

Implementação de um Filtro Adaptativo LMS Aplicado ao Cancelamento de Eco em Voz Sobre IP

Guilherme Rauter Corsetti, Moisés Coster, Ricardo Balbinot, Fladhimyr Câmara Castello, Jorge Guedes Silveira

Departamento de Engenharia Elétrica - Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre - Rio Grande do Sul - Brasil

{gcorsetti, mcoster, rbalbinot, fcastello, jguedes}@gparc.org

Resumo. O objetivo desse artigo é mostrar uma implementação de um cancelador de eco utilizando o algoritmo LMS (Least Mean Square). O algoritmo LMS é um dos pioneiros para o desenvolvimento de processos adaptativos. Sua característica simples de implementação é responsável por sua grande popularidade entre os desenvolvedores. O eco, na maioria das vezes, pode ser considerado um problema de grande relevância na telefonia tanto convencional quanto em transmissão por redes IP, sendo responsável pela má qualidade e um grande desconforto para os usuários durante as ligações. Para que seja testada a eficiência do cancelador que será implementado, serão simulados de diferentes atrasos do eco na rede. A partir dos resultados desses testes da simulação, poderá ser analisado o tempo que o cancelador necessitará para aprender as características do sinal de eco e eliminá-lo.

1. Introdução

Um filtro adaptativo pode se ajustar ao ambiente onde está sendo utilizado, transformando um sinal de entrada em um sinal de saída desejado. Para que ocorra essa transformação pode ser usado o algoritmo LMS, que será tratado com mais profundidade na Seção 2. Através de um filtro adaptativo, é possível, por exemplo, cancelar o eco em um sistema de telefonia.

O eco é um problema comum, que ocorre quando há uma reflexão do sinal de voz, por exemplo, nas redes telefônicas. Esse problema pode ter diversas origens. Existem dois tipos de eco na rede de telefonia convencional: o eco acústico e o eco híbrido. O primeiro é causado por ruídos existentes no ambiente, como reflexões do som nas paredes onde a pessoa está falando e pela realimentação do alto-falante para o microfone. O segundo é gerado em transformadores híbridos, usado na rede pública de telefonia comutada (RPTC) para a transmissão de dados, onde um link de quatro fios

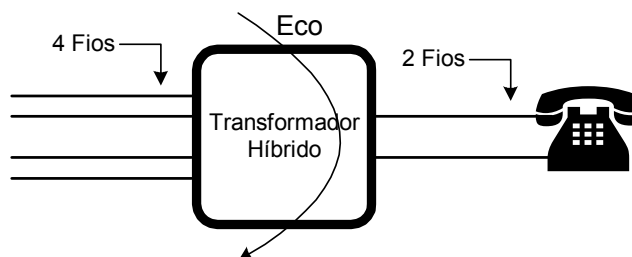


Figura 1 - Geração de eco no transformador híbrido

tem que ser transformado em um link de dois fios (para ligação à residência do usuário) (ver Figura 1).

O trabalho sobre processamento de sinais adaptativos, referenciado em [Widrow and Stearns 1985], foi um grande impulso na pesquisa de algoritmos para o processamento adaptativo. O algoritmo LMS desenvolvido nessa obra permitiu uma evolução significativa nessa área. Existem diversos outros algoritmos adaptativos além do LMS que podem ser implementados para o desenvolvimento de um cancelador de eco, como por exemplo, as redes neurais que estão sendo cada vez mais utilizadas em todos os tipos de processos adaptativos.

Neste artigo será implementado um filtro adaptativo LMS, cuja função será cancelar o eco de uma rede IP ligada a uma rede de telefonia convencional. O cancelador será testado com diferentes atrasos do eco, assim poderemos verificar se o mesmo será eficiente para o seu propósito. Para que possa ocorrer a simulação dos atrasos do eco, será implementado um algoritmo que possua tal capacidade.

2. Filtro Adaptativo LMS

O filtro adaptativo LMS tem como função, no cancelador de eco, alterar o sinal r' (ver Figura 2), de tal modo que ele gere um sinal y , na saída do filtro, que possa ser subtraído do sinal $(s + r)$ (ver Figura 2), cancelando o sinal r e assim minimizando o erro (ϵ). A velocidade com que o filtro realiza o processo de aproximação de y e r é o que garante o sucesso do cancelamento do eco. Na Figura 2 é mostrado um modelo de funcionamento de um filtro adaptativo no modo de cancelamento de ruído.

