

## Classificador Neural *OnLine* para Sonar Passivo utilizando um Processador Digital de Sinais de Alto Desempenho

J. B. O. Souza Filho<sup>1</sup>, J. M. Seixas<sup>2</sup>, W. Soares Filho<sup>3</sup>  
<sup>1,2,3</sup> Laboratório de Processamento de Sinais - LPS - COPPE/EE/UFRJ,  
CP 68504, Rio de Janeiro 21945 – 970 Brasil  
<sup>3</sup> Instituto de Pesquisas da Marinha (IPqM), Rio de Janeiro Brasil  
Emails: nash,@lps.ufrj.br<sup>1</sup>, seixas@lps.ufrj.br<sup>2</sup>, william@lps.ufrj.br<sup>3</sup>

### Abstract

*The noise radiated from ships in the ocean contains information about their machinery, being normally used by passive sonar systems for detection and identification purposes. This article describes the implementation of an on-line neural classifier using a high-speed 32-bit floating-point digital signal processor (ADSP21062). The radiated noise from ships of four classes was received by an hydrophone, which was placed far from the ship, and digitized by an analog-to-digital converter (AD1847). Digitized signals were preprocessed for feeding the input nodes of a feedforward multilayer neural classifier. The preprocessing and the neural network were coded in Assembly language, optimized for maximum speed, and accessible through an user-friendly interface, running in an IBM-PC that hosts the DSP board. Several tests for evaluating the accuracy and speed performance of the DSP implementation were performed, comparing the prototype with an off-line implementation developed using both MATLAB scripts and C and Fortran languages.*

### 1. Introdução

Uma embarcação ao navegar no mar produz uma ampla gama de sons. O ruído do sistema de propulsão, das diversas máquinas e sistemas internos, assim como os sons produzidos pelas hélices e pela fauna aquática, misturam-se às suas múltiplas reflexões em diferentes obstáculos, compondo um rico sinal de ruído. Distinguir entre diferentes classes de navios pela análise deste ruído é uma tarefa não trivial [1].

A identificação de navios por um submarino é realizada basicamente por seu sistema de sonar passivo. O ruído irradiado pelos contatos é recebido pelos sensores do sistema de sonar e é processado de forma a fornecer uma informação visual e auditiva a um operador especialmente treinado. Este operador avalia tais informações e define a identificação do contato. A eficiência deste método está diretamente relacionada à habilidade do operador em isolar e identificar as características relevantes do sinal recebido, tanto em termos

de sonoridade, quanto em termos do conteúdo do espectro de frequências.

Visando aumentar a confiabilidade e acelerar o processo de tomada de decisões, métodos inteligentes vêm sendo utilizados. Um método especialmente interessante para esta aplicação é a utilização de redes neurais. As redes neurais têm provado ser capazes de extrair características relevantes em espaços de dados de elevada dimensão, para complexas aplicações de diferentes áreas do conhecimento [2]. Devido a sua simplicidade e robustez, as redes multicamadas e sem realimentação, treinadas pelo algoritmo de *backpropagation*, são bastante populares. Esta simplicidade, especialmente na fase de produção de saídas, permite sua implementação em diferentes ambientes computacionais, inclusive em sistemas *on-line*. Aliando-se à rede neural métodos eficientes de pré-processamento, a alta dimensionalidade do espaço de entrada de uma rede típica pode ser reduzida, redução esta que é atrativa em problemas de sonar passivo, especialmente para operação *on-line*.

Para implementação *on-line* de classificadores, os processadores digitais de sinais (DSPs) são uma ótima opção. O DSP é um microprocessador cuja arquitetura interna é otimizada para a operação de produto interno, operação mais frequentemente realizada nos algoritmos de processamento digital de sinais e de processamento neural, em especial na fase de produção de saídas. Assim, os DSPs permitem a implementação de classificadores neurais razoavelmente complexos, com alto volume de pré-processamento digital, de forma extremamente veloz, constituindo uma opção economicamente viável para problemas que exijam resposta *on-line*, ou em tempo real, de alto desempenho.

Este artigo trata da implementação, na tecnologia dos DSPs, de um sistema de classificação de navios. Realizaremos uma breve abordagem sobre o processador utilizado, a qual é seguida pela descrição do sistema de pré-processamento e do projeto da rede neural classificadora. Concluem este trabalho a descrição do processo de desenvolvimento do sistema *on-line* e os resultados obtidos para este protótipo.

