# IMPLEMENTAÇÃO DE UM ALGORITMO DE REDUÇÃO DE RUÍDO EM SINAIS DE VOZ USANDO O DSP TMS320C31

LUIZ A. P. MARÇAL e JOZUÉ VIEIRA FILHO

Universidade Estadual Paulista[1]

(DEE/FEIS/UNESP)

Av. Brasil Centro, 56, Ilha Solteira/SP

## RESUMO

O melhoramento da qualidade de sinais de voz através de técnicas de redução de ruído tem sido utilizado em várias aplicações em telecomunicações, como em sistemas de reconhecimento automático de fala, codificação, telefones tipos viva-voz ("hands free telephone"), etc. Todas as aplicações exigem processamento em tempo real e, normalmente, utilizam processadores de sinais digitais (DSP). Neste trabalho, estudou-se e implementou-se em tempo real um algoritmo de redução de ruído. O sistema é baseado no DSP TMS320C31, operando em 60 MHz, e usa uma taxa de amostragem de 8kHz. O algoritmo de redução de ruído é baseado na subtração espectral e foi implementado usando a "Short Time Fourier Transform" (STFT). Os resultados mostram que o algoritmo ocupa pouco tempo de processamento e melhora significativamente a qualidade do sinal, o que permite sua aplicação em sistema práticos.

## 1. INTRODUÇÃO

O desenvolvimento e a implementação, para operação em tempo real, de novos algoritmos de redução de ruído em sinais de voz têm recebido atenção especial com o crescente desenvolvimento das telecomunicações. Nesse contexto, a redução de ruído pode ser aplicada em várias atividades, como por exemplo, em redes telefônicas convencionais, telefonia celular, codificação, telefones viva-voz ("Hands-free") e até em serviços automáticos de informação baseados em sistemas com síntese/reconhecimento de voz. Assim, dominar essa técnica, tanto do ponto de vista teórico como também de implementação prática, passa a ser de grande importância.

Na maioria das aplicações, um algoritmo de redução de ruído eficiente deverá ter duas características fundamentais:

- bom desempenho no melhoramento do sinal nos aspectos físico (recuperação da forma de onda original) e auditivo (inteligibilidade);
- baixa carga computacional, visando as aplicações em tempo real.

Uma primeira etapa do trabalho é definir o algoritmo a ser implementado. Estudos anteriores [1] mostraram que a subtração espectral baseada na relação sinal/ruído é um método que possibilita boa redução de ruído, boa qualidade no sinal processado e apresenta baixa carga computacional, sendo assim uma técnica adequada para uso em tempo real. Usar uma técnica baseada na subtração espectral significa também a necessidade de um algoritmo para detecção de intervalos de silêncio, que tem como objetivo viabilizar a estimação da potência do ruído.

Um segundo ponto é a plataforma de desenvolvimento. Após um estudo prévio sobre custos e facilidades de manipulação de DSPs, optou-se pela placa TIGER 31/PC [2], que possui um DSP de ponto flutuante, o TMS320C31, operando com 60 MHz de "clock". O ambiente para o desenvolvimento do software foi o Hypersignal RIDE [3], que é executado numa plataforma Windows98/NT. A grande vantagem deste software é a facilidade de programação, que é baseada em diagramas de blocos. Também, o RIDE possui controladores que realizam a interface entre o PC e a TIGER 31/PC. Apesar de usar uma linguagem gráfica, a implementação do filtro exigiu um estudo detalhado sobre o TMS320C31, a plataforma completa (hardware/software) e um estudo profundo do algoritmo utilizado. Além disto, dominar as técnicas de processamento digital de sinais baseadas na "Short Time Fourier Transform" é fundamental para aplicações envolvendo sinais de voz.

Neste trabalho, apresenta-se inicialmente o algoritmo de redução de ruído utilizado, detalhando-se a metodologia de implementação com base na STFT. Na seqüência apresentam-se as etapas implementadas na plataforma utilizada e, finalmente, discutem-se os resultados finais obtidos.

