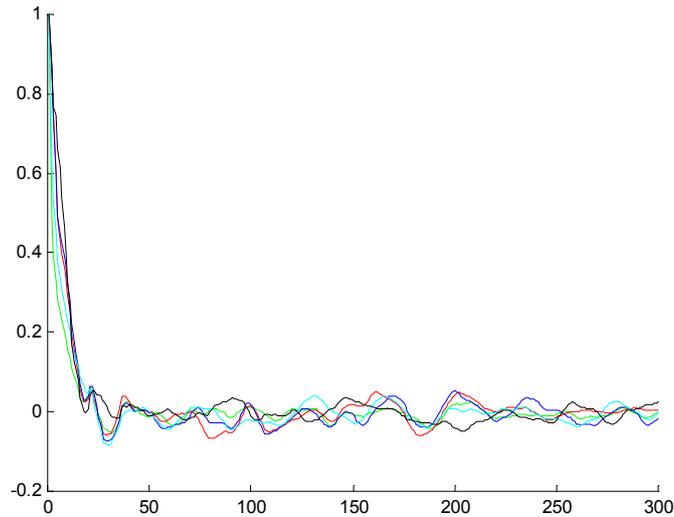


Figura 10. Coeficientes em regime permanente para o indivíduo 1. No eixo horizontal apresenta-se os coeficientes e no vertical seu valor normalizado. Cada cor representa a média de uma das realizações.

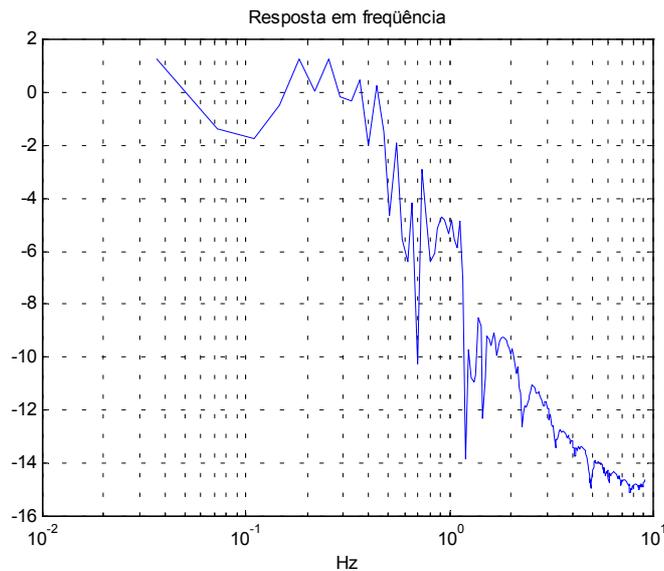


Figura 11. Resposta em frequência do sistema visual-cognitivo-motor do indivíduo 1.

A figura 10 apresenta uma comparação dos resultados normalizados de cada um dos cinco procedimentos de identificação para o mesmo indivíduo 1. Estes resultados demonstram a pequena variabilidade em procedimentos distintos.

A figura 11 apresenta a resposta em frequência do sistema visual-cognitivo-motor, obtida através da transformada de Fourier da resposta ao impulso obtida para uma realização. Analisando-a, verificamos que a mesma possui a característica passa-baixas esperada e, devido às suas características, parece improvável a existência de *aliasing* [7].

6.2 Indivíduo 2

O indivíduo 2 é do sexo masculino, idade 24 anos, aparentemente normal em termos de capacidade física e mental e consentiu em participar do experimento como voluntário.

A figura 12 apresenta o gráfico do erro quadrático (eixo vertical) pelo número de iterações (eixo horizontal) para apenas uma realização do processo de identificação. Nesta, podemos notar que o erro quadrático diminui ao longo do número de iterações. Este comportamento indica a convergência do processo de adaptação.