## 2. O ADSP21062

O ADSP-21062 é um DSP de 32 bits de ponto-flutuante, com *clock* máximo de 40 MHz. Sua arquitetura interna é conhecida como *Super Harvard Architecture Computer* (SHARC) [3]. Esta arquitetura une uma memória cache de instruções a um duplo barramento interno (dados e programa), permitindo que uma operação de multiplicação e outra de adição sejam realizadas em um único ciclo de máquina (25 ns) - (Figura 1).

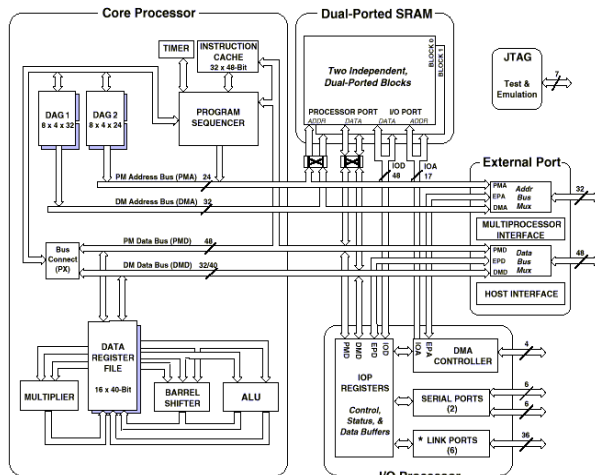


Figura 1: Estrutura do ADSP-2106x

O sequenciador de programa do SHARC permite que todas as instruções sejam condicionais e que os laços, as chamadas de subrotinas e a execução de saltos ocorram sem a introdução de ciclos de *overhead*. Quanto à programação, o SHARC possui uma linguagem *Assembly* com sintaxe lógica e suas instruções têm uma latência de execução constante e igual a dois ciclos. O paralelismo da execução das instruções é definido pelo programador na codificação da rotina, o que permite a previsão do tempo de execução da aplicação, antes mesmo de sua compilação.

### 2.1. Placa de Desenvolvimento EZ-LAB

Para o protótipo do classificador, utilizamos a placa de desenvolvimento EZ-LAB, da *Bitware Research*. Esta placa se encaixa em um barramento ISA de um computador pessoal do tipo IBM-PC e é equipada com um conversor A/D e D/A (CODEC) de 16 bits, o AD-1847. Este CODEC possui uma interface serial para transferência de dados e é totalmente configurável via *software*, realizando a conversão A/D e D/A pelo método sigma-delta [3].

A placa EZ-LAB possui ainda um conjunto de rotinas em linguagem C para o interfaceamento entre o DSP e o PC hospedeiro. Estas rotinas permitem o controle, a transferência de dados e a execução de aplicações na placa.

## 3. A Aplicação

### 3.1. Projeto do Classificador

O ruído irradiado por quatro navios de diferentes classes foi adquirido por sistema próprio, em uma raia acústica de 45 metros de profundidade. Foram realizadas 34 corridas destes 4 navios, em diferentes condições de velocidade e de operação de maquinário, originando sinais que foram amostrados por um A/D de 8 bits a 22,05 kHz.

Uma vez que a classificação dos navios é baseada no conteúdo de frequências do ruído adquirido e que a maioria do ruído produzido pelo maquinário típico de operação destes navios se encontra na faixa de DC a aproximadamente 2,5 kHz, esta faixa foi escolhida como faixa relevante para a tarefa de classificação [1]. Para tal faixa, um sistema de pré-processamento foi desenvolvido [4].

Na cadeia de pré-processamento (Figura 2), o sinal adquirido é processado por um filtro passa-baixa e sua taxa de amostragem é reduzida por 3, atingindo 7,35 kHz. Este filtro utiliza uma aproximação elíptica de oitava ordem, implementada por 4 seções biquadráticas, e possui frequência de corte em 2,87 kHz e atenuação de 60dB em 4,48 kHz, garantindo a não ocorrência do fenômeno de *aliasing* na banda passante.

Cada bloco de 2048 amostras do sinal resultante é multiplicado por uma janela de *Hanning* e o módulo de sua transformada de *Fourier* é calculado. Este módulo é submetido a uma nova filtragem e decimação, resultando em 400 pontos de espectro. Cada espectro corresponde a uma janela de aquisição de aproximadamente 278 milissegundos.