## 2. ALGORITMO DE REDUÇÃO DE RUÍDO

Seja um sinal de voz puro, $v(t)$, degradado por um ruído aditivo $r(t)$, formando um sinal ruidoso $y(t)$, dado por:

$$y(t) = v(t) + r(t) \tag{1}$$

Considerando uma aplicação com base na STFT, o que significa um trecho de sinal de voz limitado e praticamente estacionáio, tem-se:

$$y_{\Delta t}(t) = v_{\Delta t}(t) + r_{\Delta t}(t) \tag{2}$$

Após um processo de amostragem do sinal ruidoso e considerando apenas as amostras no intervalo $\Delta t$, pode-se reescrever a equação 2 no domínio da freqüência como segue:

[1] *Trabalho desenvolvido com o apoio financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo 98/04257-4, e da CAPES (bolsa de mestrado).*

$$Y_{\Delta t}(\omega) = V_{\Delta t}(\omega) + R_{\Delta t}(\omega) \quad (3)$$

Considerando que o ruído é estacionário e que sua potência pode ser estimada sem problemas, define-se o seguinte filtro redutor de ruído:

$$|H_{\Delta t}(\omega)| = \frac{\sqrt{|Y_{\Delta t}(\omega)|^2 - |R_{\Delta t}(\omega)|^2}}{|Y_{\Delta t}(\omega)|} \quad (4)$$

O filtro apresentado na equação 4 é conhecido como *subtração espectral clássica* (SE) [4] e deixa no sinal processado um inconveniente ruído residual denominado de "ruído musical". Nos estudos realizados por [1] foi proposto um novo filtro, seguindo a definição básica da equação 4, que é dado por:

$$|H_{\Delta t}(\omega)| = \sqrt{\frac{E\{SNR\_prio_{\Delta t}(\omega)\}}{E\{SNR\_prio_{\Delta t}(\omega)\}+1}} \quad (5)$$

onde $$SNR\_prio_{\Delta t}(\omega) = \frac{|V_{\Delta t}(\omega)|^2}{|R_{\Delta t}(\omega)|^2} \quad (6)$$

é a relação sinal/ruído a priori e a expressão $E\{\bullet\}$ indica um valor estimado.

O filtro definido na equação 5 não apresenta o ruído musical e permite uma excelente redução de ruído. A carga computacional é média e sua implementação em tempo real foi conseguida com sucesso.

## 3. METODOLOGIA DE IMPLEMENTAÇÃO

Considerando o uso da STFT e tendo que a fase do sinal ruidoso pode ser usada na reconstrução do sinal processado [5], a implementação do filtro proposto da equação 6, denominado aqui de Subtração Espectral Modificada (SEM), é realizada de acordo com o diagrama da figura 1.

O janelamento é aplicado ao sinal amostrado, *y(n),* e tem a função de tomar trechos de voz praticamente estacionários, permitindo o uso da STFT. Associado à janela, deve-se definir também o intervalo de sobreposição das janelas consecutivas, que depende do tipo de janela adotada. Uma valor típico é 50% para uso com a janela de Hanning, com duração que não deve exceder 40 ms [5]. Após o janelamento, obtém-se, via STFT, os espectros do sinal de voz ruidoso, do ruído e do sinal de voz estimado, indispensáveis na obtenção do filtro. A definição do número de pontos para os espectros da STFT é um fator relevante, pois existe uma relação entre resolução espectral e tempo de cálculo da STFT, isto é, com mais pontos têm-se espectros mais bem definidos, porém o processamento será mais lento e vice-versa. No projeto proposto é usada uma STFT de 256 pontos, para uma taxa de amostragem de 8 kHz.

Para a realização da filtragem proposta, que é função da Relação Sinal/Ruído a priori (SNR_prio), deve-se estimar a potência do ruído aditivo. Assim, com a indicação obtida no bloco de detecção de intervalos de silêncio (DIS), faz-se uma estimação média do ruído através de uma filtragem recursiva, dada por:

$$E\{|R_{\Delta t}(\omega)|^2\}_k = \alpha.E\{|R_{\Delta t}(\omega)|^2\}_{k-1} + (1-\alpha).\{|Y_{\Delta t}(\omega)|^2\}_k \quad (7)$$

onde $k$ indica o intervalo de análise atual. A filtragem elimina a possibilidade de variações abruptas na potência do ruído de uma janela para outra. Ressalta-se que ruídos impulsivos devem ser identificados como trechos de voz. O fator $\alpha$ determina o grau de dependência entre as janelas passadas e seu valor é, normalmente, maior do que 0,8 [1]. Para garantir a inicialização do sistema até o momento em que a DIS começa a atuar, supõe-se que não há sinal de voz nos primeiros 100 ms, o que é viável para a maioria das aplicações envolvendo telefonia (tempo em que um usuário leva para começar a falar a partir do momento em que a comunicação está estabelecida).

