

Implementação em DSP de um Cancelador de Eco de Linha Utilizando o Algoritmo IPNLMS

Rafael D. Demetri, Walter A. Gontijo e Rui Seara

Resumo— Este trabalho discute a implementação de um cancelador de eco de linha em um processador digital de sinais (DSP). O algoritmo aqui utilizado é o *improved proportional normalized LMS (IPNLMS)*, que difere dos algoritmos clássicos de filtragem adaptativa por considerar um passo de adaptação diferente para cada coeficiente. O algoritmo IPNLMS exibe melhor desempenho do que o algoritmo NLMS, usualmente considerado em aplicações de cancelamento de eco. Contudo, o primeiro pode divergir na presença de *double-talk*. É proposta uma modificação no algoritmo IPNLMS, visando torná-lo mais robusto às situações de *double-talk*. Resultados experimentais confirmam o bom desempenho do cancelador de eco implementado tanto para uma resposta sintética do caminho do eco quanto para uma resposta real.

Palavras-chave— Algoritmo IPNLMS, Cancelamento de eco, Implementação em DSP.

Abstract— This work discusses the implementation of a line echo canceller by using a digital signal processor (DSP). The used algorithm is the improved proportional normalized LMS (IPNLMS) one, which differs from classical adaptive algorithms by using a different adaptation step size for each coefficient. The IPNLMS algorithm exhibits a better performance than the NLMS one, which is usually considered in echo cancellation applications. However, the former may diverge under double-talk conditions. A modification in the IPNLMS algorithm is then proposed, making it more robust in a double-talk situation. Experimental results confirm the good performance of the implemented echo canceller for both synthetic and real echo paths.

Keywords— IPNLMS algorithm, Echo cancellation, DSP implementation.

I. INTRODUÇÃO

Canceladores de eco são dispositivos que visam remover ecos indesejáveis em uma ligação telefônica. Os tipos de eco mais comuns em telefonia são o eco acústico (que ocorre em sistemas em que existe um acoplamento acústico entre o microfone e o alto-falante de um sistema viva-voz) e o eco de linha [1]. O eco de linha surge em ligações no sistema de telefonia analógica, devido a imperfeições na híbrida (dispositivo responsável pela conversão entre o circuito a dois fios e o circuito a quatro fios desse sistema). Em situações que o atraso entre a voz e o seu eco é muito grande (mais do que 20 ms), a existência de eco pode causar desconforto aos usuários e prejudicar a fluência de uma conversação. Nesses casos, é desejável a utilização de um cancelador de eco de

linha (LEC—*line echo canceller*), visando melhorar a qualidade da conversação e o conforto dos usuários [1].

Recentemente, renovou-se o interesse em pesquisa e desenvolvimento de canceladores de eco, principalmente devido à popularização dos sistemas de voz sobre IP (VoIP). Nesses sistemas, o atraso entre a voz e o seu eco é relativamente grande (podendo chegar a mais de 300 ms), e a qualidade de conversação sem o uso de um cancelador de eco torna-se muito pobre [2].

Normalmente as implementações de algoritmos de cancelamento de eco são realizadas em um processador digital de sinais (DSP) [3]–[5]. Tal escolha é devida aos recursos computacionais fornecidos por esse processador, que facilitam a implementação e a execução de algoritmos de processamento de sinais em tempo real.

O diagrama de blocos de um sistema de cancelamento de eco típico é mostrado na Fig. 1. A notação utilizada é a seguinte: $x(n)$ é o sinal de voz do usuário distante (*far-end*), $\hat{h}(n)$ é a resposta ao impulso do filtro adaptativo, $h(n)$ é a resposta ao impulso do caminho do eco, $y(n)$ é a soma do sinal de eco $d(n)$ com o sinal de voz do usuário próximo $v(n)$ (*near-end*) e $\hat{d}(n)$ é a réplica do eco.

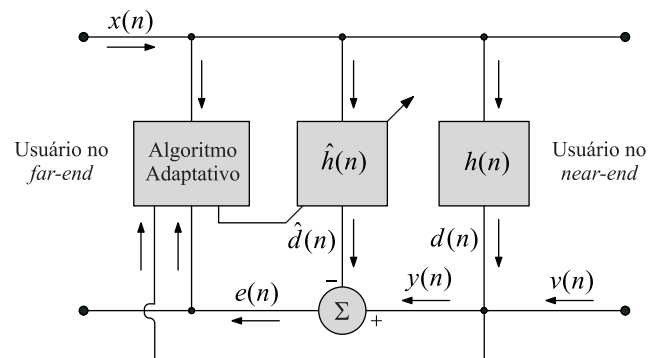


Fig. 1. Diagrama de blocos de um sistema de cancelamento de eco.

Os coeficientes do filtro adaptativo $\hat{h}(n)$ são adaptados de modo a minimizar o valor quadrático médio do sinal de erro $e(n)$. Um dos algoritmos de filtragem adaptativa mais utilizados para tal função é o LMS (*least-mean-square*) [6]. Essa escolha se deve à sua simplicidade e baixa complexidade computacional. Em aplicações de cancelamento de eco, utiliza-se geralmente o algoritmo NLMS (*normalized LMS*), que é um aprimoramento do algoritmo LMS, permitindo uma maior velocidade de convergência para sinais de entrada com diferentes níveis de potência média [6].