Inicialmente, considera-se um sinal (s) acrescido de um ruído (r) qualquer (neste caso será o eco) que vem da rede. A soma desses dois sinais ($s + r$) é chamada de sinal entrada (d). Precisa-se obter uma amostra de r de forma a termos o sinal de referência r' , que pode ser considerado uma versão distorcida, atrasada, mas correlacionada do sinal

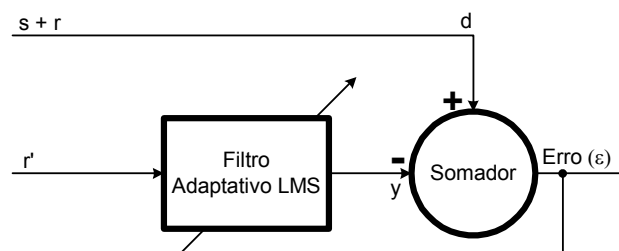


Figura 2 - Modelo de Filtro Adaptativo

r . O sinal r' passa pelo filtro que ajusta seus coeficientes para que o sinal de saída y seja igual a r , de forma que o erro (ϵ) seja apenas formado pelo sinal desejado s [Widrow and Stearns 1985].

Conclui-se que o erro é igual a: $\epsilon = s + r - y$

Chega-se ao erro ideal quando r e y são iguais, matematicamente ficará: $r = y$

Então: $\epsilon = s$

O erro ideal será igual ao sinal que estamos querendo que não seja anulado, ou seja, o sinal desejado s . As amostras dos sinais de entrada d são armazenadas em vetores que chamamos de x_k , onde k é um índice de tempo. Como existe apenas uma entrada desse sinal, ela é chamada de entrada simples. A matriz de vetores de entrada ficará assim:

$$\mathbf{X}_k = [\mathbf{x}_k \quad \mathbf{x}_{k-1} \quad \cdots \quad \mathbf{x}_{k-L}]^T$$

O combinador linear adaptativo (ALC) é utilizado no algoritmo LMS, para gerar a saída y . O ALC consiste na utilização de pesos no processo adaptativo. Os pesos ficam localizados dentro do filtro adaptativo LMS. Esses pesos funcionam de acordo com o ϵ , de forma que eles vão se ajustando até que o sinal y fique igual ao sinal r .

A definição do número de pesos utilizados no processo é muito importante, pois se forem adicionados muitos pesos o processo necessitará de muita memória e acabará, com isso, prejudicando a velocidade de execução do processo adaptativo. Caso sejam adicionados poucos pesos o processo de aprendizagem do filtro pode ficar lento demais. Os pesos são armazenados no que chamamos de vetor-de-pesos que corresponde a:

$$\mathbf{W}_k = [\mathbf{w}_{0k} \quad \mathbf{w}_{1k} \quad \cdots \quad \mathbf{w}_{Lk}]^T$$

onde \mathbf{W} é o vetor de peso, \mathbf{w}_{Lk} são os pesos, sendo k o índice de tempo e L o índice correspondente do último vetor de peso (o mais recente). Os pesos vão convergindo até chegar no vetor-de-pesos considerado ideal, \mathbf{W}^* . Esse ajuste “percorre” o que chamamos de função mean-square-error (MSE). Para que o vetor de pesos chegue ao \mathbf{W}^* , ele deve alcançar o valor mínimo do MSE, pois lá está localizado o \mathbf{W}^* . A função do MSE é definida por: $\text{MSE} = \xi = \mathbf{E}[\epsilon^2] = \mathbf{E}[\mathbf{d}^2] + \mathbf{W}^T \mathbf{R} \mathbf{W} - 2\mathbf{P}^T \mathbf{W}$

onde ϵ é o erro, \mathbf{d} é o sinal de entrada, \mathbf{W} o vetor-de-pesos, \mathbf{R} é uma matriz quadrada definida por: $\mathbf{R} = \mathbf{E}[\mathbf{X}_k \mathbf{X}_k^T]$

E o vetor \mathbf{P} é definido como: $\mathbf{P} = \mathbf{E}[\mathbf{d}_k \mathbf{X}_k]$