A estimação da relação sinal/ruído a posteriori (SNR_post) pode ser obtida diretamente do espectro de amplitude do sinal ruidoso captado e da potência do ruído estimada. Assim, considerando que $|Y_{\Delta t}(\omega)|^2$ representa o espectro de potência do sinal ruidoso para a janela analisada, tem-se:

$$E\{SNR\_post_{\Delta t}(\omega)\} = \frac{|Y_{\Delta t}(\omega)|^2}{E\{|R_{\Delta t}(\omega)|^2\}} \quad (8)$$

Para a estimação da Relação Sinal/Ruído a priori (SNR_prio), parâmetro principal do filtro, supõe-se que o sinal de voz não apresenta variações significativas de potência de uma janela para outra. Assim, utiliza-se um estimador recursivo baseado no valor estimado da SNR_prio da janela anterior e no valor da SNR_post da janela atual [6], que é dado por:

$$E\{SNR\_prio_{\Delta t}(\omega)\}_k = \beta.\frac{E\{|V_{\Delta t}(\omega)|^2\}_{k-1}}{E\{|R_{\Delta t}(\omega)|^2\}_k} + (1-\beta).T[E\{SNR\_post_{\Delta t}(\omega)\}_k - 1] \quad (9)$$

onde $k$ é a janela atual de análise, $T[\bullet]$ indica uma transformação sobre $E\{SNR\_post_{\Delta t}(\omega) - 1\}$ e $\beta$ representa a contribuição da *SNR_post.* Deve-se destacar que, de acordo com as equações 3, 6 e 8, $SNR\_post_{\Delta t}(\omega) = SNR\_prio_{\Delta t}(\omega) + 1$. A transformação $T[\bullet]$ se faz necessária porque o valor estimado de *SNR_post* pode ser menor do que 1. Neste trabalho, tem-se que:

$$Se \quad (E\{SNR\_post_{\Delta t}(\omega)\} - 1) \le \varepsilon \qquad então \quad (E\{SNR\_post_{\Delta t}(\omega)\} - 1) = \varepsilon \quad (10)$$

O valor de $\beta$ é importante pois, além de minimizar o conhecido ruído musical da SE, também reduz as distorções no sinal de voz processado. Se $\beta$ for próximo de 0 (zero), haverá menos distorções, mas o sinal apresentará muito ruído musical. Se, ao contrário, $\beta$ *for* muito próximo de 1,0 (hum), haverá mais distorções e, praticamente, não haverá ruído musical.

Finalmente, a cada janela processada, a filtragem é realizada aplicando-se a função de transferência da SEM ao espectro de amplitude do sinal ruidoso da janela em análise. Na seqüência, aplica-se a STFT inversa, recompondo o sinal no tempo. Para garantir uma reconstrução correta, é usado o método OLA (overlap-addition).

## 4. PLATAFORMA DE DESENVOLVIMENTO

A escolha de uma plataforma de hardware e software para o desenvolvimento e execução de uma aplicação de processamento de sinais digitais em tempo real é uma tarefa que requer uma avaliação do tipo de aplicação, das características do processador de sinais (DSP) que deverá executar a aplicação e de uma estimativa da quantidade de memória requerida pela aplicação. Neste projeto, o hardware é composto de uma placa de processamento de sinais TIGER 31/PC [2], acoplada a um

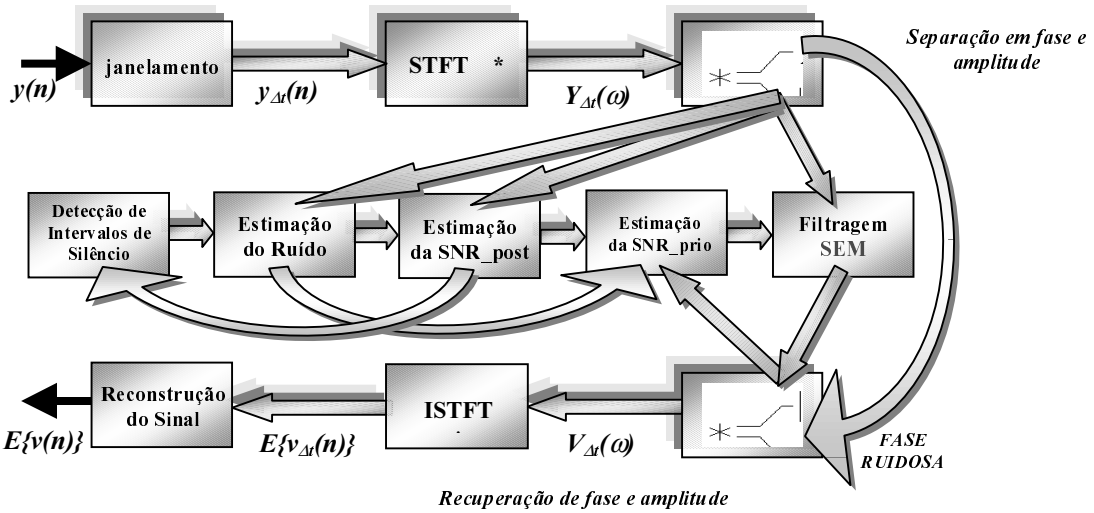


*Figura 1 – Diagrama em blocos para implementação do filtro usando a STFT*

microcomputador tipo PC, que possui 128 Kbytes de SRAM (memória RAM estática) e incorpora um processador de sinais digitais TMS320C31 operando a 60 MHz. O software escolhido para o desenvolvimento e implementação do algoritmo é o Hypersignal RIDE [3], que é executado na plataforma Windows e possui grandes facilidades de interfaceamento entre o PC e o DSP.