Novas estratégias visando melhorar o desempenho dos canceladores de eco vêm sendo desenvolvidas [7]–[9]. Em

Rafael D. Demetri, Walter A. Gontijo e Rui Seara, LINSE – Laboratório de Circuitos e Processamento de Sinais, Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, SC, E-mails: {rdemetri, walter, seara}@linse.ufsc.br.

Este trabalho foi parcialmente financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e pela empresa Dígitro Tecnologia Ltda.

sua maioria, essas estratégias são modificações e melhorias dos algoritmos clássicos de filtragem adaptativa. Uma variação do algoritmo NLMS é o *proportional* NLMS (PNLMS), que leva a resultados significativamente melhores do que o NLMS quando a resposta ao impulso do caminho do eco é esparsa [10]. No entanto, nem sempre é possível considerar que tal característica seja satisfeita, principalmente quando a duração da resposta ao impulso do caminho do eco não ultrapassa 20 ms. Para respostas com características dispersivas, o algoritmo PNLMS apresenta uma velocidade de convergência muito menor do que o NLMS [11]. Recentemente, foi proposto o algoritmo *improved proportional* LMS (IPNLMS), que busca atingir melhores resultados do que o NLMS e o PNLMS para qualquer tipo de resposta ao impulso considerada [11].

O desempenho de um cancelador de eco é afetado se a detecção dos instantes de atividade de fala nos dois lados da linha (*double-talk*) não é adequadamente realizada. Durante o *double-talk*, o algoritmo adaptativo deve interromper o processo de adaptação, evitando a degradação na estimativa do caminho do eco, visto que o sinal disponível para o cancelamento de eco não é composto puramente por eco [1]. A existência do *double-talk* é um ponto crítico para a utilização do IPNLMS, pois esse algoritmo apresenta uma degradação significativa de seu desempenho quando na presença de *double-talk*.

Este artigo é organizado como segue. A Seção II aborda os algoritmos PNLMS e IPNLMS. A Seção III propõe uma modificação ao algoritmo IPNLMS que o torna mais robusto às perturbações decorrentes do *double-talk*. A Seção IV apresenta o algoritmo para detecção de *double-talk* aplicado nesta implementação. Na Seção V, são discutidas as principais características da implementação, incluindo modificações que a tornam mais simples para os algoritmos considerados. Na Seção VI, são mostrados os resultados obtidos na implementação, tanto para sinais gerados a partir de respostas ao impulso sintéticas quanto para sinais oriundos de aquisições de conversas telefônicas. Finalmente, a Seção VII apresenta os comentários e as conclusões do trabalho em questão.

II. ALGORITMOS PNLMS E IPNLMS

A resposta ao impulso do caminho do eco que é sintetizada por um cancelador de eco de linha pode ter até 64 ms de duração. Na maioria dos casos, porém, essa resposta é não-nula apenas em um período de aproximadamente 4 ms [10]. Visando explorar a característica *esparsa* de tais respostas, foi desenvolvido o algoritmo PNLMS (*proportional normalized least-mean-square*).

A equação de adaptação dos coeficientes no algoritmo NLMS é dada por [6]

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \frac{\mu_{\text{NLMS}}}{\mathbf{x}(n)^T \mathbf{x}(n) + \delta_{\text{NLMS}}} \mathbf{x}(n)e(n) \quad (1)$$

onde $\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$ denota o vetor de excitação, L representa o número de coeficientes do filtro adaptativo, $\hat{\mathbf{h}}(n) = [\hat{h}_0(n), \hat{h}_2(n), \dots, \hat{h}_{L-1}(n)]^T$ caracteriza o vetor de coeficientes do filtro adaptativo, δ_{NLMS} é uma constante positiva que evita uma divisão por um número muito pequeno quando o valor da norma de $\mathbf{x}(n)$ é pequeno e μ_{NLMS} denota o passo de adaptação do algoritmo NLMS.

É interessante notar que (1) prevê um único passo de adaptação μ_{NLMS} para todos os coeficientes. Ao se permitir diferentes passos para diferentes coeficientes, novas estratégias de adaptação são possíveis. Uma dessas estratégias é tornar o passo de cada coeficiente proporcional ao seu valor, originando o algoritmo NLMS proporcional (PNLMS-*proportional NLMS*).