O algoritmo LMS é muito importante porque é um algoritmo simples e de fácil computação. Ele é representado por: $\mathbf{W}_{k+1} = \mathbf{W}_k - \mu \nabla_k = \mathbf{W}_k + 2\mu \epsilon_k \mathbf{X}_k$

onde, o ∇ é o gradiente do erro, \mathbf{W} é o vetor de pesos, k o número de iterações, μ é a constante de ganho que regula a velocidade e estabilidade do processo de adaptação, ϵ é o erro estimado e \mathbf{X} é o vetor de amostras de entrada. O parâmetro μ é introduzido na equação com a finalidade de dar estabilidade ao processo de aprendizagem do filtro, e seu valor está compreendido entre: $0 < \mu < 2/\text{tr}\{\mathbf{R}\}$, onde, $\text{tr}\{\mathbf{R}\}$ é o somatório da diagonal da matriz de autocorrelação.

3. Cancelador de Eco IP

Um cancelador de eco tem como função eliminar o eco que está sendo gerado em uma chamada. Normalmente o eco é gerado quando há uma conexão com a RTPC, na parte de hardware do sistema chamado híbrida (que converte a chamada de quatro fios para dois fios) e este eco pode ser eliminado já no gateway de entrada da Rede IP. O eco acústico também existe na rede IP, ele é gerado da mesma forma que na rede de telefonia convencional (originado de reflexões da fala no ambiente do locutor) e é agravado quando se utiliza um computador como meio de comunicação devido à realimentação do alto-falante para o microfone. O eco é variável de chamada para chamada e até mesmo em uma mesma chamada, ou seja, é necessário que se tenha um mecanismo que se ajuste de acordo com a variação do eco e os filtros adaptativos possuem essa capacidade.

convergência tem um limite de 50 no máximo 60ms para ser executado [Douskalis 1999]. Esse limite precisa ser respeitado, pois um dos principais problemas na comunicação IP é o atraso do sinal da voz.

4. Implementação

Para a implementação do algoritmo de cancelamento de eco, foi escolhido o ambiente Delphi. Primeiramente gravamos diversas amostras de voz com codificação PCM, a 8 bits e 8kHz. Essas amostras foram utilizadas para que o cancelador pudesse ser testado.

Sobre a amostra de voz utilizada foi acrescentado eco, primeiramente com um atraso fixo e depois fizemos o atraso do eco variar. Como o eco na realidade sofre uma atenuação em relação ao sinal original, essa característica também foi simulada. O objetivo disso é simular uma fala com um sinal mais atrasado e atenuado com relação ao original (eco acústico). Sobre o arquivo gerado foi aplicado o algoritmo de cancelamento de eco que foi desenvolvido. A seguir é mostrado um fluxograma (Figura 5), que mostra como o algoritmo foi feito.

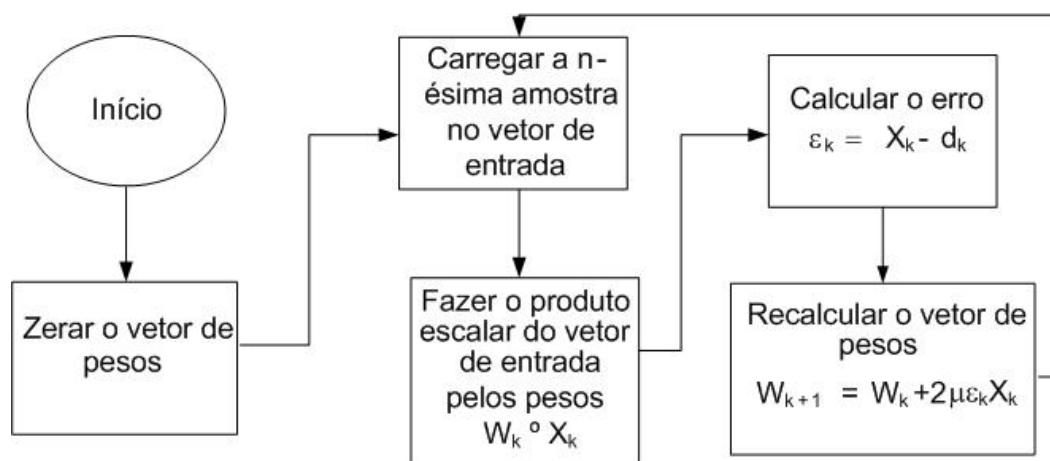


Figura 5 – Fluxograma da implementação do algoritmo de cancelamento de eco

O algoritmo ilustrado pelo fluxograma foi aplicado e após isso foi gerado um novo arquivo. Utilizamos um vetor com 128 pesos e o μ utilizado foi de 0,00007.