O espectro de 400 pontos é submetido a uma filtragem passa-alta usando o algoritmo *two-pass-split-window* (TPSW). Este algoritmo realiza uma estimativa da média local que, se utilizada para a normalização dos espectros, promove um realce de seus picos, aumentando significativamente a capacidade de discriminação do classificador. Os picos espectrais contêm informações diretamente relacionadas às máquinas em operação no interior dos navios [1].

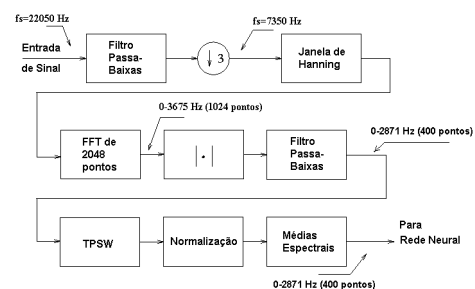


Figura 2: Cadeia de pré-processamento

Por fim, médias envolvendo os espectros resultantes são realizadas. Este procedimento melhora a relação

sinal-ruído e aumenta também a eficiência de discriminação, em especial quando utilizados na fase de produção de saídas da rede neural [5].

A rede neural classificadora é uma rede de 3 camadas, possuindo 400 nós de entrada, 6 neurônios na camada escondida e 4 neurônios na camada de saída. A cada neurônio de saída foi atribuída uma classe e a rede foi treinada pelo algoritmo de *backpropagation*. Foram adotados os valores-alvo binários +1 e -1, onde +1 significa neurônio de saída ativo e -1, neurônio inativo. Todos os neurônios possuem como função de ativação a tangente hiperbólica e para determinar a classe atribuída pelo classificador, o critério de máxima probabilidade foi utilizado [2].

### 3.2. Implementação

Todo estudo (*off-line*) realizado para o dimensionamento e configuração da cadeia de pré-processamento, assim como para treinamento e teste da rede neural, foi desenvolvido através de programas codificados em linguagem Fortran e de *scripts* MATLAB.

O objetivo do protótipo aqui descrito é implementar o sistema de identificação desenvolvido visando operação *on-line*. Em virtude do volume de processamento envolvido e da necessidade de respostas rápidas por parte do classificador, optou-se por seu desenvolvimento em linguagem *Assembly* do DSP descrito.

A linguagem *Assembly* permite implementações eficientes do ponto de vista de tempo de processamento, principalmente se as características do dispositivo alvo forem corretamente exploradas para um dado algoritmo particular. A implementação ótima de um algoritmo em uma plataforma alvo, no entanto, demanda um maior tempo de desenvolvimento, aumenta a complexidade da implementação e diminui a portabilidade do código desenvolvido.

Dado o custo de programação envolvido no desenvolvimento, optou-se por máxima otimização em termos de velocidade de processamento, tendo em mente porém, a constituição de rotinas genéricas, facilmente reconfiguráveis. O sistema constituído é flexível, facilitando futuras modificações relacionadas à incorporação de novos recursos. Seu desenvolvimento permitiu a constituição de uma biblioteca de funções de processamento de sinais otimizadas para a plataforma escolhida.

Em virtude do compromisso da aplicação final com a velocidade, o uso de *Look-Up-Tables (LUTs)* foi extensivamente aplicado. As LUTs são tabelas que armazenam em memória os valores de funções de interesse para a aplicação. Caso estas funções viessem a ser calculadas, seja por séries ou por métodos de aproximação sucessiva, vários ciclos de máquina seriam exigidos.

As LUTs são especialmente indicadas para algoritmos do tipo janelamento, cálculo de FFT, entre outros.

Nestes algoritmos, valores são multiplicados por constantes pré-estabelecidas. Estas constantes podem ser armazenadas em uma tabela e sua consulta é normalmente bastante rápida, reduzindo o tempo de processamento. As LUTs são úteis na substituição do cálculo de funções periódicas e assintóticas, tais como a tangente hiperbólica [6].

Cada etapa do processamento, desde a aquisição até a rede neural, foi implementada separadamente, através de rotinas próprias. A integração, gerenciamento e comunicação entre as rotinas foi realizada por uma outra rotina específica, chamada *sistema*. Cabe também a ela, o interfaceamento entre o classificador no DSP e o PC hospedeiro da placa de desenvolvimento.

