

# UM SISTEMA PROCESSADOR DE SINAIS PARA COMUNICAÇÃO ACÚSTICO-SUBMARINA

Paulo Osorio, Jacques Szczupak, Marcos V. Nóbrega,  
Alexandre Yamaoka, Paulo Pereira e Rogerio Alves

DEE-PUC-RIO - Caixa Postal 38063

22452-970 - Rio de Janeiro, RJ

Fax: (021)511-5154 , e-mail: osorio@ele.puc-rio.br

**RESUMO** - Este trabalho descreve um sistema acústico de telemetria, especificado para operar na faixa de 10 a 13 kHz, com uma banda máxima de 3 kHz, para ser utilizado na comunicação entre uma plataforma de petróleo e seus poços. O sistema foi implementado para operação em tempo real, usando duas placas DSP da ASPI, baseadas no processador TMS 320C25. Vários testes de laboratório foram realizados, inclusive transmissões em tempo real, na presença de ruído. Os resultados preliminares se mostraram satisfatórios, indicando a possibilidade da utilização do sistema proposto para a comunicação acústico-submarina entre estações fixas.

**ABSTRACT** - This paper describes an acoustic telemetry system specified to operate in the 10-13 kHz band with a maximum bandwidth of 3 kHz, for use in communication between offshore oil platforms and their wells. The system was implemented for real time operation using two ASPI DSP boards, based on the TMS 320C25 microprocessor. Several real time laboratory experiments were performed in the presence of noise. The preliminary results are satisfactory, indicating the feasibility of the proposed system for underwater acoustic communication between fixed stations.

**Palavras Chave:** Modem, MFSK, Processamento Digital de Sinais, e Comunicação Acústica-Submarina.

## 1. INTRODUÇÃO

A comunicação entre uma plataforma de petróleo e os diversos poços submarinos é feita atualmente por meio de cabos elétricos. Este método traz uma série de desvantagens, pois os cabos, além dos problemas mecânicos a que estão sujeitos, depois de um certo tempo começam a apresentar deterioração em suas características elétricas, o que eventualmente requer constantes substituições.

Devido ao aumento significativo da exploração de petróleo no mar começaram a surgir soluções alternativas para esse problema através de técnicas de comunicação

acústica [1] [2]. A telemetria acústica utilizada em submersíveis vem sendo estudada desde que foi reconhecido que o mar pode suportar a transmissão de sinais [3]. Entretanto, o mar tem-se mostrado um meio extremamente difícil para esta tarefa, principalmente quando se desejam altas taxas de transmissão. Como se sabe, para transmitir em altas taxas é necessário um canal com banda larga e/ou um canal muito limpo.

Na prática isso não acontece, pois a largura de banda é limitada pela absorção das frequências altas do sinal acústico e o mar é um canal reverberante com espalhamentos tanto em frequência como no tempo. Isto tudo requer um estudo muito cuidadoso das características do canal acústico para sua correta utilização. Felizmente a disponibilidade de dispositivos em VLSI, como por exemplo os *DSPs* (*Digital Signal Processors*), com capacidade computacional muito alta, permitem a implementação de sistemas processadores de sinais para comunicação acústico-submarinas, capazes de compensar muitos dos problemas causados pelas características do canal.

Este trabalho tem como objetivo o desenvolvimento de um processador de sinais para comunicação acústico-submarina entre plataformas de petróleo e os diversos poços produtores situados num raio de 10 Km e a profundidades variadas. Os dados de telemetria trocados entre a plataforma e os poços, incluem diversas informações de status como: temperatura, pressão, posição de diversas válvulas, etc. Além disto, sinais de comando serão transmitidos da plataforma para os poços, como os para a abertura e fechamento de válvulas.

O sistema descrito neste trabalho foi especificado para operar com uma largura de banda até 3 kHz, centrada na faixa de 10-13 kHz, tendo *Relação Sinal Ruído* (*RSR*) mínima de 20 dB. A taxa de transmissão requerida é baixa, dado o tipo de controle e de medições efetuados. Entre 100 a 200 bps seriam suficientes, mas, como se verá na sequência deste trabalho, esta taxa foi substancialmente aumentada por conveniência do projeto.

Devido às características do canal submarino, a codificação deve independer de informações associadas à fase do espectro do sinal, já que esta pode variar

substancialmente. Mesmo a magnitude do espectro pode sofrer variações abruptas ao longo do tempo, consequência de atenuações no canal com variação rápida. Estas restrições induzem naturalmente a uma codificação baseada em frequência e à utilização de uma taxa de transmissão alta o suficiente para não depender das flutuações da atenuação do canal marinho.

## 2. SISTEMA PROCESSADOR DE SINAIS

O Sistema Processador de Sinais desenvolvido consiste, em cada extremo do canal, de um transmissor e de um receptor associados a filtros passa-banda. Com exceção de amplificadores e dos filtros de entrada e saída, todas as tarefas são implementadas em software. Estas tarefas incluem: codificação, modulação, sincronização demodulação, e detecção. Elas serão descritas na sequência do trabalho dentro dos módulos que compõem o sistema.

### 2.1. Transmissor

Em telemetria acústica submarina, onde se requer alto grau de confiabilidade, a escolha do tipo de modulação é de suma importância. No presente caso, o método escolhido foi o *MFSK* ( *Multiple Frequency Shift Keying* ) por sua simplicidade e robustez [2], [3]. Este método pode ser visto como um arranjo de ordem  $M$  da técnica FSK. Uma codificação em frequência, portanto, como desejado.