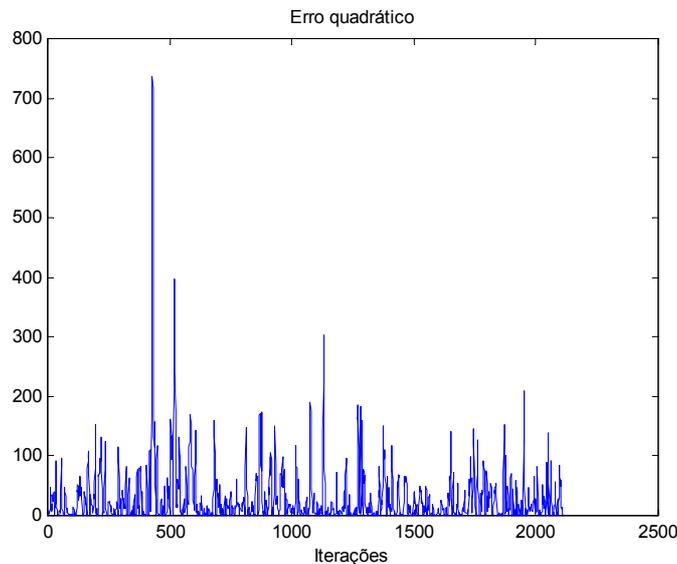


Figura 12. Erro quadrático de uma realização do processo de identificação para o indivíduo 2.

Para obter-se uma adequada estimativa dos coeficientes do filtro, foi realizada a média destes ao longo de 400 iterações. A figura 13 apresenta este resultado. Em preto visualiza-se a média das 400 amostras e em ciano as 400 individualmente, demonstrando uma maior variabilidade dos coeficientes neste trecho em comparação ao indivíduo 1. O resultado apresentado representa a resposta ao impulso do sistema visual-cognitivo-motor humano para o indivíduo 2.

A figura 13 apresenta um atraso maior na resposta do indivíduo 2 em relação ao indivíduo 1. Nota-se claramente pela posição do pico da resposta ao impulso que existe um atraso médio de duas unidades de tempo, ou seja, 0,11 segundos.

A figura 14 apresenta a resposta em frequência do indivíduo 2 em termos de resposta ao rastreamento. Pode-se verificar que, novamente, a curva é do tipo passa baixas com uma frequência de corte na faixa de 0,5 Hz, aproximadamente a mesma do indivíduo 1. Entretanto, esta curva possui um decaimento mais abrupto, na faixa de -25

dB por década contra -10 dB por década do indivíduo 1. Podemos intuir que este indivíduo possui uma limitação maior na resposta à variações de maior frequência.

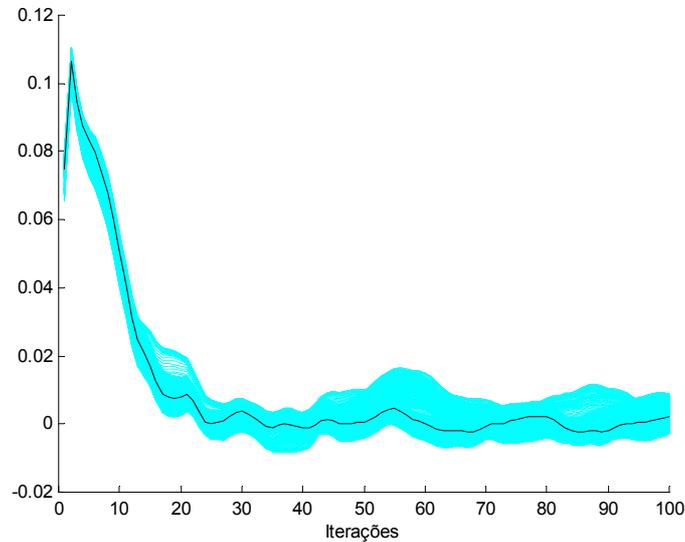


Figura 13. Coeficientes em regime permanente para indivíduo 2. Uma única realização. Em preto a média de 400 instantes de tempo e em ciano 400 instantes de tempo desenhados individualmente.