## 4.1 - A placa TIGER 31/PC

A TIGER 3 1/PC é uma placa para conexão em sistemas com barramento ISA ou para operação autônoma, numa arquitetura que permite utilizar todas as capacidades do processador TMS320C31. Além disso, possui uma arquitetura de barramentos que permite o interfaceamento com um PC e outras unidades de E/S (entrada/saída). Os principais componentes da TIGER 31/PC são o DSP, as unidades de memória SRAM, uma unidade de memória EPROM, os conversores A/D-D/A, uma inteface para linha telefônica analógica e uma porta serial RS-232.

## 4.2 - O software Hypersignal RIDE

O Hypersignal RIDE é um programa capaz de simular algoritmos em um ambiente gráfico usando uma biblioteca de blocos funcionais (DLLs). Sua arquitetura permite a implementação rápida de algoritmos com base em estruturas conceituais e é composta de dois níveis de atuação: um é a interface do usuário e o outro é a biblioteca de blocos funcionais. O software é baseado em "Dynamic Link Libraries" (DLLs), que podem ser consideradas como subrotinas ou blocos individuais e devem ser conectados entre si para formar a aplicação. Uma biblioteca padrão de blocos de simulação no PC e uma biblioteca padrão de blocos para execução no DSP estão disponíveis. Adicionalmente, novas funções podem ser desenvolvidas e incorporadas usando C ANSI e o aplicativo *Block Wizard* (parte integrante do RIDE). Também, todos os pontos críticos de uma implementação em tempo real, como configuração de memória, de interrupções, de entrada/saída, etc., podem ser manipulados. A interação da plataforma RIDE com uma placa de processamento de sinais é feita por um controlador específico, denominado de "controlador de tempo real". Esse controlador realiza toda a interface entre o PC e a placa com o DSP, fornecendo informações sobre fluxo de dados, viabilizando a conexão dos arquivos objetos, carregando código, dados e parâmetros para a memória do DSP, e monitorando todas as operações no DSP.

# 5. IMPLEMENTAÇÃO EM TEMPO REAL

O sistema de redução de ruído em sinais de voz implementado é baseado na *Subtração Espectral Modificada (SEM)*, ilustrada na figura 1, e em um algoritmo de *Detecção de Intervalos de Silêncio*, desenvolvido especialmente para a aplicação, mas que não é objeto de discussão neste trabalho. Para dar maior flexibilidade ao projeto, o diagrama apresentado na figura 1 foi dividido em subsistemas, que no RIDE representa um bloco hierárquico com entrada(s) e saída(s), que contém internamente blocos funcionais interligados. Os subsistemas desenvolvidos para uso no filtro redutor de ruído, incluindo os tamanhos dos vetores de dados utilizados, são:

- Transformação Tempo-Freqüência (128/129/256/512 dados)
- DIS e Estimação do Ruído (129 dados)
- Cálculo da Função de Transferência do Filtro (|H(w)|) (129 dados)
- Filtragem (129/256/512 dados)
- Transformação Freqüência - Tempo (128/256/512 dados)

## 5.1 - Transformação Tempo - Freqüência

O subsistema de transformação do domínio do tempo para o da freqüência é apresentado na figura 2.

Inicialmente, aplica-se aos quadros do sinal temporal original de entrada $y_{\Delta t}(n)$, obtido na conversão A/D da entrada "line in" da placa, um efeito de "overlap" de 50% e um janelamento, utilizando-se a janela de Hamming. O quadro de entrada, obtido a uma taxa de amostragem de 8 kHz, possui 128 amostras (16 ms). Na seqüência, aplica-se a FFT (otimizada para o TMS320C31) com 256 pontos. A saída da FFT é um sinal composto da parte Real (Re) e Imaginária (Im), utilizado posteriormente pelo subsistema de filtragem. A partir da FFT, calcula-se também o espectro de magnitude (otimizado para o TMS320C31) e o espectro de potência para o quadro analisado ($|Y^2_{\Delta t}(\omega)|$). A potência do sinal ruidoso é usada como entrada no subsistema "DIS e Estimação do Ruído.