A equação de adaptação do algoritmo PNLMS é obtida por [10]

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \frac{\mu_{\text{PNLMS}} \mathbf{G}(n)}{\mathbf{x}(n)^T \mathbf{G}(n-1) \mathbf{x}(n) + \delta_{\text{PNLMS}}} \mathbf{x}(n)e(n) \quad (2)$$

onde $\mathbf{G}(n)$ é uma matriz diagonal, que controla a proporcionalidade do passo de adaptação, dada por

$$\mathbf{G}(n) = \text{diag} \{g_0(n), g_1(n), \dots, g_{L-1}(n)\}. \quad (3)$$

Os coeficientes da matriz $\mathbf{G}(n)$ podem ser obtidos através das seguintes expressões:

$$\bar{\gamma}_l = \max \left[\delta_p, \left| \hat{h}_0(n) \right|, \dots, \left| \hat{h}_{L-1}(n) \right| \right] \quad (4)$$

$$\gamma_l(n) = \max \left\{ \rho \bar{\gamma}_l, \left| \hat{h}_l(n) \right| \right\} \quad (5)$$

$$g_l(n) = \frac{|\gamma_l(n)|}{\sum_{i=0}^{L-1} \gamma_i(n)}, \quad 0 \leq l \leq L-1. \quad (6)$$

O parâmetro ρ , em (5), evita que a adaptação de um determinado coeficiente $\hat{h}_l(n)$ seja muito lenta quando esse é muito menor do que os demais coeficientes. Já δ_p , em (4), regulariza a adaptação para o caso em que todos os coeficientes são nulos [10].

O algoritmo PNLMS fornece resultados melhores do que o NLMS quando a resposta ao impulso do caminho do eco é esparsa [10]. Em média, as respostas ao impulso encontradas nas redes de telefonia concentram sua energia em uma janela de aproximadamente 4 ms. Dessa forma, se a resposta ao impulso modelada pelos coeficientes do filtro adaptativo tem uma duração de, por exemplo, 64 ms, o algoritmo PNLMS pode ser aplicado. Respostas ao impulso com comprimento muito menor, no entanto, não podem ser consideradas esparsas e o PNLMS não fornece resultados satisfatórios em tais situações [11]. Essas observações sugerem que o PNLMS pode ser aperfeiçoado de modo a obter um bom desempenho para qualquer tipo de resposta. Uma proposta modificando o algoritmo PNLMS, tornando-o mais apropriado para diversos tipos de resposta, é o algoritmo IPNLMS [11].

As principais diferenças entre o algoritmo PNLMS e o IPNLMS estão na substituição do operador $\max \{ \cdot \}$ em (4) e (5) por uma opção mais suave como também a inclusão de um termo não proporcional. Assim, a equação de atualização dos coeficientes no algoritmo IPNLMS é [11]

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \frac{\mu_{\text{IPNLMS}} \mathbf{K}(n)}{\mathbf{x}(n)^T \mathbf{K}(n) \mathbf{x}(n) + \delta_{\text{IPNLMS}}} \mathbf{x}(n)e(n). \quad (7)$$

A matriz $\mathbf{K}(n)$, análoga à matriz $\mathbf{G}(n)$ em (2), é dada por

$$\mathbf{K}(n) = \text{diag} \{k_0(n), k_1(n), \dots, k_{L-1}(n)\} \quad (8)$$

onde

$$k_l(n) = \frac{1 - \alpha}{2L} + (1 + \alpha) \frac{|\hat{h}_l(n)|}{2 \|\hat{\mathbf{h}}(n)\|_1} \quad (9)$$

com

$$\|\hat{\mathbf{h}}(n)\|_1 = \sum_{i=0}^{L-1} |\hat{h}_i(n)| \quad (10)$$

para $-1 \leq \alpha \leq 1$.

Note que quando $\alpha = -1$, os algoritmos IPNLMS e NLMS são idênticos. Já para α próximo de 1, o algoritmo comporta-se como o PNLMS. Em geral, (9) é considerada a soma de dois termos, sendo um fixo e um variável (função dos coeficientes). O parâmetro α controla a importância de cada termo; valores típicos são $\alpha = 0$ ou $\alpha = 0,5$ [11].

III. ALGORITMO IPNLMS ROBUSTO

Uma característica negativa dos algoritmos PNLMS e IPNLMS é a possibilidade (grande) de divergência em situações de *double-talk*. De maneira geral, os algoritmos de *double-talk* não são capazes de “congelar” a adaptação de maneira instantânea quando inicia a atividade do usuário do *near-end* e os coeficientes são adaptados (por um pequeno período) na presença de *double-talk*. Resultados experimentais com os algoritmos PNLMS e IPNLMS mostram uma significativa degradação no desempenho do cancelador em tais situações, mesmo que o período de adaptação na presença de *double-talk* seja apenas de poucas amostras [12].

Uma estratégia que visa minimizar os efeitos do *double-talk* no algoritmo PNLMS foi proposta em [12]. Lá, é utilizado um fator de escala não-linear $s(n)$ aplicado ao sinal de erro. Dessa forma, obtém-se o algoritmo PNLMS robusto, não divergindo como o PNLMS convencional na presença de curtos período de *double-talk*. Neste artigo, é proposta uma alteração similar para o algoritmo IPNLMS, dando origem ao algoritmo IPNLMS robusto.

A equação de adaptação do algoritmo IPNLMS robusto é

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \frac{\mu_{\text{IPNLMS}} \mathbf{K}(n) \mathbf{x}(n) e_s(n)}{\mathbf{x}(n)^T \mathbf{K}(n) \mathbf{x}(n) + \delta_{\text{IPNLMS}}} \quad (11)$$

onde $e_s(n)$ é o erro escalado, dado por

$$e_s(n) = \min \left\{ \frac{|e(n)|}{s(n)}, k_0 \right\} \text{sign} \{e(n)\} s(n) \quad (12)$$

com $\text{sign} \{e(n)\}$ sendo o sinal de $e(n)$, $s(n)$, o fator de escala e k_0 , uma constante positiva. Dessa forma, o erro escalado é o próprio erro $e(n)$ se $|e(n)| > [s(n)k_0]$ e, caso contrário, $s(n)k_0 \text{sign} \{e(n)\}$.

