Cancelamento de Eco com o uso de Filtros Digitais

MATHEUS FARIAS
Universidade Federal de Pernambuco
Departamento de Eletrônica e Sistemas
matheussobreirafarias@gmail.com

RAILTON ROCHA
Universidade Federal de Pernambuco
Departamento de Eletrônica e Sistemas
rocha.railton13@gmail.com

***Resumo*— O presente projeto busca implementar um filtro digital em MATLAB, através de técnicas de processamento digital de sinais, para cancelar o eco em uma gravação de som de voz, bem como analisar parâmetros importantes da implementação.**

I. MOTIVAÇÃO

É comum, em gravações de aúdio para ambientes fechados, a presença forte de eco. Basicamente, o eco pode ser modelado matematicamente, de maneira simplificada, na Equação 1:

$$y[n] = x[n] + \alpha x[n-N] \tag{1}$$

Com x[n] sendo um sinal de voz não corrompido e $\alpha < 1$. Este é um modelo matemático para um eco que é produzido quando um sinal é refletido por uma superfície (uma parede, por exemplo). A gravação de alguém falando em um microfone em uma sala, conterá o sinal de voz que incide diretamente sobre o microfone, bem como o eco correspondente ao sinal que se propagou através da sala e foi refletido pelas paredes, atingindo posteriormente o microfone. Como o eco percorre um caminho mais longo, ele estará retardado no tempo e atenuado ($\alpha x[n-N]$).

Sabe-se que o menor intervalo de tempo entre dois sons percebido pelo ouvido humano é de 0.1 s, sendo assim, utilizando que a velocidade do som v_s é aproximadamente 340 m/s à temperatura ambiente, é possível estimar a distância mínima que uma pessoa deve ter de uma parede para que possa perceber a sua voz e o eco:

$$d_{total} = v_s \Delta t_{min} = 340 \cdot 0.1 = 34 \text{ m}$$

$$d_{total} = d_{ida} + d_{volta}$$

mas,

$$d_{ida} = d_{volta} = d_{parede}$$

então:

$$d_{parede} = 34/2 = 17 \text{ m} \tag{2}$$

Por simplicidade, outras reflexões não são consideradas. No que se segue, considere $N = 1000$ e $\alpha = 0.5$.

II. PROJETO

O projeto todo é feito com os sinais $y[n]$, $y_2[n]$ e $y_3[n]$ obtidos através do arquivo *lineup.mat*

A. Problemas Básicos

Primeiramente, encontra-se a função transferência de 1 facilmente, tomando a transformada Z da equação de diferenças:

$$y[n] = x[n] + \alpha x[n-N]$$

Tomando a transformada Z, tem-se:

$$Y(z) = X(z) + \alpha X(z) z^{-N}$$

$$Y(z) = X(z)[1 + \alpha z^{-N}]$$

E portanto:

$$H_e(z) = Y(z)/X(z) = 1 + \alpha z^{-N} \tag{3}$$

com a resposta ao impulso $h_e[n]$ abaixo:

$$h_e[n] = [\underbrace{1\ 0\ 0\ \dots\ 0\ \alpha}_{N+1}] \tag{4}$$

Tendo a resposta ao impulso $h_e[n]$ que representa a adição do eco ao sinal de entrada, é preciso que se tenha um sistema de cancelamento de eco, para isso, é construido o sistema de cancelamento representado pela Equação 5 abaixo:

$$v[n] + \alpha v[n-N] = y[n] \tag{5}$$

Com $y[n]$ sendo a entrada do sistema e v[n] sua saída com o eco removido. É facil perceber que 1 e 5 definem sistemas inversos, basta observar a multiplicação de suas funções transferência. Primeiramente, obtem-se a função transferência $H_{ce}(z)$ de 5:

$$v[n] + \alpha v[n-N] = y[n]$$

Aplicando a transformada Z:

$$V(z) + \alpha V(z) z^{-N} = Y(z)$$

$$V(z)[1 + \alpha z^{-N}] = Y(z)$$

E portanto:

$$H_{ce}(z) = V(z)/Y(z) = \frac{1}{1 + \alpha z^{-N}} \tag{6}$$

Então, multiplicando 3 e 6:

$$H_{res}(z) = H_{ce}(z) H_e(z) = \left(\frac{1}{1 + \alpha z^{-N}} \right) (1 + \alpha z^{-N}) = 1$$