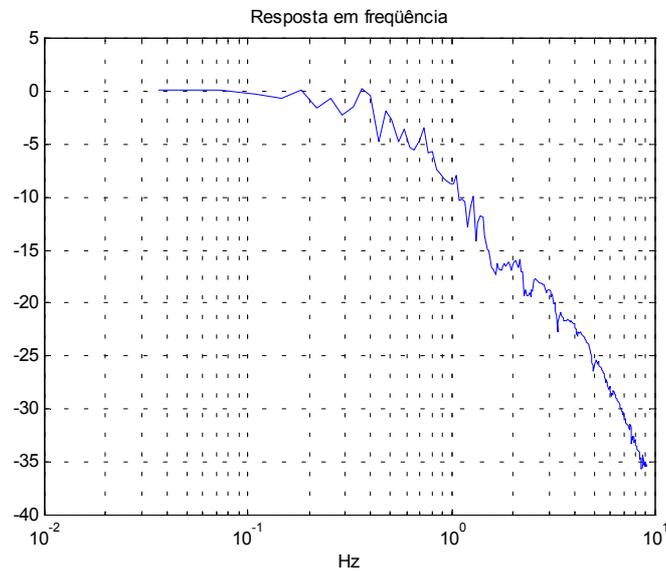


Figura 14. Resposta em frequência do sistema visual-cognitivo-motor do indivíduo 2.

6.3 Indivíduo 3

O indivíduo 3 é do sexo masculino, idade 22 anos, aparentemente normal em termos de capacidade física e mental e consentiu em participar do experimento como voluntário.

A figura 15 apresenta o gráfico do erro quadrático (eixo vertical) pelo número de iterações (eixo horizontal) para apenas uma realização do processo de identificação. Novamente, pode-se intuir a convergência do algoritmo adaptativo. A figura 16 apresenta os coeficientes em regime permanente no mesmo formato da figura 9. Os resultados são bastante parecidos com os do indivíduo 1 apresentando uma resposta quase imediata (em média).

Finalizando, a figura 17 apresenta uma estimativa da resposta em frequência para o indivíduo 3, cujas características são: frequência de corte de 0,5 Hz e queda de – 15 dB por década.

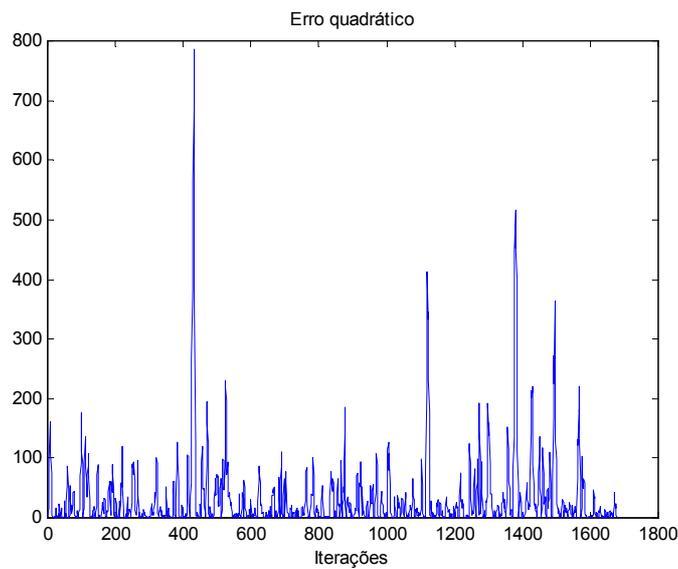


Figura 15. Erro quadrático de uma realização do processo de identificação para o indivíduo 3.