O fator de escala $s(n)$ é determinado recursivamente por

$$s(n) = s(n-1)\lambda + k_0 \left(\frac{1-\lambda}{\beta} \right) \Phi_{\text{en}} \quad (13)$$

$$\Phi_{\text{en}} = \min \{|e(n)|, s(n)\} \quad (14)$$

onde β é uma constante obtida a partir de k_0 [12].

Os parâmetros λ e k_0 controlam a taxa de atualização de $s(n)$. Uma atualização lenta para $s(n)$ tende a proporcionar ao algoritmo maior robustez ao *double-talk*, contudo podendo levar a uma menor taxa de convergência [12].

IV. DETECÇÃO DE DOUBLE-TALK

A correta detecção dos instantes de *double-talk* é de fundamental importância para o bom funcionamento de um cancelador de eco. Se o sinal $y(n)$ recebido pelo cancelador não for composto apenas pelo eco, pode haver sensível degradação em seu desempenho caso a adaptação dos coeficientes não seja “congelada”.

Diversas estratégias para detecção de *double-talk* vêm sendo propostas. A estratégia utilizada nesta implementação é o método da correlação cruzada normalizada [13]. A principal vantagem dessa abordagem é não depender de um conhecimento *a priori* das características dos sinais envolvidos e do caminho do eco.

A variável de decisão do método da correlação cruzada normalizada é

$$\xi = \sqrt{\mathbf{r}_{\mathbf{x}y}^T (\sigma_y^2 \mathbf{R}_{\mathbf{x}x})^{-1} \mathbf{r}_{\mathbf{x}y}} \quad (15)$$

onde $\mathbf{R}_{\mathbf{x}x}$ é a matriz de autocorrelação de $\mathbf{x}(n)$, σ_y^2 é a variância de $y(n)$ e $\mathbf{r}_{\mathbf{x}y}$ é o vetor de correlação cruzada entre $\mathbf{x}(n)$ e $y(n)$. Mostra-se que (15) é equivalente a

$$\xi = \frac{\sqrt{\mathbf{h}^T \mathbf{R}_{\mathbf{x}x} \mathbf{h}}}{\sqrt{\mathbf{h}^T \mathbf{R}_{\mathbf{x}x} \mathbf{h} + \sigma_v^2}} \quad (16)$$

Observa-se que $\xi = 1$ quando $v(n) = 0$ e $\xi < 1$ quando $v(n) \neq 0$. O valor de ξ para $v = 0$ não depende da resposta ao impulso $\mathbf{h}(n)$. Dessa forma, é realizada a comparação de ξ com um parâmetro pré-definido T ($T \leq 1$) e assume-se *double-talk* quando $\xi < T$.

Existem diversas formas de implementar (15), evitando a necessidade do cálculo de $\mathbf{R}_{\mathbf{x}x}^{-1}$ e $\mathbf{r}_{\mathbf{x}y}$ [13]. Supondo que o filtro adaptativo convergiu, tem-se

$$\hat{\mathbf{h}}(n) \approx (\mathbf{R}_{\mathbf{x}x})^{-1} \mathbf{r}_{\mathbf{x}y} \quad (17)$$

Refazendo o desenvolvimento do método da correlação cruzada normalizada utilizando (17), é possível mostrar que [14]

$$\xi = \sqrt{\frac{\sigma_d^2}{\sigma_y^2}} \quad (18)$$

pode ser implementada de maneira simples através da razão direta dos desvios padrão de $\hat{d}(n)$ e $y(n)$.

Assumindo que os sinais envolvidos têm média zero, (18) pode ser interpretada como uma comparação entre a potência média da réplica do eco e àquela do sinal do *near-end*. Assim, quando não existe sinal de voz do usuário no *near-end*, a potência média do eco e de sua réplica devem ser iguais. Quando existe sinal de voz do usuário no *near-end*, é natural que a potência média do sinal $y(n)$ aumente, fazendo com que $\xi < 1$. Observa-se ainda que o *double-talk* é automaticamente assumido quando não existe voz do usuário no *far-end*, pois nesse caso $\hat{d}(n) = 0$.

A simplicidade de (18) torna-a atrativa para implementações práticas. No entanto, a derivação apresentada supõe que os coeficientes $\hat{\mathbf{h}}(n)$ convergiram para a resposta ao impulso do caminho do eco $\mathbf{h}(n)$. Na prática, nem sempre pode-se fazer tal suposição. O exemplo mais trivial é a inicialização do algoritmo, quando $\hat{\mathbf{h}}(n) = 0$. Nesse caso, $\hat{d}(n) = 0$, fazendo

com que $\xi = 0 < T$. Assim, o *double-talk* seria permanentemente detectado e não haveria adaptação dos coeficientes em qualquer momento.