## 5.2 - DIS e Estimação do Ruído

O subsistema de DIS e Estimação do Ruído é mostrado na figura 3, onde os blocos contidos no retângulo tracejado realizam a detecção dos intervalos de silêncio, baseada na variância da relação sinal/ruído ***(DIS)***. A saída do bloco denominado "Multiplex 1" corresponde ao sinal que controla a atualização do ruído. Este sinal é identificado como a saída da *DIS* e será igual a 1 (um), nos intervalos de silêncio detectados pela *DIS*, ou 0 (zero), caso contrário.

A estimação da potência do ruído é realizada a partir do espectro de potência do sinal ruidoso obtido nos intervalos de silêncio. Para realizar esta estimação, o subsistema da figura 3 foi implementado como segue:

- se o valor fornecido pela DIS for igual a 1 (um), isto significa a detecção de um intervalo de silêncio no trecho do sinal ruidoso em processamento;
- se o valor fornecido pela DIS for igual a 0 (zero), isto significa presença de voz no trecho em processamento.

Os cinco atrasos de quadro indicados na figura 3 dão robustez ao detetor de intervalos de silêncio. A atualização do ruído é realizada no bloco "Média Estimada do Ruído" que, juntamente com os blocos "Multiplex 2" e "Multiplex 3", também mantém, durante um trecho de voz, a média calculada no último intervalo de silêncio. Os blocos denominados "Multiplex 2" e "Multiplex 3" funcionam como chaves seletoras, fazendo com que o ruído seja atualizado somente se a saída da DIS for igual a 1 (um), ou seja, eles fornecem o espectro de potência do sinal ruidoso ao bloco "Média Estimada do Ruído" somente nos intervalos de silêncio, onde apenas o ruído aditivo estará presente. Caso contrário, eles manterão na saída a última média calculada. Desta maneira, no processamento de trechos de voz, o filtro manterá na sua saída a atualização feita no processamento da última janela de intervalo de silêncio detectado. Uma nova atualização do ruído só será realizada quando o próximo intervalo de silêncio for detectado, ativando-se os blocos "Multiplex 2" e "Multiplex 3".

O bloco "Armazenamento" foi configurado para armazenar as quatro últimas médias calculadas para a SNR_post, de onde obtém-se a variância. Posteriormente, um sistema recursivo representado pelo bloco "Variância Média" calcula uma média das variâncias, considerando as variâncias calculadas anteriormente e a variância atual.

Na inicialização do sistema, supõe-se que as cinco primeiras janelas de sinal são apenas ruído. A partir desse ponto a saída do bloco "Comparação" é transferida para a saída do bloco "Multiplex 1", isto é, a DIS passa a ser automática.

## 5.3 - Função de Transferência do Filtro (|H(ω)|)

A função de transferência do filtro usando a SEM depende somente do valor estimado da relação sinal/ruído a priori (SNR_prio). O subsistema de cálculo da função de transferência deste filtro é mostrado na figura 4, onde os blocos contidos no retângulo tracejado realizam a *Estimação da SNR_prio*.

O uso da relação sinal/ruído a priori (SNR_prio) é o que diferencia este filtro de outros baseados somente na SE Clássica. Dessa forma, a correta estimação desse parâmetro é fundamental para o bom desempenho do método.

A estimação da SNR_prio utiliza, como parâmetros de entrada, o valor estimado da potência da voz filtrada da janela anterior, a potência estimada do ruído e a relação sinal ruído a posteriori estimada (E{SNR_post}) da janela atual. Visando otimizar o tempo de processamento, verificou-se que uma modificação na expressão dada pela equação 6, para a obtenção da SNR_prio, significaria um melhor desempenho do sistema. A alteração proposta consiste em rescrever a equação 6 como função apenas da SNR_post, que é o parâmetro de entrada deste subsistema. Assim, da equação 6 tem-se:

$$\frac{E\{|V_{\Delta t}(\omega)|^2\}_{k-1}}{E\{|R_{\Delta t}(\omega)|^2\}_k} = \frac{\{|H_{\Delta t}(w)|^2 \times |Y_{\Delta t}(\omega)|^2\}_{k-1}}{E\{|R_{\Delta t}(\omega)|^2\}_k} = \{|H_{\Delta t}(\omega)|^2 \times SNR\_post_{\Delta t}(\omega)\}_{k-1} \quad (11)$$

Substituindo a equação 11 na equação 9, obtém-se:

$$E\{SNR\_prio_{\Delta t}(\omega)\}_k = \beta.\{|H_{\Delta t}(\omega)|^2 . SNR\_post_{\Delta t}(\omega)\}_{k-1} + (1-\beta).\Re[E\{SNR\_post_{\Delta t}(\omega)\}_k - 1] \quad (12)$$

onde *k* representa o índice de interação, incrementado a cada janela analisada.