O sistema de aquisição é responsável pela configuração das portas seriais, do controlador de DMA e do vetor de interrupções do ADSP-21062, estabelecendo uma correta comunicação entre o CODEC (AD-1847) e o DSP. Além desta configuração, o sistema de aquisição manipula as amostras recebidas, armazenando-as para processamento futuro.

Para cada amostragem realizada pelo AD-1847, três palavras são produzidas e transmitidas serialmente para o DSP, duas relacionadas às amostras (canais direito e esquerdo) e outra relacionada ao controle. Para cada conjunto, o DMA realiza uma interrupção que chama a rotina de recepção de dados do sistema de aquisição. Esta rotina contabiliza uma nova amostra recebida, realiza uma subtração de *offset* a esta amostra, se este recurso estiver ativado, e armazena a amostra correspondente a um dos canais em um dos *buffers* de aquisição, no formato de 16-bits ponto-flutuante. Os *buffers* são preenchidos em seqüência e quando um deles está cheio, uma *flag* específica é habilitada.

A rotina *sistema* fica em ciclo de espera até sua habilitação (sinalização por varredura). Quando habilitada, um bloco de amostras está disponível, e este é copiado por *sistema* para a região de entrada do primeiro filtro. Completados dois blocos de amostras, as diversas etapas de processamento são chamadas, na forma de sub-rotinas.

A primeira filtragem passa-baixa da cadeia de pré-processamento é realizada por um algoritmo que implementa um filtro genérico biquadrático em cascata. A filtragem é realizada em blocos de amostras de tamanho configurável e o algoritmo realiza a manutenção dos estados necessários a uma correta filtragem, a despeito do processamento ser realizado em blocos [7].

A decimação é realizada por meio de uma leitura e escrita promovida através de dois ponteiros dos geradores de endereço do DSP. O ponteiro de leitura é ajustado para promover a leitura de  $k$  em  $k$  amostras, onde  $k$  é o fator de decimação, enquanto que o ponteiro de gravação armazena as amostras em seqüência. É definido um buffer circular em *software*, permitindo uma decimação correta para sinais contínuos, ainda que realizada em blocos.

O janelamento de *Hanning* é realizado através de uma LUT armazenada em memória, cujos valores são multiplicados pelo conjunto de amostras recebidas. Diferentes janelas podem ser aplicadas, tanto em relação ao formato, quanto ao número de amostras envolvidas.

A FFT de 2048 pontos é calculada pelo algoritmo de raiz-2 [7]. Uma vez que as amostras são reais, realiza-se o empacotamento, método que consiste em distribuir metade das amostras na entrada real da FFT e a outra metade na entrada imaginária, realizando apenas uma FFT de 1024 pontos, cujas saídas, se devidamente combinadas, passam a apresentar os mesmos valores da primeira FFT [8]. O recurso de *bit-reversing* dos geradores de endereço também é utilizado para que as saídas se encontrem corretamente ordenadas.

O cálculo do módulo é realizado através do produto de registradores. Para o cálculo da raiz quadrada é realizado um algoritmo de aproximação sucessiva (Newton Rapson), que fornece, em apenas duas iterações, o valor preciso para esta função [9].

A filtragem passa-baixa do espectro é realizada através de um filtro FIR. Este filtro foi integrado ao decimador que o segue, sendo realizada primeiramente a decimação e, posteriormente, a filtragem. A possibilidade da troca de ordem destes blocos reduziu o volume de processamento [7]. Neste algoritmo, são configuráveis o fator de decimação e a ordem do filtro utilizado, assim como seus coeficientes.

A estimação da média local (TPSW) é realizada para cada espectro por uma convolução das entradas com uma janela constante e simétrica. É realizada uma primeira convolução, seguida por um mecanismo de comparação, que busca evitar polarizações da média pelos picos do espectro, e, finalmente, uma segunda convolução [1]. Esta média é utilizada na normalização do espectro, que é implementada através da multiplicação de cada ponto espectral por uma constante, multiplicação esta que é seguida por uma divisão pela média local correspondente.

A implementação do TPSW foi otimizada, especialmente no cálculo da convolução. À medida em que a janela se desloca no conjunto de amostras, uma nova convolução é gerada. A otimização consiste em determinar o valor da nova convolução apenas pela inclusão e exclusão de pontos associados aos extremos das janelas, reduzindo sensivelmente o número de operações de cálculo envolvidas. Esta otimização é bastante atrativa para o uso de janelas largas, como a utilizada neste protótipo (100 pontos). Na normalização, a divisão foi implementada por um algoritmo de aproximações sucessivas, baseado numa aproximação inicial de 8 *bits* de precisão fornecida por instrução específica [9].