Com isso, os sistemas são inversos, portanto:

$$V(z)/X(z) = \frac{V(z)}{Y(z)} \cdot \frac{Y(z)}{X(z)} = H_{res}(z) = 1$$

Consequentemente:

$$v[n] = x[n]$$

B. *Problemas Intermediários*

O sistema de cancelamento de eco representado pela Equação 5 possui uma resposta ao impulso de duração infinita (é um filtro IIR). Para encontrar tal resposta, usa-se uma função do MATLAB chamada *filter()*, que recebe 3 entradas: o numerador da função transferência do filtro correspondente, o denominador da função transferência do filtro correspondente, e o vetor que servirá de entrada para o filtro. Para se encontrar a resposta ao impulso do sistema de cancelamento de eco, a entrada d usada será uma aproximação da função impulso para o tamanho de 4000 zeros:

$$d = [\underbrace{1\ 0\ 0\ \ldots\ 0\ 0}_{4000+1}]$$

Sendo assim, aplicando na função *filter()* com o numerador sendo 1 e o denominador a:

$$a = [\underbrace{1\ 0\ 0\ \ldots\ 0\ \alpha}_{N+1}]$$

Tem-se:

```
a = zeros(1,N+1);
a(1) = 1;
a(N+1) = alpha
d = [1 zeros(1,4000)]
hce = filter(1,a,d)
```

Agora, para fazer a filtragem propriamente dita, deve-se usar a função *filter()* com os parâmetros da Equação 5, porém a entrada agora será o som que busca-se remover o eco, o som $y[n]$:

```
z = filter(1,a,y)

plot(z)
sound(z,8192)
```

Com o resultado obtido, é possível entender que o som falado no audio é a expressão line-up, que era praticamente inaudível inicialmente. Segue abaixo o plot do sinal antes e depois da filtragem:

Note que, como os sistemas $H_e(z)$ e $H_{ce}(z)$ são inversos, a convolução de suas respostas ao impulso deve retornar a função impulso, uma vez que a multiplicação de suas funções transferência resulta no valor 1. Para corroborar isso, faz-se a convolução entre $h_e[n]$ e $h_{ce}[n]$ através da função do MATLAB, *conv*(), que recebe como entrada os dois sinais que serão convoluidos. Em aspectos práticos, perceba que o numerador a é exatamente igual ao sinal $h_e[n]$. Sendo assim:

```
he = a

hs= conv(he,hce)
```

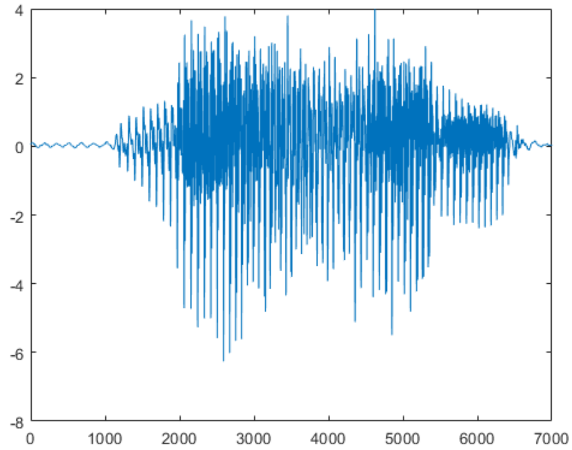

Fig. 1. O som $y[n]$

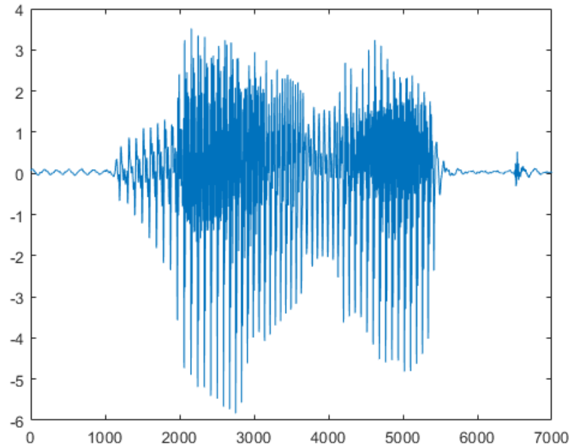

Fig. 2. O som $z[n]$ resultante da filtragem em $y[n]$

Porém, ao plotar $h_s[n]$, é possível ver que não retorna-se uma função impulso:

```
stem(hs)
```