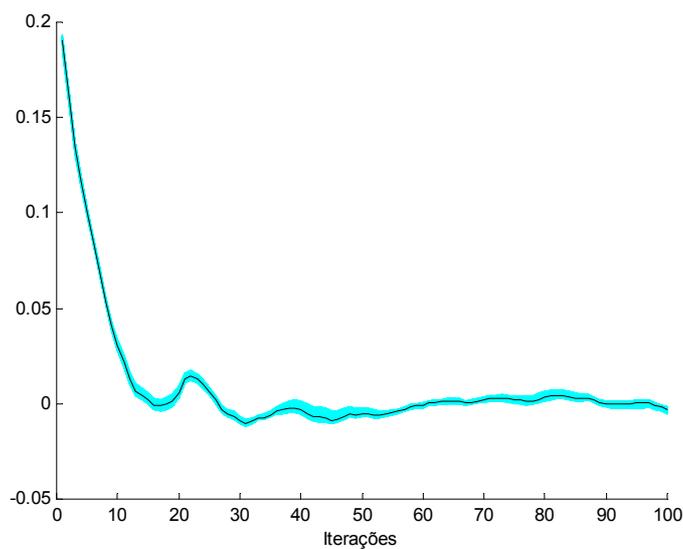


Figura 16. Coeficientes em regime permanente para o indivíduo 3: média de 400 instantes de tempo (preto) e 400 instantes de tempo desenhados individualmente (azul).

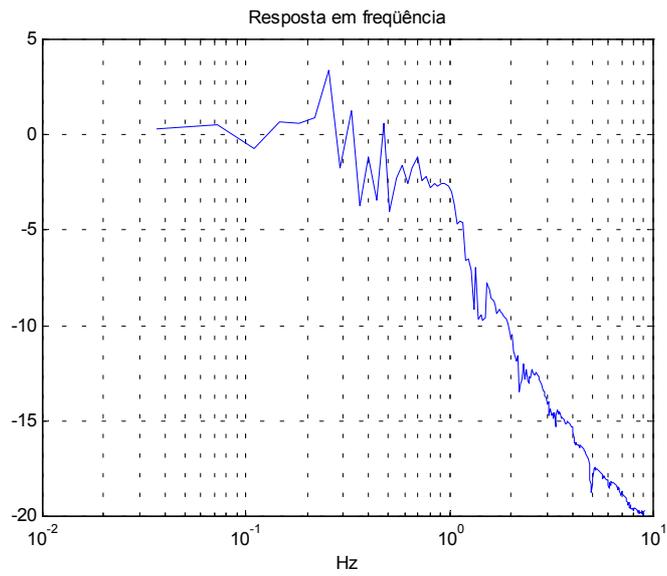


Figura 17. Resposta em frequência do sistema visual-cognitivo-motor do indivíduo 3.

7 DISCUSSÃO E CONCLUSÕES

Ao longo deste trabalho foram discutidas as diversas características e formas de funcionamento de um sistema de identificação da resposta humana ao rastreamento. As bases físicas e matemáticas foram analisadas, resultando na proposta de um protótipo para a implementação prática do sistema.

O sistema construído faz uso de técnicas de filtragem adaptativa como algoritmo de identificação. Esta estratégia foi implementada em linguagem C contendo a rotina de interface gráfica, a de temporização e a de leitura do *joystick*. Os dados obtidos são armazenados em arquivo para posterior processamento. Foi também desenvolvida uma rotina de identificação de sistemas, baseada no algoritmo NLMS, no programa Matlab, que permite a obtenção da resposta impulso do sistema analisado.

Através dos resultados obtidos na seção 6 foi verificada a potencialidade do sistema de identificação desenvolvido neste trabalho.

Os testes realizados nos levam a crer que a capacidade de rastreamento não é determinada pela frequência de corte obtida a partir da resposta em frequência do indivíduo, mas sim pela ordem do sistema em si (decaimento em altas frequências).

Como conclusão, o sistema desenvolvido permite a verificação das propriedades e características da resposta humana ao rastreamento, possibilitando o estudo e avaliação dos diversos fatores determinantes de seu desempenho.

Em anexo a este relatório encontra-se o programa com as rotinas implementadas.