A média dos espectros é realizada através de uma acumulação dos vários espectros envolvidos em um *buffer* específico, que é limpo a cada média realizada. A cada espectro produzido, o algoritmo lê a soma dos espectros anteriores (ponto-a-ponto), adiciona a esta

soma o novo espectro e atualiza este *buffer* com o novo valor da soma. Caso o número de espectros acumulados seja menor que o envolvido nas médias, um desvio é realizado e todas as etapas de pré-processamento anteriores são novamente procedidas. Em caso negativo, o conteúdo do *buffer* de acumulação é normalizado por  $1/N$ , onde  $N$  é o número de espectros envolvidos na média, habilitando-se o início do processamento neural.

A codificação neural implementa uma rede neural genérica de duas camadas na fase de propagação de entradas [2]. Para implementação da função de ativação dos neurônios (tangente hiperbólica), foi utilizada uma LUT. Um estudo realizado em aplicações de classificadores neurais *online* para físicas de partículas [10] conduziu à escolha de uma tabela simétrica, de resolução constante, com saturação em 4 e com 1025 posições (512 positivas, 512 negativas e um zero). A rede neural com esta LUT manteve a mesma capacidade de discriminação da rede *off-line* desenvolvida, cuja tangente hiperbólica foi calculada por diretiva de *software* em dupla precisão.

Para o interfaceamento entre o PC e o classificador, é também utilizado o método de varredura. O PC fica em estado de espera até que uma posição específica da memória do DSP seja habilitada e os resultados possam ser coletados de sua memória.

#### 4. Resultados Obtidos

Foram realizados vários testes a fim de se verificar a acuidade numérica da aplicação desenvolvida no DSP (referida como *on-line*) e seu tempo de processamento. Em virtude do uso de LUT na implementação da função de ativação da rede neural, o teste de acuidade foi realizado em duas etapas: uma primeira, envolvendo apenas a cadeia de pré-processamento; e uma segunda, envolvendo o pré-processamento mais a rede neural.

Os testes de acuidade foram realizados por uma aplicação desenvolvida em linguagem C que promove a leitura dos arquivos das corridas, realiza a transferência de blocos de amostras do arquivo para a entrada do primeiro filtro, comanda a execução do processamento e armazena em disco os resultados. Os resultados coletados (saídas da cadeia de pré-processamento e da rede neural) são comparados com os dados da análise *off-line* (codificação em alto-nível: MATLAB e linguagem Fortran), gerando as estatísticas de acuidade. Este procedimento busca não só verificar uma correta implementação das rotinas, assim como mensurar o impacto do uso de precisão simples na aritmética dos algoritmos (a análise *off-line* foi realizada em precisão-dupla) e da utilização da LUT escolhida na rede neural classificadora.

Para ambas estatísticas, o conjunto das 34 corridas foi utilizado, o que totaliza 12741 janelas de espectro, onde uma janela corresponde a uma identificação. O re-

curso de médias espectrais não foi utilizado, embora implementado e testado. As estatísticas de acuidade são baseadas na distribuição do módulo da diferença entre os valores obtidos nas implementações *on-line* e *off-line* (erro absoluto).

Para a cadeia de pré-processamento, o histograma da Figura 3 e as estatísticas da Tabela 1 foram obtidos. Podemos perceber que os erros médios se encontram na oitava casa decimal, o que corresponde aos 32-bits de precisão do DSP. Vale lembrar que os valores típicos obtidos na análise *off-line* se encontram na faixa de 0,04 a 0,3

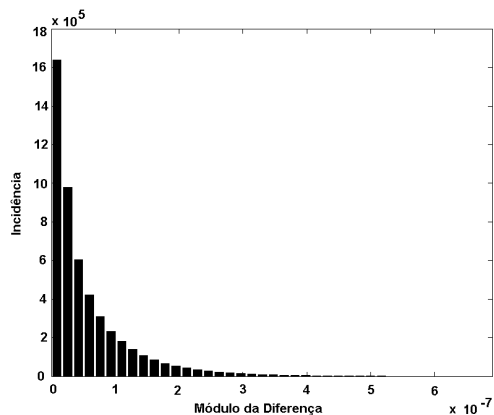


Figura 3: Histograma dos erros (*online x offline*) da Cadeia de Pré-processamento