Utilizando a função *find* do MATLAB, é possível ver mais facilmente que o sinal $h_s[n]$ possui valor zero em todos os pontos, menos no ponto zero (que é o valor 1) e no ultimo ponto, o 5001, onde vale 0.0313.

```
find(hs~=0)
```

Tal valor no último ponto está associado ao fato de se fazer uma aproximação para a resposta ao impulso do filtro IIR. Pela própria definição, tal resposta ao impulso deve ser infinita, porém, na prática, ter uma resposta infinita é inviável para questões de hardware. Sendo assim, no próprio software MATLAB tem que se realizar uma aproximação para o filtro, limitando sua resposta e consequentemente, criando erros. Nesse caso, utilizou-se uma aproximação de 4000 zeros para a função impulso.

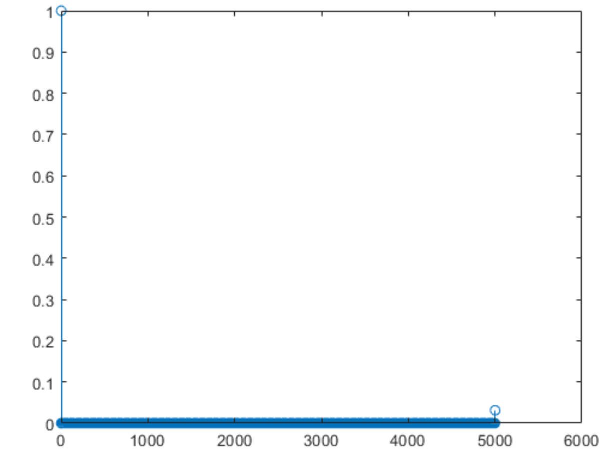

Fig. 3. O sinal h_s resultante da convolução

Por fim, plota-se o diagrama de polos e zeros do sistema $H_e(z)$ e de $H_{ce}(z)$, é claro que se espera que os diagramas difiram no fato de os zeros de $H_e(z)$ serem os polos de $H_{ce}(z)$, e os polos de $H_e(z)$ serem os zeros de $H_{ce}(z)$

```
    zplane(a, 1)
    zplane(1, a)
```

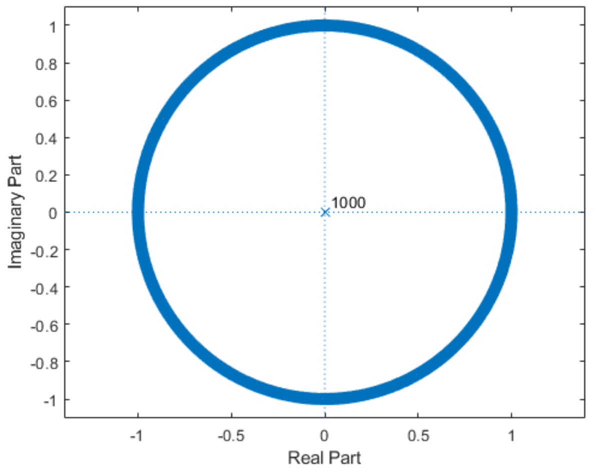

Fig. 4. O diagrama de polos e zeros de $H_e(z)$

C. Problemas Avançados

Naturalmente, não se sabe os parâmetros N e α de um dado sinal, e por isso surge a importância de se determinar um método para estimação das variáveis dado um sinal $y[n]$.