Os blocos limitadores do diagrama da figura 4 têm a função de limitar valores, podendo ser definido tanto um limite inferior como um limite superior para os valores de saída do bloco. No subsistema de cálculo da função de transferência do filtro, a parcela da SNR_prio, calculada a partir da SNR_post, foi limitada a um valor mínimo de ε = 0,1, tendo como objetivo minimizar o ruído musical. Além disso, o valor estimado da SNR_prio foi limitado a valores entre 0,5 e 100, evitando desta maneira uma forte atenuação do sinal, o que diminui as distorções na voz processada.

## 5.4 – Filtragem e Transformação Freqüência Tempo

Para melhorar o desempenho do sistema, foram utilizados blocos de FFT/IFFT otimizados, que operam com sinais compostos, constituídos de parte real (Re) e imaginária (Im). Entretanto, essa separação pode aumentar a complexidade do sistema, pois uma aplicação direta do filtro via magnitude exige uma conversão retangular→polar e vice-versa. Porém, mostra-se que estes cálculos são equivalentes a aplicar a função de transferência às partes real e imaginária do sinal composto proveniente da FFT, o que torna direta a transformação do domínio da freqüência para o do tempo via IFFT otimizada. Assim, o subsistema que realiza a filtragem foi elaborado como mostra a figura 5.