5. Testes e Resultados

Após a implementação foi verificado que o processo de cancelamento de eco teve êxito. Com os três arquivos (o original, o com o eco adicionado e com o eco cancelado), pudemos fazer comparações entre os mesmos. Apenas “ouvindo” o som de cada arquivo já foi possível verificar o cancelamento de eco.

Para que pudesse ser uma melhor avaliação dos resultados foi implementado um algoritmo, onde cada amostra de áudio pudesse ser analisada visualmente. Na Figura 6 pode-se notar claramente que o eco foi cancelado.

O primeiro gráfico mostra o arquivo com a amostra original, o segundo mostra a amostra com eco adicionado e o terceiro mostra a amostra após a aplicação do cancelador de eco. O eco adicionado nesse caso tinha 1 db de atenuação com relação a amostra original e 100ms de atraso constante.

Como se pode notar o gráfico com o cancelamento de eco está mais semelhante ao original do que ao do eco. Os gráficos não ficaram iguais, isso se deve ao tempo necessário para que o filtro se adapte.

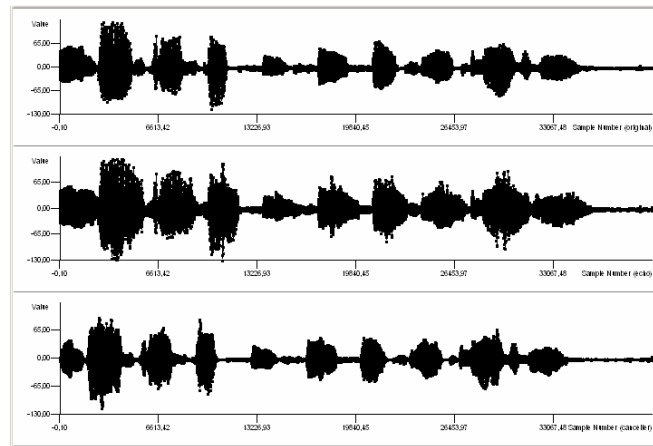


Figura 6 – Gráficos de comparação

O cancelador de eco também foi testado em um ambiente onde o atraso do eco varia. O seu desempenho obteve resultados semelhantes ao de quando o eco era constante. O filtro convergiu corretamente e dessa forma teve a capacidade de cancelar o eco.

6. Conclusões

Após a implementação e os testes que foram feitos, pudemos concluir que o algoritmo LMS é um algoritmo de baixa complexidade computacional. Sua implementação é simples, comparada a de outros algoritmos adaptativos.

Notamos também que o LMS, início do processo de adaptação é um algoritmo lento. Isso porque ele inicia sua adaptação apenas quando o número de amostras analisados pelo filtro é igual ao tamanho do vetor de pesos. O valor do μ foi extremamente importante, pois dependendo do μ utilizado o filtro não consegue convergir corretamente e não funciona.

7. Bibliografia

- B. Widrow, and S. D. Stearns, "Adaptive signal processing", Prentice-Hall, Englewood Cliffs, 1985, pp. 1-364.
- B. Douskalis, "IP Telephony the Integration of Robust VoIP Services", Prentice-Hall, 1999, pp.253-258.
- The International Engineering Consortium, "Echo Cancellation", Web Forum Tutorials, http://www.iec.org/online/tutorials/echo_cancel/glossary.html, acesso 05/07/2004.
- "Echo Analysis for Voice over IP," Version 1, 2001, pp. 1-24.
- P.R.G. Franco, M.C.F. De Castro and F.C.C. De Castro, "Introdução ao Processamento Adaptativo de Sinais Digitais", PUCRS – FENG – DEE – Mestrado em Engenharia Elétrica, Porto Alegre, 2001, pp. 1-19.
- M. Rudson, "Acoustic Echo Cancellation Using Digital Signal Processing", University of Queensland, 2003, pp. 69 – 72.