Para tal estimativa, considere a operação autocorrelação, definida na Equação 7

$$R_y[n] = \sum_{m=-\infty}^{+\infty} y[m]^* y[n+m] \tag{7}$$

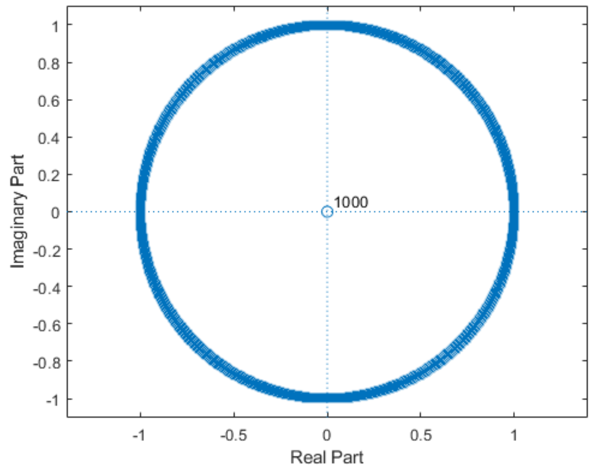

Fig. 5. O diagrama de polos e zeros de $H_{ce}(z)$

No caso considerado, tem-se $y[n]$ real e seguindo a Equação 1, logo:

$$y[n] = x[n] + \alpha x[n-N]$$

$$R_y[n] = \sum_{m=-\infty}^{+\infty} y[m]y[n+m]$$

$$R_y[n] = \sum_{m=-\infty}^{+\infty} (x[m] + \alpha x[m-N])(x[n+m] + \alpha x[n+m-N])$$

Realizando a distributiva, e percebendo que $\sum_{m=-\infty}^{+\infty} x[m]x[m+n] = \sum_{m=-\infty}^{+\infty} x[m-N]x[m-N+n] = R_x[n]$, onde $R_x[n]$ é a autocorrelação do sinal $x[n]$, tem-se:

$$R_y[n] = (1+\alpha^2)R_x[n] + \alpha R_x[n-N] + \alpha R_x[n+N] \tag{8}$$

Note que pela Equação 8, se espera que haja um pico em $n = 0$ e picos em $n = \pm N$, portanto, uma estimativa boa para N é observar pelo gráfico a distância entre o pico central e o próximo pico positivo.

Ainda é possível estimar o valor de α considerando os valores de $R_y[n]$ para $n = kN$, $k \in \mathbb{Z}$. Aplicando $n = 0$, N e $2N$ na Equação 8 e observando que $R_x[n]$ é uma função par, é possível obter o sistema:

$$\begin{cases} R_y[0] = (1+\alpha^2)R_x[0] + \cancel{2\alpha R_x[N]} & \text{(9.a)} \\ R_y[N] = \cancel{(1+\alpha^2)R_x[N]} + \alpha R_x[0] + \cancel{\alpha R_x[2N]} & \text{(9.b)} \end{cases}$$

Por $x[n]$ ser um sinal de voz pouco autocorrelacionado (sinal de fala), é possível considerar que sua autocorrelação, $R_x[n]$, terá amplitudes muito baixas para valores de n distantes de 0. Dessa forma, $R_x[N]$ tem uma amplitude muito menor do que $R_x[0]$ e portanto desconsidera-se os termos $R_x[N]$ e $R_x[2N]$ do sistema. Com isso, através de algumas manipulações no sistema, chega-se na Equação 10:

$$\alpha^2 - \frac{R_y[0]}{R_y[N]}\alpha + 1 = 0 \tag{10}$$

Para o sinal $y[n]$ considerado anteriormente nos Problemas Intermediários, $R_y[0] = 9966.9$ e $R_y[N] = 3957.0$, portanto através da função *roots()* do MATLAB, é possível encontrar facilmente a solução da equação quadrática, obtendo o valor de $\alpha = 0.4938$ (é importante lembrar que $\alpha < 1$!).

```
p = [1  -2.5188  1]
r = roots(p)
```

Existe uma função no MATLAB chamada *autocorr*() que recebe como parâmetros o sinal que será autocorrelacionado e a quantidade de amostras que será calculada, e retorna a função de autocorrelação normalizada, segue a resposta para o sinal $y[n]$:

```
corre = autocorr(y,2000)
plot(corre)
```