Para implementar o MFSK, optou-se por utilizar palavras de 8 bits, um byte, sendo cada bit associado a dois tons (redundância a ser utilizada no futuro como um reforço à detecção de erros). Considerando que sinais reais possuem tons aos pares com componentes positiva e negativa em frequência, o byte tem seu espectro associado a 32 tons.

A velocidade de transmissão depende da largura de banda disponível e do espaçamento entre os tons,  $\Delta f$ . Para a maior largura de faixa especificada, 3 kHz, resulta um espaçamento máximo entre tons  $\Delta f_M = 93,75 \text{ Hz}$ . Apesar de apropriado, o espaçamento entre tons foi reduzido, limitando-se o valor de  $\Delta f$  a 70 Hz. Duas considerações devem ser feitas sobre esta escolha. Primeiro, uma separação de 70Hz é suficiente para manter a identificação de tons vizinhos, mesmo que estes possam sofrer pequenos deslocamentos ao longo do canal). Segundo, a largura da banda base fica restrita a 2240 Hz. Esta largura de faixa vai permitir explorar a periodicidade do espectro do sinal discreto, evitando com isto um estágio clássico de modulação banda base, que eventualmente sobrecarregaria a recepção.

O tempo de transmissão para um conjunto de tons é de  $1/\Delta f$ , e portanto para o espaçamento entre tons de 70 Hz resulta em uma duração do quadro de informação (1 byte) de 14,29 ms. Isto corresponde a uma taxa de transmissão de dados de 560 bps, que é mais do que o suficiente para a aplicação em vista [4].

A codificação é executada byte a byte. Em um destes bytes, o mapeamento de cada bit no espectro de frequência correspondente, pode ser feito de várias maneiras, dependendo inclusive do protocolo a ser utilizado. Nesta primeira fase da concepção do projeto, escolheu-se um mapeamento conforme indicado na Figura 1. Note-se que, por segurança, os bits são transmitidos em duas posições do espectro de frequência, tendo valores complementares. Observa-se ainda que, se uma das componentes do mapeamento estiver localizada junto ao extremo da faixa (região presumidamente mais sensível a ruídos de canais vizinhos), a outra estará junto ao centro do canal (região mais protegida). Esta redundância na informação e as características da codificação complementar deverão ser exploradas no futuro para a detecção e localização de erros.

Desta forma, para cada byte de informação é atribuído um espectro em frequência para o quadro correspondente. Como pode ser observado, ainda há dois graus de liberdade na determinação deste espectro, representados pela energia e fase de cada componente em frequência. A escolha adotada foi de uniformizar a energia das diversas componentes em frequência, mas o mesmo não pode ser feito com a fase.

Devido às características do canal acústico-submarino, a fase de cada componente poderá estar totalmente deformada ao alcançar o receptor, não se prestando para transmitir informação. Por outro lado, a escolha adequada das fases no transmissor pode evitar picos de energia na saída do mesmo. Estes picos de energia são indesejáveis devido às naturais restrições na potência do sistema de alimentação, especialmente para equipamentos no fundo do mar. A escolha das fases de cada componente do espectro atende a limitar o pico de energia, adicionando ainda a restrição de prover uma transição suave entre os quadros, evitando assim um indesejável transbordo do espectro para fora da banda especificada. Cada byte tem um conjunto ótimo de fases para atender a estes objetivos, o que será abordado mais tarde.

Especificadas as fases, o quadro banda base está pronto para ser montado no domínio da frequência. Ele deverá então ser mapeado para o domínio do tempo, forma em que será transmitido. Isto é feito a partir do seu espectro em frequência, por meio de uma FFT inversa de 32 pontos.

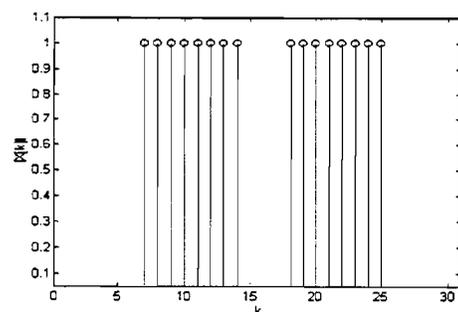


Figura 1 - Magnitude do Espectro correspondente à palavra  $(01)_H$ .

## Interpolação como Modulação

O sistema foi especificado para operar numa faixa de dezenas de kHz, o que indica a necessidade de se modular a banda base. Entretanto, ao invés de se utilizar o processo clássico de multiplicação por senóide, optou-se por uma interpolação direta do sinal por 5, deslocando-se numa das imagens replicadas em frequência o centro da faixa banda base para 11,2 kHz. Para isto, ao invés de se utilizar a solução clássica (interpolação no tempo e na frequência seguida de decimação, de difícil implementação usando FFTs de base 2), optou-se por explorar o pré-conhecimento do espectro do sinal a ser transmitido, principalmente de suas fases iniciais.

Aplicando-se a cada componente do sinal um correspondente incremento de fase, chega-se à próxima amostra do sinal interpolado, repetindo-se o processo até que quatro amostras sejam geradas e inseridas na sequência original, compondo a nova sequência interpolada por 5. Estes incrementos de fase correspondem a deslocamentos iguais no tempo em cada uma das senóides do espectro. Com este processo podem-se obter interpolações inteiras de qualquer ordem, o que o torna bastante flexível [4].