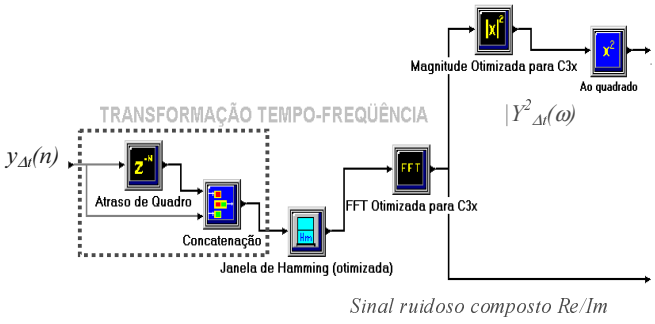


*Figura 2– Subsistema Transformação Tempo - Freqüência*

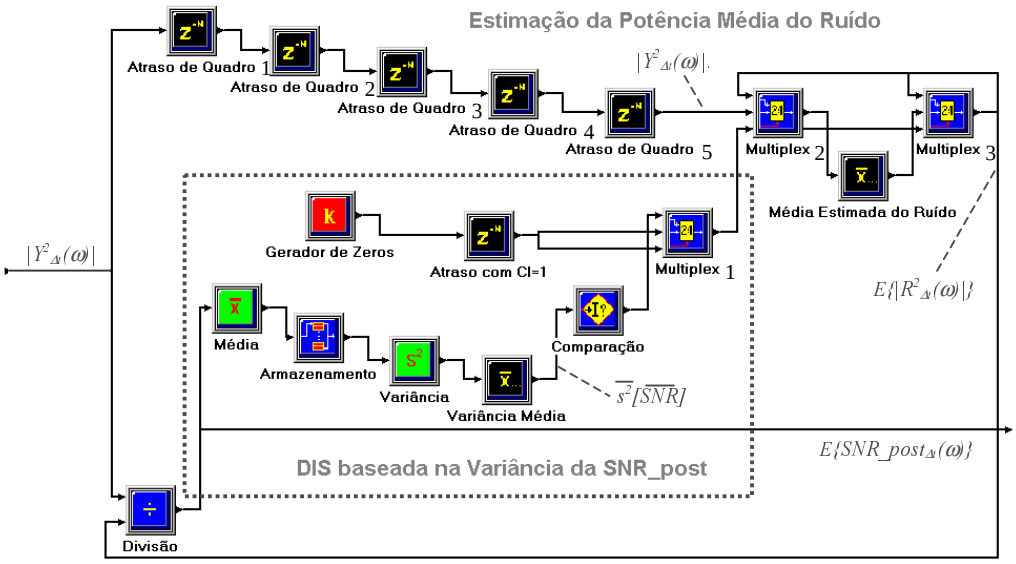


*Figura 3– Subsistema de DIS e Estimação da Potência do Ruído*

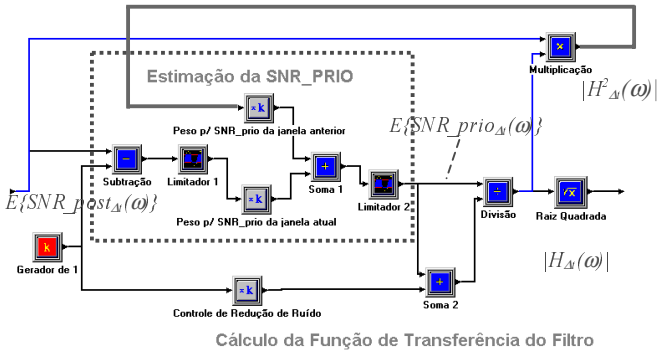


*Figura 4 –Subsistema Função de Transferência*

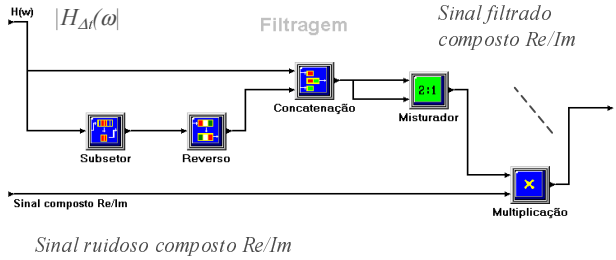


*Figura 5 – Subsistema Filtragem*

A preparação da função de transferência, realizada pelos blocos "Subsetor", "Reverso", "Concatenação" e "Misturador", na figura 5, é baseada na maneira como o sinal proveniente do bloco de FFT é composto (intercalando as partes real e imaginária, amostra a amostra) e na simetria das partes real e imaginária de um espectro complexo da FFT. Após o ajuste da função de transferência para um vetor com 512 amostras, este é multiplicado pelo sinal ruidoso composto, proveniente da FFT otimizada (subsistema Transformação Tempo-Freqüência).O resultado da multiplicação é um sinal filtrado composto, usado pelo bloco de IFFT no subsistema "Transformação Freqüência-Tempo", como ilustrado na figura 6.

Os conversores A/D e D/A operam de modo síncrono e a taxa de amostragem pode ser alterada pelo usuário no bloco de conversão D/A, podendo variar de 7350 Hz a 44 kHz. Porém, a quantidade e complexidade dos blocos envolvidos na aplicação determinou uma freqüência de amostragem máxima permitida para o processamento do filtro em tempo real. Também, observa-se que a quantidade de memória de dados requerida na placa TIGER 31/PC é diretamente proporcional ao tamanho de quadro escolhido.

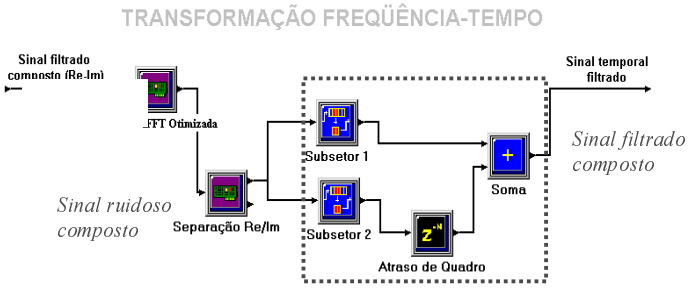


*Figura 6 – Subsistema Transformação Freqüência - Tempo*

## 6. RESULTADOS E AVALIAÇÕES

Para avaliar o sistema foi utilizado um sinal de voz puro, ao qual foi somado um ruído de carro. Os parâmetros de atualização do ruído e da SNR_prio foram $\alpha$=0,84 (equação 7) e $\beta$=0,99 (equação 9).

## 6.1 – Desempenho do TMS320C31

O quadro da tabela I apresenta um resumo do consumo de tempo na implementação do algoritmo. Os resultados mostram que, com uma taxa de 8 kHz de amostragem , o TMS320C31 ocupa apenas 36% do tempo disponível para processamento (considera-se o atraso de um quadro de 128 amostras). Assim, pode-se utilizar uma taxa de amostragem de até 17,4 kHz, tendo um atraso de apenas 6ms. Em termos de memória, foi utilizada uma quantidade de 87 Kbytes, o que corresponde a quase 80% da memória disponível (dados e programa) na placa. Os resultados são compatíveis com os obtidos em [7], que implementou a subtração espectral clássica, com uma carga computacional bem inferior ao método implementado neste trabalho.