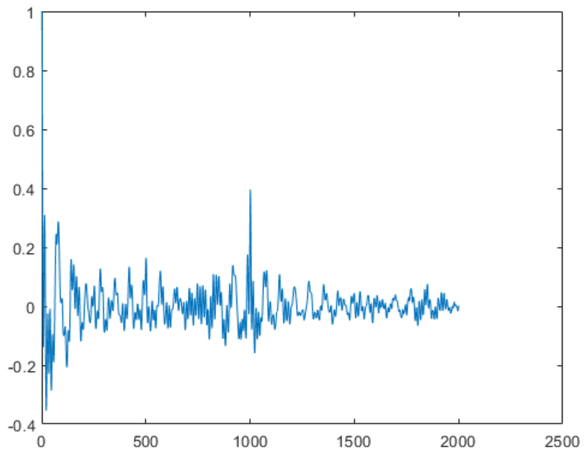

Fig. 6. Função de autocorrelação de $y[n]$

Como esperado, a distância entre os 2 picos, neste caso, é de 1000 amostras, que é justamente o N em questão.

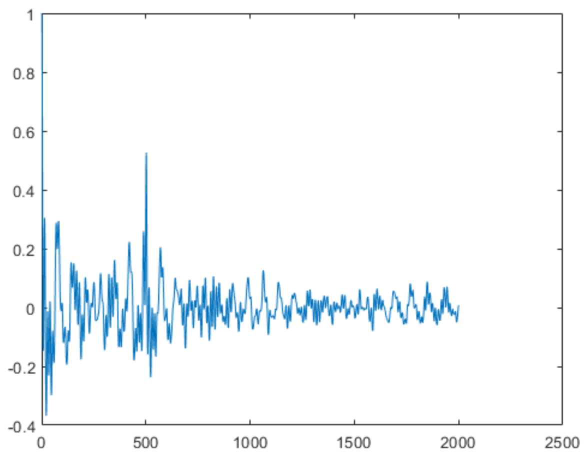

Fig. 7. Função de autocorrelação de $y_2[n]$

Para o sinal $y_2[n]$, mostrado na Figura 7 a distância entre os picos é de 500 amostras, ou seja, $N' = 500$ neste caso. Observando a Equação 10, tem-se que a validade da mesma é para valores de $R_{y2}[0]/R_{y2}[N] > 2$, no caso, tem-se $R_{y2}[0] = 13163$ e $R_{y2}[N] = 6937.7$, ou seja, $R_{y2}[0]/R_{y2}[N] = 1.8973 < 2$, que resulta em uma solução complexa, $\alpha' = 0.949 \pm 0.316i$. A explicação para este resultado estranho é vista logo quando se escuta o sinal $y_2[n]$, não se percebe eco, e por isso não é de interesse a elaboração do filtro.

Suponha agora que o som $x[n]$ sofra a atuação de duas fontes de eco, e então o sinal $y_3[n]$ será como representado na Equação 11

$$y_3[n] = x[n] + \alpha_1 x[n - N_1] + \alpha_2 x[n - N_2] \tag{11}$$

Ao tomar-se a autocorrelação deste sinal, e depois de algumas manipulações, pode-se chegar na relação dada pela Equação 12:

$$\begin{aligned} R_{y3}[n] = {} & R_x[n] + \alpha_1^2 R_x[n] + \\ & + \alpha_2^2 R_x[n] + \alpha_1 R_x[n - N_1] + \alpha_2 R_x[n - N_2] + \\ & + \alpha_1 R_x[n + N_1] + \alpha_2 R_x[n + N_2] + \\ & + \alpha_1 \alpha_2 R_x[n + N_2 - N_1] + \alpha_1 \alpha_2 R_x[n + N_1 - N_2] \end{aligned} \tag{12}$$

Note que nesse caso, ocorre um fenômeno de batimento entre os ecos, e por isso surge o pico da diferença entre os dois ecos. Assume-se, sem perda de generalidade, que $N_2 > N_1$, ou seja, o pico associado ao N_2 seria o pico mais distante do centro, enquanto o pico associado ao N_1, seria o pico mais próximo ao centro, conforme visto na Figura 8