Uma estratégia proposta para contornar tal problema é fazer $T = 0$ no início da adaptação e incrementá-lo gradualmente até que $T = T_{\text{final}}$. Assim, a adaptação inicial dos coeficientes não depende da detecção *double-talk*, dando ao algoritmo algum tempo para convergir. À medida que T aumenta e $\hat{\mathbf{h}}$ torna-se próximo a $\mathbf{h}(n)$, a detecção de *double-talk* torna-se possível.

Ainda que o equacionamento apresentado mostre que $\xi = 1$ quando $v(n) = 0$, o valor de T_{final} deve ser ajustado experimentalmente levando em conta a existência de ruídos e erros de estimação [13].

V. CARACTERÍSTICAS DA IMPLEMENTAÇÃO

O cancelador de eco é implementado no processador digital de sinais (DSP) TMS320C5410A-126 [15], que pertence à família TMS320C54x da *Texas Instruments*. Os processadores dessa família são DSPs de 16 bits ponto fixo, com frequências de operação de 80 MHz a 160 MHz e memórias RAM e ROM integradas (*on-chip*). O processador utilizado opera em 160 MHz, possuindo 128 kB de memória RAM e 32 kB de memória ROM *on-chip*.

A programação dos algoritmos no DSP é realizada diretamente em linguagem *assembly*. Assim, uma maior otimização do código é possível, tanto em termos de complexidade computacional quanto em utilização de memória. O ambiente de desenvolvimento utilizado é o *Code Composer Studio* (CCS) [16]. Nesse ambiente, é possível realizar simulações do código implementado, permitindo uma avaliação realista de seu desempenho.

Uma das primeiras etapas no desenvolvimento de um cancelador de eco é determinar a duração máxima da resposta ao impulso do caminho de eco para uma dada aplicação. Testes realizados com sinais adquiridos de ligações telefônicas mostraram que um filtro adaptativo com $L = 128$ coeficientes modela apropriadamente o caminho de eco.

Visando tornar os algoritmos de filtragem adaptativa e de detecção de *double-talk* mais adequados às aplicações práticas, são realizadas algumas modificações em seu equacionamento.

Uma primeira modificação no algoritmo IPNLMS é permitir a atualização dos coeficientes apenas quando a potência média do sinal de entrada $\mathbf{x}(n)$ for maior do que -40 dB, reduzindo assim o número de operações nos períodos de silêncio. A potência média do sinal de entrada é determinada através da seguinte equação recursiva:

$$P_{\mathbf{x}} = (1 - \nu) P_{\mathbf{x}} + \nu [x(n)]^2 \quad (19)$$

onde $\nu = 2^{-7}$. As variâncias do sinal de réplica do eco $\hat{d}(n)$ e do sinal no *near-end* $y(n)$ são determinadas de maneira semelhante a $P_{\mathbf{x}}$ (supondo que esses sinais têm média zero).

As equações (7) e (9) são reescritas de modo a minimizar o número de divisões necessárias. Multiplicando (9) por $2 \|\hat{\mathbf{h}}(n)\|_1$, obtém-se

$$k_l(n) = \frac{1 - \alpha}{L} \|\hat{\mathbf{h}}(n)\|_1 + (1 + \alpha) |\hat{h}_l(n)|. \quad (20)$$

Como (20) é sempre zero no início da operação do LEC [pois, nessa situação, $\hat{\mathbf{h}}(n) = 0$], é necessário fazer

$$k_l(n) = \max \{k_l(n), k_{\min}\}. \quad (21)$$

Considerando que $\mathbf{x}(n)^T \mathbf{K}(n) \mathbf{x}(n) \approx P_{\mathbf{x}}$, o denominador de (7) e (11) torna-se

$$P_{\mathbf{x}} \left\| \hat{\mathbf{h}}(n) \right\|_1 + \delta_{\text{IPNLMS}}. \quad (22)$$

A seguir, são detalhados os passos utilizados na implementação do algoritmo IPNLMS robusto:

- i) Adquirir as amostras $x(n)$ e $y(n)$.
- ii) Calcular a estimativa do eco $\hat{d}(n)$.
- iii) Calcular o erro residual.
- iv) Obter a estimativa de potência média do sinal do *far-end* através de (19).
- v) Calcular as variâncias (potências médias) da réplica do eco e do sinal no *near-end*.
- vi) Calcular a variável ξ através de (18).
- vii) Detectar *double-talk* através da comparação de ξ com o limiar T .
- viii) Se a potência do *far-end* é maior do que -40 dB e não for detectado *double-talk*, são executados os seguintes passos:

- 1) Incrementar T se $T < T_{\text{final}}$.
- 2) Atualizar $s(n)$ através de (13).
- 3) Calcular $\|\hat{\mathbf{h}}(n)\|_1$ através de (10).
- 4) Calcular a matriz $\mathbf{K}(n)$ através de (20) e (21).
- 5) Calcular o erro escalado $e_s(n)$ através de (12).
- 6) Atualizar os coeficientes segundo (11), substituindo seu denominador por (22).