8 BIBLIOGRAFIA

- [1] Gittleman, B., Dwan, T.E. e Smiley, C.S. (1992), “System Identification: Human Tracking Response”, IEEE Transactions on Education, volume 35, número 1, fevereiro, páginas 31-37.
- [2] Brito, D.S. e Barros, A.K. (2002), “Influência de Ruídos de Frequências muito Baixas na Estimativa de Sinais Quasi-periódicos pelo Algoritmo LMS”, XIV - Congresso Brasileiro de Automática.
- [3] Born, R.S. (2000), Filtros Adaptativos Aplicados a Sinais Biomédicos, Monografia de Conclusão de Curso, Bacharelado em Ciência da Computação, Universidade Federal de Pelotas.
- [4] Schildt, H. (1997), C Completo e Total, Makron Books, terceira edição.
- [5] Morrison, J. (1995), “joy.c”, jack@robotics.jpl.nasa.gov.
- [6] Carvalho, D.B. (2000), Introdução ao MatLab, Apostila do Curso de Engenharia Elétrica e Eletrônica da Universidade Católica de Pelotas.
- [7] Lyons, R.G. (2001), Understanding Digital Signal Processing, Prentice-Hall.

9 APRECIACÃO CRÍTICA

Ao longo deste período o aluno demonstrou assiduidade, interesse e capacidade para desenvolver um projeto de pesquisa sob orientação. Demonstrou também capacidade de trabalho em grupo.

Pelotas, 15 de dezembro de 2003.

Márcio da Silva Goulart
(bolsista)

Márcio Holsbach Costa
(orientador)

Antônio Carlos da Rocha Costa
(co-orientador)