Portanto, o algoritmo que implementa o processo de *interpolação parte do espectro original de um dado* quadro, onde cada componente,  $i$ , tem fase conhecida. A seguir computa os correspondentes desvios de fase,  $\theta_{ik}$ , onde  $k$  identifica o número de pontos a serem interpolados (4 neste caso de interpolação por 5). A partir daí são gerados os blocos espectrais e para cada um deles é computada uma FFT inversa. O quadro interpolado terá  $32 \times 5$  amostras, sendo a primeira amostra, a primeira do primeiro bloco (sem desvio de fase), a segunda a primeira do segundo bloco, e assim sucessivamente. Com a interpolação por 5, uma das imagens do espectro aparecerá na faixa de 10.080Hz a 12.320Hz, centrada em 11.200Hz. Cada quadro terá, portanto, 160 amostras [4].

As principais vantagens deste método são:

- a utilização de FFTs de apenas 32 pontos;
- a reconstituição precisa e muito simples do sinal
- a possibilidade de interpolar o sinal por qualquer número inteiro, o que permite em combinação com a largura de banda alcançar a vizinhança da portadora desejada.

## Determinação das Fases Iniciais

Dado que foi fixada uma mesma energia para cada frequência do espectro, fica-se apenas com um grau de liberdade que é o controle das fases iniciais. Por exemplo, não é possível zerar todas as fases pois haveria a ocorrência de picos no sinal a ser transmitido, que estão associados a picos de consumo máximo de energia por parte do transmissor. Isto não é aconselhável para equipamentos submersos que são alimentados por baterias

de difícil substituição. Deve-se notar que o consumo total de energia/quadro independe da escolha da fase inicial das componentes dos espectros, o que não acontece com a potência. Assim, um dos objetivos na escolha das fases iniciais é distribuir a potência da maneira mais uniforme possível ao longo do período de transmissão de cada quadro. O outro, será o de prover uma transição suave entre os quadros.

No protocolo adotado, a transmissão de uma mensagem começa com um quadro de sincronização, que nada mais é que o envio de uma senóide na frequência da portadora (algo no entorno de 11,2 kHz), habilitada durante 14,29 ms. Este quadro é seguido por um pacote formado por 8 quadros com dados. A Figura 2 mostra um quadro típico de dados, já interpolado, com 160 amostras.

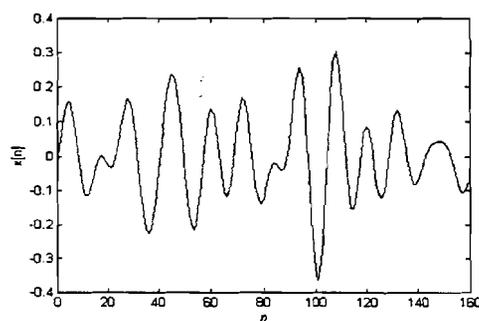


Figura 2 - Forma de onda de um quadro de dados correspondente à palavra (01)<sub>H</sub>.

Desta forma, o sinal no tempo será formado por uma sucessão de pacotes de informação, intercalados por quadros de sincronização. Os quadros de sincronização são formados por uma senóide portadora, que inicia com amplitude zero e com inclinação positiva e termina também em zero com a mesma inclinação. Afim de se evitar um espalhamento do espectro para fora da banda estabelecida, é necessário que haja uma transição suave entre todos os quadros, como o indicado na Fig. 2. Para que isto seja possível, desenvolveu-se um algoritmo que utiliza o conjunto de fases iniciais, como parâmetros de um processo de otimização tipo minmax, assegurando que cada quadro inicia com amplitude zero e termina com o mesmo valor, mas com uma equitativa distribuição em amplitude dos picos.

A Figura 3 apresenta um sinal modulado visto na saída do conversor A/D do receptor, consistindo de um quadro de sincronismo seguido por oito quadros iguais de dados. A hipótese que esta suavização entre quadros limitaria a largura do canal utilizado ao valor especificado foi comprovada em laboratório.

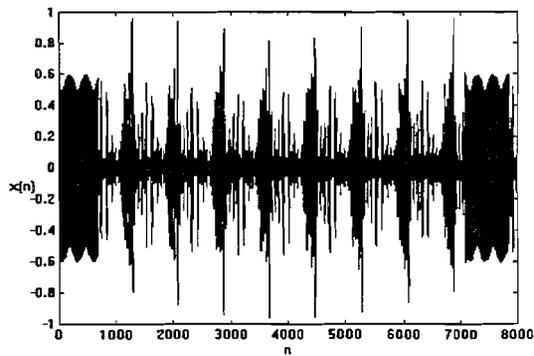


Figura 3 - O sinal modulado visto na saída do conversor A/D do receptor.

## 2.2. Receptor

O receptor apropriado para recuperar a informação transmitida deve realizar três operações básicas: sincronização, demodulação, e detecção. A demodulação incoerente do quadro *MFSK* transmitido requer sincronizações externas. A sincronização de símbolos serve para indicar o início e o final de um pacote, enquanto a sincronização de portadora fornece informações precisas desta frequência. Ambas informações serão utilizadas mais tarde na etapa de demodulação. O receptor trabalha com uma taxa de amostragem de 5 vezes a frequência da portadora, que é na realidade de 11.200,72 Hz, devido a limitações no hardware da placa DSP.