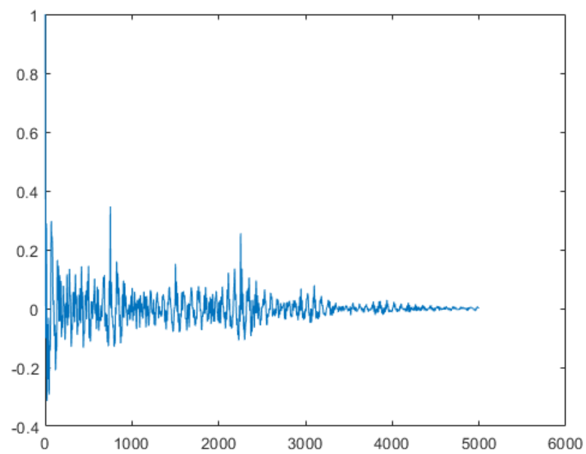

Fig. 8. Função de autocorrelação de $y_3[n]$

Com isso, tem-se $N_2 = 2253$, $N_1 = 752$ e $N_2 - N_1 = 1501$. Para estimar α_1 e α_2, considera-se a mesma aproximação feita para encontrar a Equação 10[1], obtendo o sistema abaixo:

$$\begin{cases} R_{y3}[0] = R_x[0](1+\alpha_1^2+\alpha_2^2) & \text{(13.a)} \\ R_{y3}[N_1] = \alpha_1 R_x[0] & \text{(13.b)} \\ R_{y3}[N_2] = \alpha_2 R_x[0] & \text{(13.c)} \end{cases}$$

E consequentemente as Equações 14 e 15:

$$\alpha_1^2\left(\frac{R_{y3}[N_2]^2}{R_{y3}[N_1]} + R_{y3}[N_1]\right) + \alpha_1 R_{y3}[0] + R_{y3}[N_1] = 0 \quad (14)$$

$$\alpha_2 = \frac{R_{y3}[N_2]}{R_x[0]} \quad (15)$$

Com

$$R_x[0] = \frac{R_{y3}[N_1]}{\alpha_1}$$

```
p = [7533.45320  -14008.69500
    4883.67835]
r = roots(p)
```

Para o caso em particular, tem-se $R_{y3}[N_1] = 4883.7$, $R_{y3}[N_2] = 3597.3$ e $R_{y3}[0] = 14008.7$, logo, usando a função *roots*, encontra-se $R_x[0] = 10507.3$, $\alpha_1 = 0.4648$ e $\alpha_2 = 0.3423$.

III. Conclusão

Com o resultado obtido no desenrolar do projeto, é possível perceber a importância do uso da filtragem de cancelamento de eco no tratamento de sinais de audio, uma vez que o som inicial do sinal $y[n]$ era praticamente impossível de perceber do que se tratava, e após o procedimento de filtragem, foi possível entender com muito mais clareza.

Na vida real, não se sabe os parâmetros α e N associados aos sinais de interesse, e portanto o procedimento discorrido na seção de *Problemas Avançados* se mostra bastante interessante para que se torne possível a filtragem em aplicações práticas.

Um fato curioso é que a estratégia de encontrar a defasagem N entre os sinais também pode ser usada para identificar a localização de eventos sonoros! Pois uma vez que sabe-se as defasagens dos sons recebidos por microfones dois a dois, e sabendo a velocidade do som, é possível traçar raios de circunferência com as distância obtidas, e disso, bastaria encontrar a intersecção entre as circunferências para determinar a localização do evento[2].

Bibliografia

[1] OPPENHEIM, A. V., SCHAFER, R. W., BUCK, J. R *Discrete-time signal processing*. Upper Saddle River, N.J., Prentice Hall.

[2] http://www.innovatefpga.com/cgi-bin/innovate/teams.pl?Id=AS026

[1] Aqui, vale ressaltar que a aproximação so é possível para o pico de batimento $N_2 - N_1$ resultando em um valor grande, ou seja, a aproximação é válida para $N_2 \gg N_1$.

[2] Para saber mais sobre esta aplicação, consultar o link do projeto iOwlT: Sound Geolocalization System[2], elaborado por Matheus Farias, Davi Moreno e Gabriel Firmo na bibliografia do presente projeto.