10 ANEXOS

- **Anexo 1:** Programa em C para a implementação do sistema de identificação

```
/******  
/* Programa de Identificacao da Resposta Humana ao Rastreamento */  
/* Programadores: Marcio Goulart e Marcio Costa */  
/* Plataforma: Turbo C++ 2.01 */  
/* Rastrea.c */  
/* versao: 15/12/2003 */  
/******  
  
#include <conio.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <graphics.h>  
#include <dos.h>  
#include <math.h>  
  
/******  
/* Definicoes do programa */  
/******  
  
#define NTAB 32 // definições da rotina de números  
#define IQ 127773 // aleatórios  
#define IA 16807  
#define IM 2147483647  
#define IR 2836  
#define NDIV (1+(IM-1)/NTAB)  
#define AM (1.0/IM)  
#define EPS 1.2E-7  
#define RNMX (1.0-EPS)  
  
#define JPORT 0x201 // define porta para o joystick  
#define CMAX 9999  
#define PASSO 2 // define passo maximo  
  
#define CORTRI 15 // cor da trilha  
#define CORBOR 14 // cor da borda  
#define CORCUR 4 // cor do cursor  
#define TAMCUR 5 // tamanho do cursor  
#define POSCUR 0.8 // posicao do cursor  
  
#define VARI 80 // variancia do ruido branco  
#define NUMCOEFS 20 // filtro passa-baixas para correlacionar o sinal de excitacao  
#define GNUMBUF 16 // tamanho do buffer de gravacao  
  
/******  
/* Variaveis globais */  
/******  
  
int xi, xf, yi, yf; // tamanho da tela grafica  
int x, y; // posicao do cursor  
int trilha[5000]; // vetor de pontos da trilha  
int numtrilha; // n.º de elementos da trilha  
  
int v[4]; // posicao do joystick  
char b[4]; // botoes do joystick  
void *Saucer; // cursor  
  
long SEMENTE = -2; // semente do gerador de numeros aleatorios (negativa)  
long *SEM = &SEMENTE;  
  
float coef[NUMCOEFS]; // coeficientes do filtro adaptativo  
float memo[NUMCOEFS]; // buffer do sinal de referencia  
int Xcentro; // posicao do centro da tela  
  
FILE *stream; // arquivo de saida
```

```
int buffer[GNUMBUF]; // buffer para gravacao
int contador; // contador do buffer de gravacao
```

```
/* **** */
/* Chamadas das subrotinas */
/* **** */
```

```
void joystick(int v[4], char b[4]);
void interrupt(*velha_int_08h)();
```

```
/* **** */
/* Gerador de numeros aleatorios: distribuicao uniforme (0,1) */
/* **** */
```

```
float ran1(long *idum)
{
    int j;
    long k;
    static long iy = 0;
    static long iv[NTAB];
    float temp;

    if (*idum <= 0 || !iy) {
        if (-(*idum) < 1) *idum = 1;
        else *idum = - (*idum);
        for (j=NTAB+7; j>=0; j--) {
            k=(*idum)/IQ;
            *idum = IA*( *idum-k*IQ)-IR*k;
            if (*idum < 0) *idum += IM;
            if (j < NTAB) iv[j] = *idum;
        }
        iy = iv[0];
    }
    k = (*idum)/IQ;
    *idum = IA*( *idum-k*IQ)-IR*k;
    if (*idum < 0) *idum += IM;
    j = iy/NDIV;
    iy = iv[j];
    iv[j] = *idum;
    if ((temp=AM*iy) > RNMX) return RNMX;
    else return temp;
}
```

```
/* **** */
/* Gerador de numeros aleatorios: distribuicao gaussiana (media zero e variancia unitaria) */
/* **** */
```

```
float gasdev(long *idum)
{
    float ran1(long *idum);
    static int iset = 0;
    static float gset;
    float fac, rsq, v1, v2;

    if (*idum < 0) iset = 0;
    if (iset == 0) {
        do {
            v1 = 2.0*ran1(idum) - 1.0;
            v2 = 2.0*ran1(idum) - 1.0;
            rsq = v1*v1 + v2*v2;
        } while (rsq >= 1.0 || rsq == 0.0);
        fac = sqrt(-2.0*log(rsq)/rsq);
        gset = v1*fac;
        iset = 1;
        return v2*fac;
    } else {
        iset = 0;
        return gset;
    }
}
```

```

    }
}

/*****
/*   Rotina do joystick   */
*****/

void joystick(int v[4], char b[4])
{
    int i, j;
    int mask = 15;

    disable();
    outp(JPORT, 0);
    for (i=1; mask && i<CMAX; i++) {
        j = inp(JPORT) ^ mask;
        if (j & 1) { v[0] = i; mask ^= 1; }
        if (j & 2) { v[1] = i; mask ^= 2; }
        if (j & 4) { v[2] = i; mask ^= 4; }
        if (j & 8) { v[3] = i; mask ^= 8; }
    }

    enable();
    j = inp(JPORT);
    b[0] = !(j & 0x10);
    b[1] = !(j & 0x20);
    b[2] = !(j & 0x40);
    b[3] = !(j & 0x80);
}

/*****
/*   Rotina de interrupcao   */
*****/

void interrupt int_08h()
{
    int cont;
    float filt;

    (*velha_int_08h());           // interrupção original
    joystick(v, b);              // leitura do joystick

    putimage(x-TAMCUR, y, Saucer, XOR_PUT); // apaga cursor
    x = x + PASSO * (5*(v[0]-512)/512); // nova posicao do cursor
    if (x < xi+1+TAMCUR) x = xi + 1 + TAMCUR; // limita posicao do cursor a esquerda
    if (x > xf-1-TAMCUR) x = xf - 1 - TAMCUR; // limita posicao do cursor a direita

    setcolor(0);                 // apaga trilha antiga
    drawpoly(numtrilha, trilha);

    for(cont=NUMCOEFS-1; cont>0; cont--) // buffer circular do sinal branco
        memo[cont] = memo[cont-1];
    memo[0] = (float)VARI * gasdev(SEM) + Xcentro; // armazena sinal branco no buffer de correlação

    filt = 0;                    // correlação do sinal branco
    for(cont=0; cont<NUMCOEFS; cont++)
        filt = filt + memo[cont]*coef[cont];

    for(cont=numtrilha-1; cont>=2; cont--) // linha de atraso do sinal de excitação correlacionado
        trilha[2*cont-2] = trilha[2*cont-4];
    trilha[0] = (int)filt;        // atualização da trilha

    setcolor(CORTRI);           // desenha nova trilha
    drawpoly(numtrilha, trilha);

    putimage(x-TAMCUR, y, Saucer, OR_PUT); // desenha cursor na nova posição
    putimage(x-TAMCUR, y, Saucer, XOR_PUT);
    putimage(x-TAMCUR, y, Saucer, XOR_PUT);

    /* Abre um arquivo texto para escrita */
    buffer[contador] = trilha[0];
    buffer[contador+1] = x + Xcentro;

    if (contador==GNUMBUF-2)