### Sincronização de Portadora

Para a sincronização de portadora foram estudados três tipos de filtros digitais adaptativos: filtros preditivos de 2ª ordem, utilizando algoritmos LMS (Least Mean Square), RLS (Recursive Least Square) e uma estrutura baseada num filtro notch IIR de 2ª ordem, que usa o método do gradiente para a detecção de senóides [5]. O desempenho dos algoritmos baseados em preditores se mostrou muito dependente do ruído aditivo presente nas senóides. Devido à baixa *RSR* inerente às transmissões acústicas submarinas, os algoritmos LMS e RLS não se mostraram adequados a esta aplicação em particular. O algoritmo com melhor desempenho foi o do filtro notch IIR. A estrutura geral do filtro notch adaptativo é a apresentada na Figura 4 [5].

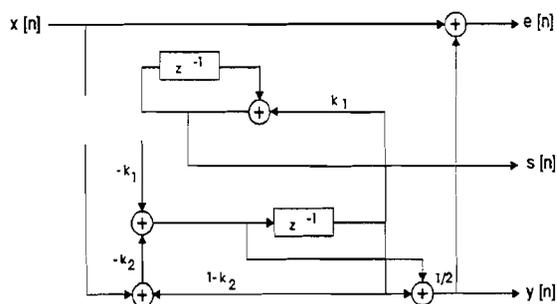


Figura 4 - Diagrama de fluxo do filtro notch.

A função de transferência desta estrutura é dada por: [4]

$$H_N(z) = \frac{2-k_2}{2} \frac{1 - [2(2-k_2-k_1^2)/(2-k_2)]z^{-1} + z^{-2}}{1 - (2-k_2-k_1^2)z^{-1} + z^{-2}}$$

Nessa rede, a frequência do notch pode ser alterada através da variação de  $k_1$  e a sua seletividade é definida através da escolha de  $k_2$  (fixo durante a adaptação). O algoritmo adapta o coeficiente  $k_1$  de modo a minimizar o valor médio quadrático da saída  $e[n]$  do notch. A adaptação de  $k_1$  é feita através de um algoritmo baseado em gradiente normalizado, com equação de atualização

$$k_1[n+1] = k_1[n] - \mu \frac{e[n]s[n]}{p[n] + p_{\min}}, \quad (1)$$

onde  $s[n]$  é o gradiente de  $e[n]$  com respeito a  $k_1[n]$ ,  $p[n]$  é a energia estimada do gradiente  $s[n]$ ,  $\mu$  o coeficiente de adaptação, e  $p_{\min}$  é uma pequena constante positiva para prevenir singularidades.

A energia do gradiente pode ser estimada recursivamente usando

$$p[n+1] = \lambda p[n] + (1-\lambda) |s[n]|^2,$$

onde  $\lambda$  é um fator de esquecimento. O coeficiente  $k_2$  está relacionado com o raio  $r$  do par de polos complexo conjugados do filtro notch por  $k_2 = 1 - r^2$ .

Para uma entrada senoidal, de frequência  $\omega_0$ , o erro médio estimado  $E[e^2[n]]$  é minimizado por

$$\hat{k}_1 = 2\sqrt{(1-k_2/2)} \sin(\omega_0/2).$$

Sendo assim, através da monitorização do valor de  $k_1$  é possível a identificação da frequência da portadora, permitindo ajustar o sincronismo.

O processo de identificação da frequência da portadora é inicializado a 20% da frequência de amostragem do receptor, o que corresponde a uma taxa de 5 amostras por período da portadora. Considerando os limites para o deslocamento de frequência introduzido pelo meio de transmissão, como 50Hz, e a diferença máxima entre as frequências de oscilação dos cristais do transmissor e do receptor, como 10 Hz, o afastamento máximo para efeitos de sincronização será de 60 Hz.

São três os principais fatores que influenciam na convergência do algoritmo de sincronização: a *RSR*, o coeficiente de adaptação  $\mu$ , e o raio  $r$  do filtro notch. Devido aos problemas de quantização de  $k_2$  do filtro,  $r$  não pode assumir valores muito próximos de 1. Valores muito pequenos para  $r$  tornam a convergência lenta. Assim sendo, os valores de  $r$  que melhor se adaptaram às características do sistema foram determinados, por simulação, no intervalo de 0,8 a 0,85 [4].

Como o algoritmo de adaptação é baseado no valor do gradiente na saída do filtro notch, o valor do erro entre a

freqüência do sinal recebido e a freqüência do notch influí na escolha do valor de  $\mu$ . Quando o valor deste erro for pequeno,  $\mu$  elevados podem criar fortes oscilações em torno da freqüência alvo. Por outro lado, quando este erro for grande, coeficientes de valor muito baixo tornam lenta a convergência. Esses fatores, comuns a técnicas gradientes, criaram a necessidade de um algoritmo de adaptação de  $\mu$ , em que é fixado o valor 0,305 para as cinquenta iterações iniciais, quando se observa significativa estabilização da saída notch. A partir daí  $\mu$  muda para 0,03, valor obtido em laboratório, que garante a convergência (trajetória de  $k_I$  penetrando a faixa final de incerteza no entorno do seu valor real) em menos de 100 iterações [4] com RSR = 20 dB.

O critério de convergência utilizado foi que a trajetória de  $k_I$  ficasse dentro de um espalhamento correspondente a um afastamento máximo de 10 Hz da freqüência verdadeira. É desta faixa final de incerteza ao redor do  $k_I$  ótimo que se projetou o número máximo de quadros com informação, 8, antes de se repetir um quadro de sincronização. Com este erro, no máximo perde-se a última amostra correspondente ao último quadro de informação. De qualquer forma, o valor médio de  $k_I$  é o realmente utilizado, reduzindo substancialmente este espalhamento através do filtro  $k_{Im}=(k_{Im}+k_I)/2$ , executado a partir da amostra 50 e até a amostra 750 do quadro da portadora.