VI. RESULTADOS

Para avaliação do desempenho do cancelador de eco, são realizadas simulações no CCS do código implementado. Dessa forma, os resultados obtidos são os mesmos que se verificaria em uma situação real.

Os testes realizados incluem simulações com sinais reais adquiridos de conversações telefônicas e sinais de eco obtidos a partir de uma resposta ao impulso sintética. Para cada caso, é realizada uma comparação entre os algoritmos NLMS, IPNLMS e IPNLMS robusto.

A resposta ao impulso utilizada foi extraída de [17] e corresponde a um modelo sintético de resposta ao impulso do caminho do eco em sistemas de telefonia.

O desempenho dos algoritmos é medido através do ERLE (*echo return loss enhancement*), definido como

$$\text{ERLE} = 10 \log \frac{P_{\text{near}}}{P_{\text{erro}}} \text{ [dB]}. \quad (23)$$

Os algoritmos NLMS, IPNLMS e IPNLMS robusto possuem alguns parâmetros no seu equacionamento que devem ser ajustados. Os valores utilizados foram obtidos através de uma série de experimentos, de modo a obter um conjunto de parâmetros que apresente resultados satisfatórios em diversas situações. Os parâmetros utilizados no algoritmo NLMS são $\mu_{\text{NLMS}} = 0,01$ e $\delta_{\text{NLMS}} = 0,125$. No algoritmo IPNLMS, os parâmetros são $\mu_{\text{IPNLMS}} = 0,8$, $\delta_{\text{IPNLMS}} = 0,01$ e $\alpha = 0$. Já no

algoritmo IPNLMS robusto, utiliza-se também $k_0 = 0,8$ (o que implica em $\beta \approx 0,56$), $\lambda = 0,995$ e $s(-1) = 0,03$.

A. Testes com Resposta ao Impulso Sintética

A resposta ao impulso sintética utilizada nos testes é mostrada na Fig. 2. O comprimento dessa resposta é $L_{resp} = 65$. A figura 3 mostra a atenuação obtida com os algoritmos NLMS, IPNLMS e IPNLMS robusto, considerando a resposta ao impulso sintética.

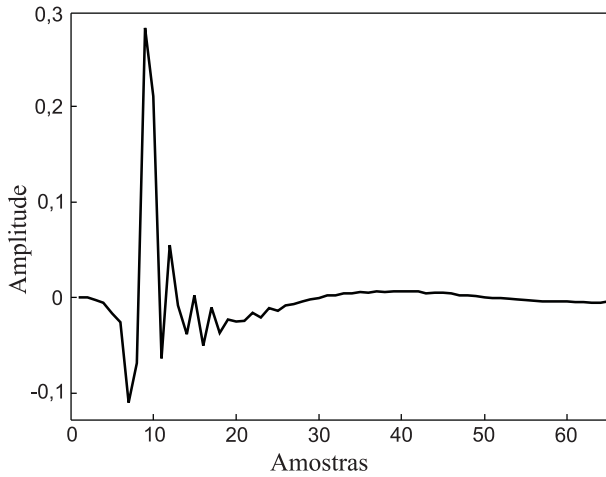


Fig. 2. Resposta ao impulso sintética utilizada.

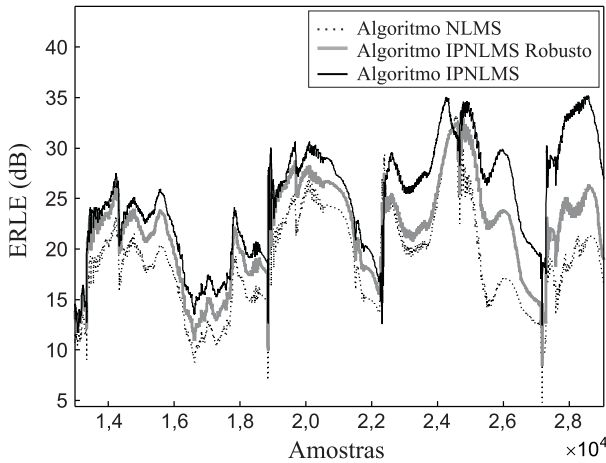


Fig. 3. Atenuação para resposta ao impulso sintética.

Observa-se que o algoritmo IPNLMS obteve resultados de atenuação superiores ao algoritmo IPNLMS robusto, que por sua vez obteve melhor resultado que o algoritmo NLMS convencional. Em alguns trechos, a diferença da atenuação obtida entre o algoritmo IPNLMS e o NLMS chega a mais do que 10 dB.