```

```

        {
            fwrite(buffer,sizeof(int),GNUMBUF,stream);
            fflush(stream);
            contador = -2;
        }
    contador = contador + 2;

}

/*****
/*      Rotina principal      */
*****/

main()
{

int placa, modo, coderro;
int tamx, tamy;
int ulx, uly, lrx, lry;
int tam, cont;

/* Inicializacao do modo grafico */
detectgraph(&placa,&modo);
initgraph(&placa,&modo,"c:\\tc\\bgi");

/* Resultado da inicializacao do modo grafico */
coderro = graphresult();
if (coderro != grOk)
    {
        printf("Erro de inicializacao do modo grafico: %s\n", grapherrormsg(coderro));
        printf("Pressione uma tecla para sair:");
        getch();
        exit(1);
    }

/* Descobre o tamanho da tela */
tamx = getmaxx();
tamy = getmaxy();

/* Determina o tamanho da janela ativa */
xi = 0.2*tamx;      // x inicial com 20% do tamanho da tela
xf = 0.8*tamx;      // x final com 80% do tamanho da tela
yi = 0;             // y inicial 0, pois queremos iniciar bem de cima
yf = tamy;          // y final tamy, pois queremos terminar no fim de baixo

/* Desenha o cursor do joystick */
ulx = xi;
uly = yi;
lrx = xi + 2*TAMCUR;
lry = yi + 2*TAMCUR;
setcolor(CORCUR);
line((ulx+lrx)/2,uly,lrx,lry);    // desenha cursor triangular
line(ulx,lry,lrx,lry);
line((ulx+lrx)/2,uly,ulx,lry);
setfillstyle(1,CORCUR);           // define cor do cursor
floodfill(xi+TAMCUR,yi+TAMCUR,CORCUR); // preenche o cursor
tam = imagesize(ulx,uly,lrx,lry); // determina tamanho do cursor
Saucer = malloc(tam);             // reserva memoria para o cursor
getimage(ulx, uly, lrx, lry, Saucer); // desenha o cursor
putimage(ulx, uly, Saucer, XOR_PUT); // apaga o cursor

/* Posicionamento inicial do cursor */
memset(v, 0, sizeof(v));          // inicializa com zeros
x = tamx / 2;                     // meio da horizontal
y = (int)(POSCUR * tamy);         // parte baixa da vertical
moveto(x,y);                      // move para as coordenadas x e y
putimage(x-TAMCUR, y, Saucer, XOR_PUT); // desenha o cursor
Xcentro = x;

/* Abre arquivo para gravacao */
stream = fopen("c:\\goulart\\Result.dat", "a+b");
fwrite(&y,sizeof(int),1,stream);

/* Inicializacao da trilha */

```

```

numtrilha = yf - yi - 2;
for(cont=0;cont<=(numtrilha-1);cont++)
    {
        trilha[2*cont] = Xcentro;
        trilha[2*cont+1] = yi + 1 + cont;
    }

setcolor(CORTRI);
drawpoly(numtrilha,trilha);

/* Filtro de correla#o do sinal de excita#o */
for (cont=0; cont<NUMCOEFS; cont++)
    coef[cont] = 1 / (float)NUMCOEFS;

for (cont=0; cont<NUMCOEFS; cont++)
    memo[cont] = Xcentro;

/* Desenha o retangulo */
setcolor(RED); // define cor da borda
rectangle(xi,yi,xf,yf); // desenha o retangulo

contador = 0;

/* Inicia as interrupcoes */
velha_int_08h = getvect (0x8);
setvect(0x8,int_08h);

while (!kbhit()) // espera uma tecla ser pressionada
    {
    }

/* Finaliza o Programa */
setvect(0x8,velha_int_08h); // encerra as interrup#oes
free( Saucer );
closegraph();

/* Esvazia o buffer */

if (contador != 0)
    {
    fwrite(buffer,sizeof(int),contador,stream);
    }
fclose(stream);
}