Devido à dificuldade de implementação de operações de divisão no DSP utilizado, a normalização de energia descrita na Equação (1) não foi implementada na forma indicada, sendo substituída por um controle automático de ganho à entrada do DSP, regulando a energia da portadora.

## Demodulação

O primeiro passo no processo de demodulação é monitorar a freqüência da portadora, determinando-se o parâmetro  $k_I$  do filtro notch. Após a convergência,  $k_I$  tem um espalhamento muito reduzido em torno da freqüência da portadora, e o início de um quadro de dados é detectado por um aumento deste espalhamento. Este método baseado num gradiente positivo da saída do filtro notch tende a eliminar os erros de sincronismo de símbolo devido à presença de fontes impulsivas de ruído. O método apresentou bom desempenho com RSR de até 10 dB, o que foi comprovado em testes de laboratório, como indicado na seqüência do trabalho..

Após a estimação da freqüência da portadora, ou seja, após a determinação de  $k_{Im}$  (médio) executa-se, durante a recepção dos quadros de dados, um procedimento para fazer com que a freqüência de amostragem do receptor seja igual a cinco vezes a freqüência da portadora. Para implementar este procedimento pensou-se inicialmente em fazer um ajuste direto da freqüência de amostragem do receptor. Isto não foi possível devido às limitações impostas pelo hardware da placa DSP, como será visto a seguir.

Tem-se que o erro de fase  $ef(n)$ , após  $n$  amostras no receptor em função da freqüência de amostragem no receptor  $f_a$ , e da freqüência de amostragem ideal do receptor  $f_{ao}$  (igual a 5 vezes a freqüência portadora, ou seja 56.003,6Hz) é dado por:

$$ef(n) = \frac{2\pi \cdot n}{5} \cdot \left( \frac{f_{ao}}{f_a} - 1 \right)$$

Estipulou-se (valor comprovado posteriormente em laboratório) que o erro de fase após cada quadro (800 amostras) deveria ser menor do que  $\pi/5$ . Conseqüentemente, a diferença em freqüência  $\Delta f_a = |f_a - f_{ao}|$ , deveria ser menor que 35Hz. No receptor o instante de amostragem é determinado pela interrupção do temporizador interno ao TMS320C25, e esta ocorre com a taxa  $f_i$  dada por:

$$f_i = \frac{12,5 \cdot 10^6}{PRD + 1}$$

onde  $12,5 \times 10^6$  é a freqüência do relógio mestre (50MHz) dividida por 4 e PRD é um registro do TMS320C25 que determina o período de interrupção [7][8]. Pode-se constatar verificando a Tabela 1 que a variação mínima em torno da freqüência central é de aproximadamente 250 Hz, valor muito superior ao desejado, o que impossibilita o ajuste direto da freqüência de amostragem do receptor.

**Tabela 1 - Relação entre o valor do registro PRD e a freqüência de amostragem na placa DSP.**

PRD	Freq. de Amostragem (Hz).
221	56.306,3
222	56.053,8
223	55.803,6

Inicialmente, a solução adotada para este problema baseava-se em uma freqüência de amostragem fixa, no caso 56.053,8Hz, e na correção do erro de posicionamento das amostras através de um processo de interpolação das amostras recebidas, o que foi implementado através da utilização de uma tabela que tinha como entrada uma indicação do valor da freqüência da portadora medida  $k_{Im}$  como saída o índice das amostras que participariam da interpolação e o fator de interpolação. Os resultados obtidos por este método não foram satisfatórios para uma interpolação linear. A interpolação de senóides por mínimos quadrados deu bons resultados, à custa de sobrecarga computacional.

Assim sendo, optou-se por uma outra solução de implementação mais simples. Ela consiste na utilização de uma freqüência de amostragem não uniforme ao longo de cada quadro de dados, ou seja, em alterar finamente a freqüência de amostragem do receptor durante a aquisição de dados, de forma que a freqüência de amostragem média satisfaça os requisitos de resolução em freqüência.

Esta solução, pouco ortodoxa, é implementada da seguinte forma: de posse de  $k_{Im}$ , utiliza-se este valor para selecionar de uma tabela uma fila circular contendo  $N$  valores para  $PRD$ . A cada nova interrupção do temporizador, armazena-se em  $PRD$  o valor seguinte da fila circular e, conseqüentemente altera-se o período de interrupção.

A frequência média de interrupção para uma fila contendo  $N$  valores de  $PRD$  é

$$\bar{f}_i = \frac{N \cdot 12,5 \cdot 10^6}{N + \sum_{k=1}^N PRD_i(k)}$$

Usando-se uma tabela contendo várias filas circulares cada uma com 16 valores do divisor  $PRD$  chega-se aos valores de frequência média, dados pela Tabela 2.

Da Tabela 2 pode-se ver que o  $\Delta f_a$  é de aproximadamente 15Hz, o que satisfaz o requisito de resolução em frequência. Pode-se verificar também que a variação máxima de fase ao longo da fila ocorre quando todos elementos da fila são idênticos com exceção de um único elemento, cujo valor difere de uma unidade em relação aos outros. Este erro é de aproximadamente 0,017 rad, no caso apresentado na Tabela 2 (n-2), e por ser muito baixo pode ser desprezado.