Tabela 1: Estatísticas de erros (pré-processamento)

|                 | Valor:    |
|-----------------|-----------|
| <b>Máx:</b>     | 3,96 E-06 |
| <b>Médio:</b>   | 5,96 E-08 |
| <b>Mediana:</b> | 3,24 E-08 |
| <b>RMS:</b>     | 7,85 E-08 |

Considerando o erro absoluto entre as saídas das redes neurais (*on-line x off-line*), o histograma da Figura 4 e a estatísticas da Tabela 2 apresentam um erro médio na quarta casa decimal, erro diretamente relacionado ao uso da LUT na implementação da função de ativação da rede neural. Os valores típicos de saída da rede são, em módulo, próximos a um, logo o erro relativo médio cometido é de aproximadamente 0,06 % .

Tabela 2: Estatísticas de erro (rede neural + pré-processamento)

|                 | Valor:    |
|-----------------|-----------|
| <b>Máx:</b>     | 2.22 E-02 |
| <b>Médio:</b>   | 5.52 E-04 |
| <b>Mediana:</b> | 5.99 E-04 |
| <b>RMS:</b>     | 9.06 E-04 |

Como próximo passo, buscou-se verificar o impacto da diferença numérica das implementações na eficiência de discriminação. Para tal levantamento, foram obtidas as eficiências descritas na Tabela 3. Cada efi-

ciência foi calculada, tanto no sistema *off-line* quanto no *on-line*, como a razão do número de janelas corretamente identificadas pelo número total de janelas espectrais correspondentes a uma classe. Através da análise destes dados, verificamos que as diferenças numéricas não provocaram impacto significativo na capacidade de discriminação da aplicação *on-line*, resultando em uma eficiência total média de 96,5 %.

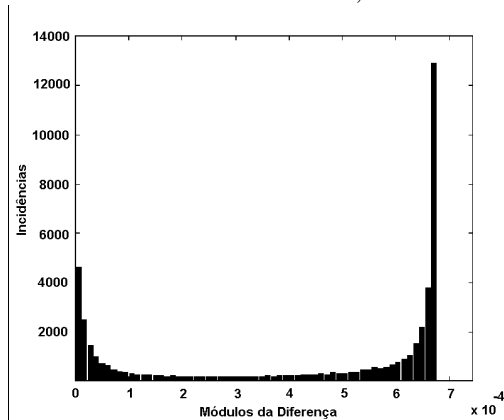


Figura 4: Histograma dos erros (*on-line x off-line*) da rede neural

Como teste final, procedemos a simulação das condições reais de operação do protótipo. Por facilidades operacionais, esta simulação foi realizada através da reprodução dos arquivos (digitalizados) das corridas pela placa de som do computador, a qual foi conectada, através de sua saída analógica, à entrada do conversor da placa de desenvolvimento. A reprodução do sinal das corridas, seguida por sua reamostragem pela placa EZ-LAB, provoca a degradação do sinal e introduz distorções causadas pelas não-linearidades da placa de som e do AD-1847. Estas distorções alteram os espectros vistos pela rede e tendem a acarretar uma redução da eficiência de discriminação.

Tabela 3: Comparação das Eficiências de Discriminação (*on-line x off-line*)

| Classes         | Eficiência Off-line | Eficiência On-line |
|-----------------|---------------------|--------------------|
| <b>Classe 1</b> | 99,01 %             | 99,01 %            |
| <b>Classe 2</b> | 96,67 %             | 96,69 %            |
| <b>Classe 3</b> | 93,79 %             | 93,68 %            |
| <b>Classe 4</b> | 96,34 %             | 96,34 %            |

Novamente foi desenvolvido um sistema em C que promovia o armazenamento dos espectros de entrada da rede neural e promovia o teste de eficiência da identificação baseada no ruído adquirido. Através dele, as distriuições das saídas dos neurônios para as diversas classes foi realizada, originando o gráfico da Figura 5. Nesta matriz de histogramas, as colunas representam as distribuições de cada neurônio de saída para eventos restritivos a cada classe, definidos pelas linhas da matriz. Portanto, para uma discriminação ótima, os

histogramas da diagonal principal devem ter distribuições próximas ao valor alvo (1), indicando a atividade do neurônio, enquanto que os demais histogramas de-

vem indicar a inatividade do neurônio (valores próximos a -1).