B. Testes com Aquisições Reais

Os testes com aquisições reais de conversações telefônicas permitem uma avaliação mais precisa do desempenho do cancelador operando em condições normais. A Fig. 4 mostra

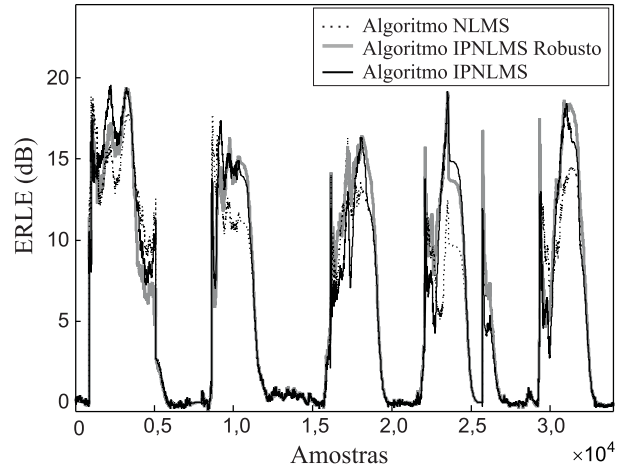


Fig. 4. Atenuação obtida com sinais reais.

o resultado para uma situação em que existe apenas eco (ou seja, não existe atividade do usuário no *near-end*).

Observa-se que os algoritmos IPNLMS e IPNLMS robusto obtiveram desempenho similar e ambos alcançaram melhores resultados do que o algoritmo NLMS.

A Fig. 5 mostra um sinal em que existe eco e voz do usuário no *near-end*. Está indicado na figura o início do trecho em que existe simultaneamente voz e eco (*double-talk*). A atenuação obtida nessa condição está mostrada na Fig. 6. Observa-se, nesse caso, que o algoritmo IPNLMS robusto fornece melhores resultados na presença de *double-talk*.

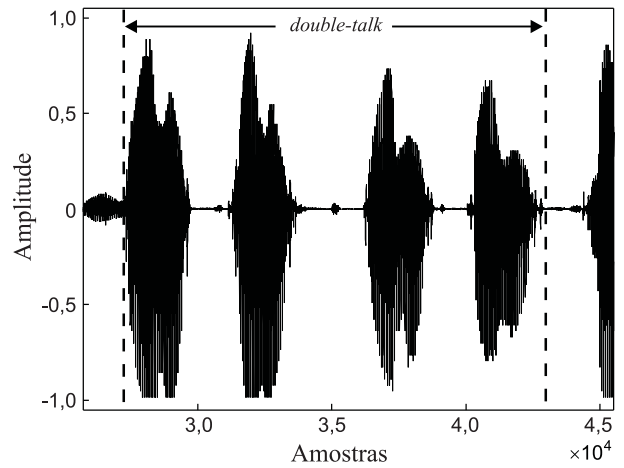
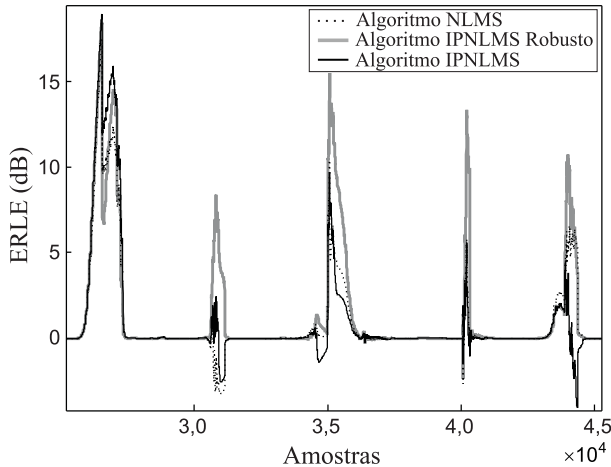


Fig. 5. Sinal no *near-end* com *double-talk*.

C. Complexidade Computacional

A implementação em *assembly* do algoritmo NLMS, para um filtro adaptativo de 128 coeficientes, consome cerca de 6,5 MIPS (milhões de instruções por segundo) do DSP. Já para a implementação do algoritmo IPNLMS, o resultado obtido é de cerca de 10 MIPS. O algoritmo IPNLMS robusto consome cerca de 11 MIPS.

O aumento em complexidade computacional decorre principalmente das operações matemáticas necessárias para calcular a matriz $\mathbf{K}(n)$. Uma possível modificação visando reduzir a

Fig. 6. Atenuação obtida com sinais reais e *double-talk*.

complexidade computacional necessária para a implementação do algoritmo é atualizar a matriz $\mathbf{K}(n)$ a cada M amostras [10]. Testes mostram que valores de M até 128 não prejudicam o desempenho do algoritmo e reduzem sensivelmente a complexidade computacional do cancelador de eco. Uma outra vantagem decorrente dessa abordagem é que o algoritmo se torna menos susceptível a erros de precisão finita, devido à diminuição no número de operações aritméticas realizadas.

A Fig. 7 mostra uma comparação da atenuação obtida pelo algoritmo IPNLMS robusto e pelo IPNLMS com complexidade reduzida (denotado por IPNLMS robusto_r) usando $M = 64$. Observa-se que ambos comportam-se de maneira similar, com IPNLMS robusto_r atingindo valores maiores de atenuação em determinados instantes. Para o valor de M utilizado, a complexidade computacional requerida pelo cancelador do eco reduz-se a cerca de 8 MIPS.