A tabela em questão foi montada de forma que a frequência de amostragem pudesse variar de 55633,6Hz a 56369,6Hz, ou seja, de 370Hz abaixo da frequência da amostragem ideal ( $5 \times f_{po} = 56003,6$  Hz) até 366Hz acima dela. Isto permite que a frequência da portadora possa variar de 11126,7Hz até 11273,9 Hz, ou equivalentemente 74Hz abaixo da portadora ideal ( $f_{po} = 11200,7$ Hz) até 73Hz acima dela, o que satisfaz a tolerância especificada de  $\pm 60$ Hz.

## Decimação como Demodulação

Como já havia sido mencionado, o receptor deveria decimar por cinco o sinal amostrado. No entanto, como a frequência de amostragem do receptor é 5 vezes a frequência de amostragem do transmissor, a decimação, na realidade, é por 25.

A partir da detecção de início de quadro, o receptor deve começar a executar as seguintes tarefas: decimar por 25, armazenar as amostras e, ao final de cada quadro, disparar a detecção (FFT + decisão). No entanto, pela dificuldade de se identificar, em presença de ruído aditivo no canal, a fase ótima (que maximiza o nível do sinal decodificado), ou seja, a amostra correspondente ao primeiro pico do sinal modulado, optou-se por armazenar alternadamente em cinco buffers as amostras de 5 seqüências consecutivas, decimadas por 25, conforme apresentado na Figura 5, e ao final do quadro, selecionar o buffer com maior energia, entregando-o para o estágio de detecção.

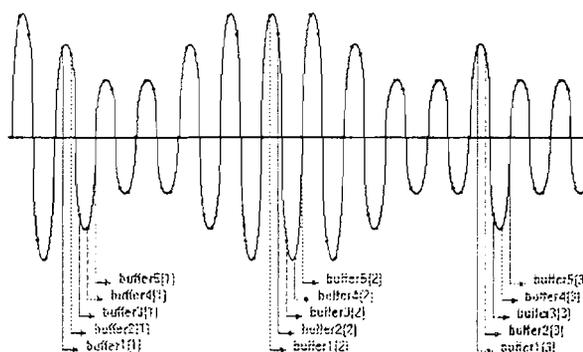


Figura 5 - Processo de preenchimento dos 5 buffers de dados.

O fato da frequência de amostragem do receptor ser definida como cinco vezes a frequência da portadora tem duas razões: durante o quadro da portadora esta taxa facilita a estimação da frequência da portadora; durante os quadros de dados, esta taxa também é conveniente pois facilita o procedimento para escolha da fase inicial visando maximizar a amplitude do sinal [1].

## Deteção

Como visto, o procedimento de decodificação dos símbolos baseia-se na FFT do sinal, de forma a identificar os tons do espectro em frequência. Após selecionar, dos 5 buffers, o que possui maior energia, executa-se uma FFT de 32 pontos. Feito isto, passa-se para o estágio de decisão que tem por objetivo estimar, a partir da observação do espectro do sinal recebido, quais as raia de frequência contidas no quadro transmitido. A Figura 6 mostra a magnitude do espectro obtido na saída do receptor, para um sinal modulado com ruído branco aditivo.

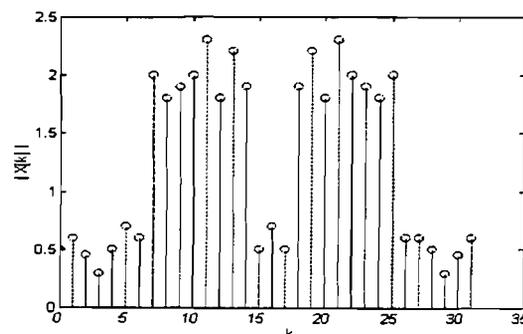


Figura 6 - Espectro de magnitude na saída do receptor, para a palavra (01)<sub>H</sub>.

Como pode ser visto na Figura 1, as raia na primeira metade do espectro foram alocadas de acordo com o código binário da palavra a ser transmitida, enquanto as raia na segunda metade do espectro aparecem como o complemento da primeira parte. Esta característica do sinal transmitido é utilizada para decidir a existência ou não de uma dada raia. Isto é feito comparando a amplitude do espectro da raia  $n$  ( $1 \leq n \leq 8$ ) à amplitude da raia  $n+8$ . Caso a amplitude da raia  $n$  seja maior que a

amplitude da raia  $n+8$  decide que o bit  $n$  é 1, caso contrário decide-se que o bit  $n$  é zero.

### 3. IMPLEMENTAÇÃO EM TEMPO REAL

O hardware utilizado para implementar o sistema consiste de duas placas de desenvolvimento da Atlanta Signal Processors Incorporated (ASPI) [8], baseadas no microprocessador TMS320C25, que opera em 50 MHz, uma memória rápida de 24K palavras, e conversores A/D e D/A de 16 bits, com frequência máxima de amostragem de 200.000 amostras/s. Há também dois filtros externos passa-faixas ativos de 4ª ordem, com banda passante de  $11,2 \pm 1,12$  kHz, sendo que um deles é usado na saída da placa do transmissor e o outro na entrada da placa do receptor. Todas as funções do transmissor e receptor, descritas anteriormente, foram implementadas em software, usando a linguagem Assembly do TMS320C25.