```

- **Anexo 2:** Programa em Matlab para a implementa#ao do algoritmo NLMS.

```

% -----
% Rastreamento.m
%
% Algoritmo adaptativo NLMS aplicado ao sistema de
% rastreamento.
%
% 15/12/2003
% -----

% par#metros

ARQ = 'c:\goulart\haical5.dat'; % nome do arquivo de entrada
N = 500; % n#mero de coeficientes
MU = 0.1; % passo de converg#ncia

% leitura dos sinais

fid = fopen(ARQ,'r'); % abre arquivo
atraso = fread(fid,1,'int16');
z = fread(fid,inf,'int16'); % l# amostras
fclose(fid);

% separa sinal de excita#ao e sinal desejado

```

```

TAM = length(z);
x1 = z(1:2:TAM);
d1 = z(2:2:TAM);

% descarta o atraso

a = (TAM/2); % tamanho do sinal
sinJ = d1((atraso+1):a);
b = length(sinJ);
sinT = x1(1:b);

% retira média dos sinais

medJ = mean(sinJ);
medT = mean(sinT);
sinJ = sinJ - medJ;
sinT = sinT - medT;

% algoritmo LMS

NUM = length(sinT); % número de iterações

mem = zeros(N,1); % linha de atraso
C = zeros(N,NUM+1); % coeficientes do filtro adaptativo
e = zeros(1,NUM); % sinal de erro
d = 0;
y = 0;

for cont = 1 : NUM,
    mem = [ sinT(cont) ; mem(1⊗N-1) ]; % linha de atraso do sinal de entrada
    d = sinJ(cont); % sinal desejado
    y = C(:,cont)*mem; % sinal produzido pelo filtro
    e(cont) = d - y; % erro entre sinal desejado e sinal produzido
    C(:,cont+1) = C(:,cont) + MU * e(cont) * mem / (mem'*mem); % equação de atualização
end

% resultados
figure(1);
hold on;
plot(sinT, '-r');
plot(sinJ, '-b');
hold off;
title('(1) vermelho: trilha (2) azul: usuário');
xlabel('Amostras');

figure(2);
plot(e.^2, '-b');
title('Erro quadrático');
xlabel('Iterações');

```

- **Anexo 3**: Programa em Matlab para a visualização dos resultados.

```

% -----
% Coefs.m
%
% Média dos coeficientes do erro
%
% 10/12/2003
% -----

INI = 1000;
FIM = 1399;

media = zeros(N,1);

for cont = INI : FIM,
    media = media + C(:,cont);
end
media = media / (FIM-INI+1);

```

```

figure(3);
hold on;
for cont = INI : FIM,
    plot(C(:,cont),'-c');
end
stem(media,'-k');

figure(4);
hold on;
for cont = INI : FIM,
    plot(C(:,cont),'-c');
end
plot(media,'-k');
xlabel('Iterações');

```

- **Anexo 4:** Programa em Matlab para a obtenção da resposta em frequência.

```

% -----
% Espectro.m
%
% Resposta em frequência.
%
% Márcio Costa
% 15/12/2003
% -----

NUMC = 500;      % número de amostras dos coeficientes
FAMOS = 18.2;    % frequência de amostragem

% resposta em frequência
X = abs(fft(media));
figure(6);
f = ( 0 : NUMC-1 ) * FAMOS / NUMC;
semilogx(f(1:NUMC/2),20*log10(X(1:NUMC/2)));
title('Resposta em frequência');
xlabel('Hz')
grid;

```