## 6.2 – Desempenho do algoritmo

Na figura 7 são apresentadas as formas de onda dos sinais puro, ruidoso e processado. Na figura 8 é apresentado o ganho na relação sina/ruído. Nota-se uma redução média de 12 dB, o que garante uma boa qualidade no sinal processado. Isto foi confirmado com um teste subjetivo informal. Nesse teste, 10 (dez) pessoas ouviram duas vezes cada sinal. Ao final, foram unânimes em afirmar que o sinal processado era bem superior ao sinal ruidoso.

| Subsistema | Tempo de Processamento (em ms) | Percentual do tempo total |
|---|---|---|
| Conversão A/D e D/A | 0,10 | 0,7% |
| Transformação Tempo-Freqüência | 1,72 | 10,7 % |
| DIS e Estimação do Ruído | 0,84 | 5,3% |
| Cálculo de $\|H(\omega)\|$ | 1,20 | 7,5% |
| Filtragem | 0,45 | 2,8% |
| Transformação Freqüência-Tempo | 1,55 | 9,7% |
| **Tempo disponível** | 10,14 | 63,9% |
| TOTAL: | **16,00** | **100,0%** |

Tabela I – Desempenho do TMS320C31-60MHz

## 7. CONCLUSÕES

Neste trabalho foi apresentado um sistema de redução de ruído em sinais de voz para operação em tempo real. O algoritmo usado é baseado na subtração espectral, mas apresenta resultados bem superiores aos obtidos com a subtração espectral clássica. O sistema foi implementado num DSP TMS320C31, operando com um "clock" de 60 MHz. A taxa de amostragem usada foi de 8 kHz, o que resultou num consumo de tempo de 40 %, considerando um quadro de 128 amostras. O nível de redução de ruído é muito bom e a qualidade subjetiva do sinal processado, avaliada informalmente, é superior à do sinal ruidoso original. O sistema apresentado será usado em testes de melhoraria da qualidade de sinais de voz em linhas telefônicas analógicas.

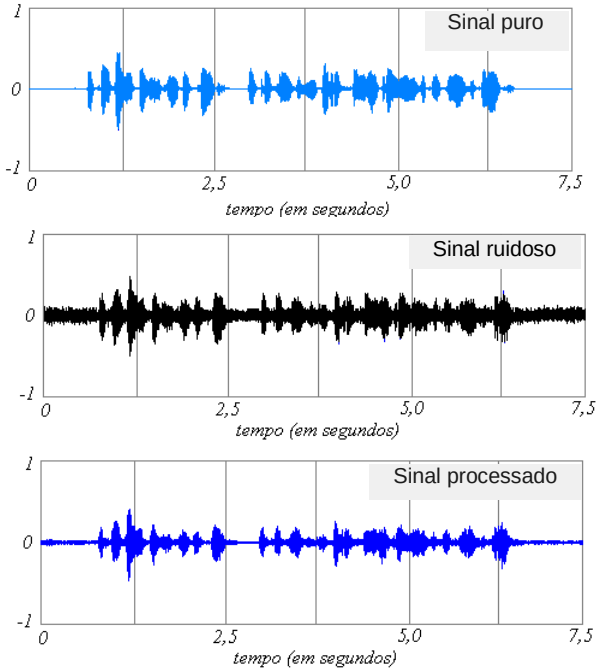


*Figura 7 – Formas de onda dos sinais de voz*

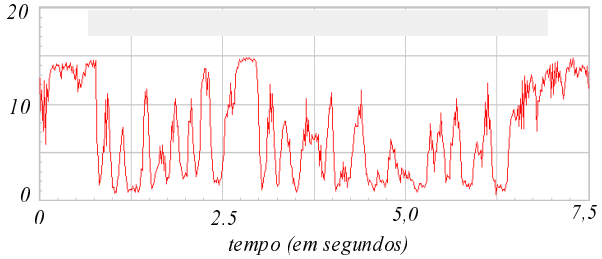


*Figura 8 – Curva de atenuação do ruído*

## REFERÊNCIAS BIBLIOGRÁFICAS


[1]-SCALART, P., & VIEIRA FILHO, J., *"Speech Enhancement Based on a Priori Signal-to-Noise Ratio Estimation"*, in: 1996 IEEE Int. Conf. on Acoust., Speech and Signal Processing, pp: 629-632, Atlanta-USA, May, 1996.

[2]-TIGER 31/PC *Reference/User's Manual*, DSP Research Inc., 1995

[3]-RIDE *User's Manual and Function Reference*, Hyperception Inc., 1998

[4]-BOLL, S. F. "*Suppression of Acoustic Noise in Speech using Spectral Subtraction*", *IEEE Trans. Acoust., Speech, and Signal Processing* (ASSP), vol. 29, p113-120, April-1979.

[5]-FLANAGAN, J. L. "Speech Analysis Synthesis and Perception", Ed. Springer-Verlag, 1972.

[6]-EPHRAIM, Y & MALAH, D., "*Speech Enhancement Using Minimum Mean Square Error Short-Time Spectral Amplitude Estimator*", IEEE Trans. on Acoust., Speech and Signal Processing, vol. 32, N° 6, December-1984.

[7]-DAVIDEK, V. et alli, " *Implementing a Noise Cancellation System with the TMS320C31*", in: First European DSP Education and Research, Conference, Paris, sept. 1996.