#### Software do Transmissor

A primeira tarefa que o TMS executa é a rotina de inicialização, que consiste na definição das portas de entrada e saída, inicialização de variáveis, carregamento da tabela com senos e cossenos que será utilizada no processo de interpolação. A seguir é gerado o quadro correspondente ao primeiro byte a ser transmitido. Uma vez gerado o primeiro quadro, ele é transferido para um buffer de saída (buffer 1), de onde as amostras irão para o D/A, por um processo de interrupção que ocorre a cada  $44,5 \mu s$  ( que é a frequência de amostragem na saída ). Enquanto o buffer 1 está sendo esvaziado, o próximo byte começa a ser processado e o resultado vai para o buffer 2. A transferência dos dados deste buffer para o buffer 1 ocorre após este último enviar a última amostra para o D/A. Esta transferência não leva mais do que  $40 \mu s$ . O tempo total para o cômputo de um quadro é de 1 ms, e como um quadro é transmitido em 14,9 ms, sobram ainda 13,9 ms para o processador realizar outras tarefas, como por exemplo executar rotinas de auto-testes.

#### Software do Receptor

As rotinas de inicialização seguem o mesmo procedimento do transmissor, exceto que agora há uma

tabela circular que controla a frequência de amostragem do receptor. A aquisição de dados é feita por interrupções a intervalos de  $17,86 \mu s$ . Durante a recepção do quadro de sincronização é computado o parâmetro  $k_{im}$  e um teste é realizado para se determinar o início do primeiro quadro de dados. Tão logo o primeiro quadro for detectado, a rotina de sincronização é desabilitada, e começam os processos de decimação e detecção descritos anteriormente. No caso do receptor o tempo de processamento é suficiente, embora muito crítico. Isto sugere a utilização de um outro processador para dividir as tarefas e eventualmente complementá-las, como por exemplo através de código detetor de erros, etc.

### 4. RESULTADOS

Foram realizadas diversas simulações para testar os desempenhos do transmissor e do receptor, incluindo os efeitos de quantização. Para isto, todas as operações aritméticas foram feitas em ponto fixo com palavras de 16 bits. Foram utilizados 26 arquivos compostos por uma portadora de tamanho variável, seguida de 9 quadros de informação, gerados por um algoritmo em linguagem C. A codificação dos sinais segue o mesmo padrão utilizado no TMS para a transmissão (inteiro de 16 bits).

Com uma RSR de 10 dB, o período de adaptação até se alcançar o sincronismo de portadora, para os 26 sinais, variou entre 42 e 98 amostras. Como um quadro fornece 800 amostras, esta taxa para a convergência garante o sincronismo de portadora num intervalo de tempo inferior ao da duração dos quadros.

Os experimentos em tempo real foram realizados no laboratório, com o transmissor e o receptor conectados por um cabo, como mostra a Figura 7.

#### Avaliação do Desempenho com Ruído Aditivo no Canal

Foram executados vários testes visando estudar o comportamento do receptor em operação com sinais corrompidos por ruído aditivo no canal. Durante os teste variou-se o nível do ruído branco fornecido pelo gerador externo de forma a alterar a RSR medida na entrada do receptor (antes do filtro "anti-alias"), e observou-se o desempenho do sistema. O resultado deste teste pode ser

Tabela 2 - Relação dos valores das filas circulares de PRD com a frequência de amostragem.

fila	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	freq (Hz)
n-2	222	222	222	222	222	222	222	222	222	222	222	222	222	222	222	223	56038,1
n-1	223	222	222	222	222	222	222	223	222	222	222	222	222	222	222	222	56022,4
n	223	222	222	222	222	223	222	222	222	222	223	222	222	222	222	222	56006,7
n+1	223	222	222	222	223	222	222	222	222	223	222	222	223	222	222	222	55991,4

**Tabela 3 - Desempenho do sistema na presença de ruído aditivo no canal.**

Simulação Nro.	RSR (dB)	Nro. de Erros de Sincronismo	Taxa de Acerto de Byte %	Taxa de Acerto de Bit %	Comentários
1	10	219	96.97	99.28	42% dos erros no 1 quadro. 8% nos quadros restantes
2	10	257	96.68	99.20	aproximadamente mesma ocorrência.
3	15	4	99.99	99.999	1 erro no primeiro quadro. 1 erro no 7 quadro
4	15	7	99.99	99.999	1 erro no primeiro quadro. 1 no 7 e 1 no 8
5	20	0	100	100	
6	20	0	100	100	

resumido pela Tabela 3. Em cada simulação foram transmitidos cerca de 30000 bytes, com codificações aleatórias.

### Desempenho do Receptor em Função de Desvios da Frequência da Portadora

Estes testes têm por finalidade verificar o comportamento do receptor em função dos desvios na frequência da portadora, e/ou na frequência do relógio do receptor. Para tal variou-se a frequência da portadora do sinal transmitido e observou-se o comportamento do receptor. O resultado deste teste é apresentado na Tabela 4.2, podendo-se observar ser bastante satisfatório.

## 5. . CONCLUSÕES

O trabalho focalizou o desenvolvimento de um sistema de telemetria acústica para a comunicação entre estações fixas em águas profundas, enfatizando os algoritmos utilizados para tornar esta comunicação mais confiável. Os resultados da simulação em tempo real no laboratório foram excelentes. O ponto crítico do projeto está no receptor, onde não há tempo suficiente para operações adicionais, forçando ao uso de técnicas pouco

convencionais como a da interpolação-decimação ao invés de modulação-demodulação por senóides e a sincronização fina de portadora por contagem média no quadro. Isto sugere a necessidade de um outro processador que possa tomar conta das operações de demodulação e detecção, deixando o atual processador com as tarefas de aquisição, sincronização e decimação. Com esta alternativa será possível processar as outras quatro seqüências de 32 amostras que resultam do processo de decimação em cada quadro de dados, o que acarretará maior segurança na recepção. Além disto pode-se utilizar uma codificação extra para detecção de erros e/ou recuperação de dados, não empregada durante os testes.