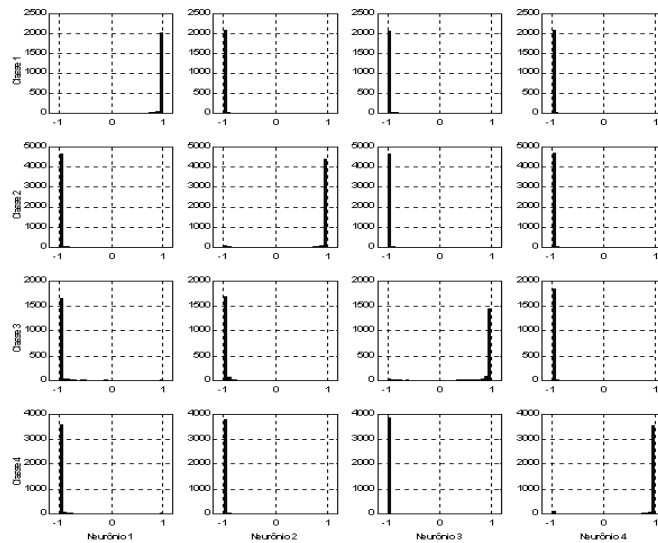


Figura 5: Distribuição das Saídas da Rede para as Diversas Classes

A Tabela 4 confirma que há uma redução da eficiência de discriminação, embora esta não seja significativa. A eficiência total média é de 94,5%, apenas dois pontos percentuais abaixo da implementação *off-line*.

Tabela 4: Eficiências de Discriminação (aquisição)

| Classes  | Corretos | Total | Eficiência |
|----------|----------|-------|------------|
| Classe 1 | 1994     | 2104  | 94,77 %    |
| Classe 2 | 4547     | 4788  | 94,97 %    |
| Classe 3 | 1697     | 1823  | 93,08 %    |
| Classe 4 | 3639     | 3856  | 94,37 %    |

Quanto ao tempo de processamento, toda a aplicação demanda aproximadamente 320.000 ciclos de máquina (7,8 ms), para uma identificação por janela de espectro. Este tempo é plenamente satisfatório para a operação real do protótipo.

#### 4. Conclusão

Um sistema *on-line* de classificação de navios foi apresentado. Sua implementação foi realizada em um DSP de ponto-flutuante de alta performance, o ADSP-21062, através de algoritmos desenvolvidos em linguagem *Assembly*. Várias otimizações foram empreendidas, principalmente o uso extensivo de LUTs, para máxima velocidade de processamento. Análises de acuidade, performance e eficiências foram desenvolvidas, comprovando a adequação do DSP escolhido para a aplicação. O sistema *online* reproduziu satisfatoriamente a eficiência do sistema *off-line* numa simulação da operação real. Eficiências ainda maiores de

classificação podem ser obtidas se o recurso de médias espectrais for explorado.

#### 4. Agradecimentos

Os autores gostariam de agradecer o apoio do CNPq, da FAPERJ e do IPqM dados a este projeto.

#### Referências

- [1] R.O. Nielsen, *Sonar Signal Processing*, Artech, 1991.
- [2] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Second Edition. Prentice Hall, 1999.
- [3] Analog Devices, *DSP/MSP Products Reference Manual*, 1995
- [4] J. M. Seixas, W. Soares Filho, J. B. O. Souza Filho, D. O Damazio, N.N. Moura, *A Compact Online Neural System for Classifying Passive Sonar Signals*, ICSPAT 99, EUA, 1999.
- [5] W. Soares Filho, J.M Seixas, L. P. Calôba, *Averaging Spectra to Improve the Classification of the Noise Radiated by Ships Using Neural Network*, SBRN 2000, Brasil, 2000
- [6] Texas Instruments, *Table Lookup and Interpolation on TMS320C2xx*, Application Report, 1996.
- [7] A. V. Oppenheim e R.W Schafer, *DiscreteTime Signal Processing*, Prentice Hall, 1989.
- [8] Texas Instruments, *Implementing Fast Fourier Transform of RealValued Sequences with the TMS320 DSP Family*, Application Report, 1997.
- [9] Analog Devices, *ADSP21000 Family Application Handbook, Volume 1*, 1995.
- [10] J. B. O. Souza Filho, J. M. Seixas, Implementação de Classificadores Neurais na Tecnologia de Processadores Digitais de Sinais, IV CBRN, Brasil, 1999.