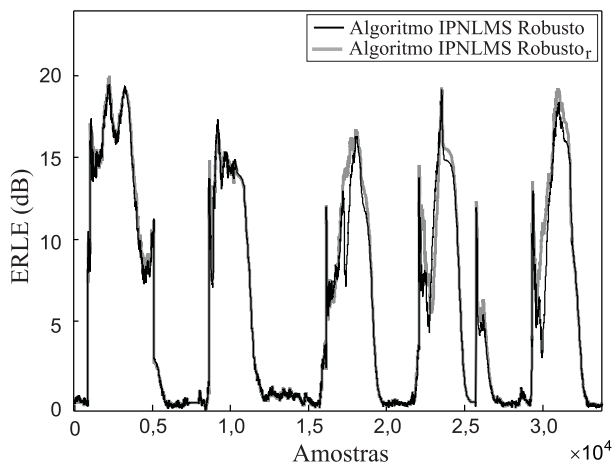


Fig. 7. Comparação entre IPNLMS robusto.

VII. CONCLUSÕES

Neste artigo, foi apresentada uma implementação em DSP do algoritmo IPNLMS para aplicação em cancelamento de eco de linha. Foi também proposta uma modificação desse algoritmo, tornando-o mais robusto às perturbações decorrentes

do *double-talk*. O algoritmo resultante é denominado IPNLMS robusto.

Os resultados apresentados mostram que os algoritmos IPNLMS e IPNLMS robusto atingem, na situação em que não existe *double-talk*, resultados melhores do que o algoritmo NLMS convencional. Nas situações de *double-talk*, o algoritmo IPNLMS robusto não apresenta divergência e seu desempenho é melhor do que os outros algoritmos considerados. Dessa forma, o algoritmo IPNLMS robusto torna-se mais adequado para aplicações de cancelamento de eco.

Foi também proposta uma modificação no algoritmo IPNLMS robusto, reduzindo o número de operações aritméticas advindas do cálculo da matriz $\mathbf{K}(n)$. Dessa forma, a implementação do algoritmo IPNLMS robusto (modificado) apresenta uma carga computacional compatível com as implementações do algoritmo NLMS.

REFERÊNCIAS

- [1] L. S. Gay and J. Benesty, *Acoustic Signal Processing for Telecommunication*, 1st ed. Norwell, USA: Kluwer Academic Publishers, 2000.
- [2] J. H. James, B. Chen, and L. Garrison, "Implementing VoIP: A voice transmission performance progress report," *IEEE Commun. Mag.*, vol. 42, no. 7, pp. 36–41, July 2004.
- [3] R. D. Demetri, W. A. Gontijo e R. Seara, "Implementação de um cancelador de eco acústico baseado em um processador digital de sinais," *Anais do XVI Congresso Brasileiro de Automática*, Salvador, BA, Out. 2006, pp. 2990–2995.
- [4] *Implementing a Line-Echo Canceller Using the Block Update and NLMS Algorithms on the TMS320C54x DSP*, Texas Instruments, abr. 1997.
- [5] Y. Haneda, S. Makino, J. Kojima, and S. Shimauchi, "Implementation and evaluation of an acoustic echo canceller using duo-filter control system," in *Proc. European Signal Processing Conf.*, vol. 2, Trieste, Italy, Sept. 1996, pp. 1115–1118.
- [6] S. Haykin, *Adaptive filter theory*, 3rd ed. New Jersey, USA: Prentice Hall, 2000.
- [7] A. Maleki and K. Nayebi, "A new efficient pnms based algorithm for adaptive line echo cancellation," in *Proc. Seventh International Symposium on Signal Processing and Its Applications*, vol. 2, Paris, France, July 2003, pp. 555–558.
- [8] P. De Leon and D. Etter, "Experimental results of subband acoustic echo cancelers under spherically invariant random processes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, Atlanta, USA, May 1996, pp. 961–964.
- [9] F. Kuch and W. Kellermann, "Nonlinear line echo cancellation using a simplified second order volterra filter," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, Orlando, USA, May 2002, pp. 1117–1120.
- [10] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 5, pp. 508–518, Sept. 2000.
- [11] J. Benesty and S. Gay, "An improved PNLS algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, Orlando, USA, May 2002, pp. 1881–1884.
- [12] T. Gansler, S. L. Gay, M. Sondhi, and J. Benesty, "Double-talk robust fast converging algorithms for network echocancellation," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 656–663, Nov. 2000.
- [13] J. Benesty, D. R. Morgan, and J. H. Cho, "A new class of doubletalk detectors based on cross-correlation," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 2, pp. 168–172, Mar. 2000.
- [14] M. Waseem, M. Omer, A. Bhatti, and S. Shah, "A new doubletalk detector with fuzzy decision thresholding," in *Proc. IEEE Students Conf.*, vol. 1, Lahore, Pakistan, Aug. 2002, pp. 117–121.
- [15] Texas Instruments. (2007, 02) Digital Signal Processor - TMS320VC5410A - TI Product Folder. [Online]. Available: <http://focus.ti.com/docs/prod/folders/print/tms320vc5410a.html>
- [16] —. (2007, 02) Code Composer Studio IDE New for CCStudio from Texas Instruments. [Online]. Available: <http://focus.ti.com/dsp/docs/dspsupportnp.tsp?sectionId=3&tabId=451&familyId=44>
- [17] *G.168*, International Telecommunication Union, Genebra, June 2002.