## AGRADECIMENTOS

Este trabalho foi financiado pela PETROBRÁS através do CENPES e pelo MCT.

## REFERÊNCIAS

- [1] J.A.Catipovic "Performance Limitations in Underwater Acoustic Telemetry", in IEEE Journal of Oceanic Engineering, vol15, No 3, July 1990.

**Tabela 4 - Desempenho do sistema com desvios na frequência da portadora.**

Simulação Nro.	Freq. Da portadora (Hz)	Delta f em torno de 11200.7Hz (Hz)	RSR (dB)	Nro. de Erros de Sincronismo	Taxa de Acerto de Byte %	Taxa de Acerto de Bit %
1	11140.8	59.9	(sem ruído)	0	100	100
2	11261.3	60.6	(sem ruído)	0	100	100
3	11140.8	59.9	20	0	100	100
4	11261.3	60.6	20	0	100	100

[2] J.A. Catipovic and L.E. Freitag, "Analysis of high

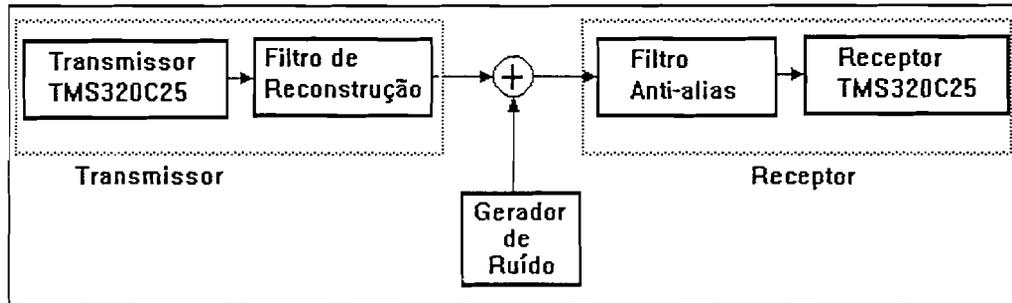


Figura 7 - Configuração utilizada nos testes.

frequency multitone transmission propagation in the marginal ice zone", in *J. Acoust. Soc. Am.* 86(1), July 1990.

- [3] L. E. Freitag and J. A. Catipovic, "A signal processing system for underwater acoustic ROV communication", in *Proc. 6th Int. Symp. Unmanned Untethered Submersible Tech.* (Baltimore, MD), June 1989, pp. 34-41.
- [4] P.L. Osorio, J. Szczupak, A. Yamaoka, P. Rosa and R. Alves, "Processamento Digital de Sinais para Comunicação Acústico-Submarina", Relatório Final, Convênio DEE-CENPES, 1993, 151 pp.
- [5] M. Petraglia, S. K. Mitra, and J. Szczupak, "Adaptive sinusoidal detection using notch filters and multirate techniques", *Proc. IEEE ISCAS*, May 1990, pp. 271-274.
- [6] P.L. Osorio, J. Szczupak, e M.V. Nóbrega, "Aprimoramento e Estimativa de Desempenho em Condições Ruidosas do Sistema de Processamento de Sinais para Comunicação Acústico-Submarina", Relatório Final, Convênio DEE- CENPES, 1994, 69 pp.
- [7] Texas Instruments - "Second Generation TMS320 User's Guide" - 1990
- [8] Atlanta Signal Processors, Incorporated - "ADP Chimera Algorithm Development Package for the Texas Instruments TMS32020 Family of Digital Signal Processors Microcomputers" - Version 1.33 - August, 1988

**Paulo L. Osorio** é formado em Engenharia Elétrica pela UFRGS, em 1967. Tem Mestrado em Engenharia Elétrica pela PUC-RIO, de 1970, Doutorado e Pós-Doutorado em Engenharia Elétrica pela University of Houston, de 1976 e 1985, respectivamente. É Professor do DEE da PUC-RIO desde 1968.

**Jacques Szczupak** é formado em Engenharia Elétrica pela UFRJ em 1964. Tem Mestrado em Engenharia Elétrica pela COPPE-UFRJ, de 1967 e Doutorado em Engenharia Elétrica pela University of California at Davis, de 1975. Trabalhou na COPPE de 1967 a 1977, no CEPTEL de 1978 a 1985, na COPPE de 1985 a 1987, e na PUC-RIO a partir de 1987.

**Marcos Vinicius da Silva Nóbrega** é formado em Engenharia Elétrica pela PUC-RIO em 1990. Tem Mestrado em Engenharia Elétrica pela PUC-RIO, de 1993. Trabalha atualmente na ELEBRA.

**Alexandre Yamaoka** é formado em Engenharia Elétrica pela UFRJ, em 1990. Ingressou no Mestrado da PUC-RIO, em 1991, tendo completado todos os créditos. Trabalha na Golden Cross.

**Paulo Rosa Pereira** é formado em Engenharia Elétrica pela UFRJ, em 1990. Tem Mestrado em Engenharia Elétrica pela PUC-RIO, de 1995. Trabalha atualmente na IBM.

**Rogério Guedes Alves** é formado em Engenharia Elétrica pela UFRJ, em 1990. Tem Mestrado em Engenharia Elétrica pela PUC-RIO, de 1993. Atualmente está concluindo o Doutorado na COPPE.