

handson08_Matlab

July 13, 2017

1 Handson 08 - Matlab

1.1 Ruído AWGN

O ruído AWGN (Additive White Gaussian Noise) é um modelo de ruído usado para simulações de efeito em muitos processos aleatórios que ocorrem na natureza.

- **Additive** : Porque é adicionado a qualquer ruído que possa ser intrínseco ao sinal;
- **** White **** : Refere-se à possuir uma potência uniforme em toda a banda de frequência do sinal;
- **Gaussian** : Possui uma distribuição normal;
-

1.2 Noise : Ruído.

Usa-se a distribuição normal para simular a maioria dos fenômenos naturais, por convergir a um determinado valor. Para gerar um ruído AWGN deve-se seguir as seguintes etapas:

1) Conhecida a SNR, mede-se a potência do vetor $x(t)$ pela seguinte equação:

$$E_s = \frac{1}{N} \sum_{i=0}^{N-1} |x(i)|^2$$

Para $L = \text{length}(x)$ Comprimento de $x(t)$;

2) Converte uma dada SNR em dB para escala linear, gerando um vetor ruído usando a equação abaixo:

- Real:

$$Rudo_{real} = randn(1, L) \cdot \sqrt{\left(\frac{E_s}{SNR_{linear}}\right)}$$

- Complexo:

$$Rudo_{complexo} = [randn(1, L) + j \cdot randn(1, L)] \cdot \sqrt{\left(\frac{E_s}{2 \cdot SNR_{linear}}\right)}$$

3) Realiza a soma do sinal $x(t)$ com o ruído

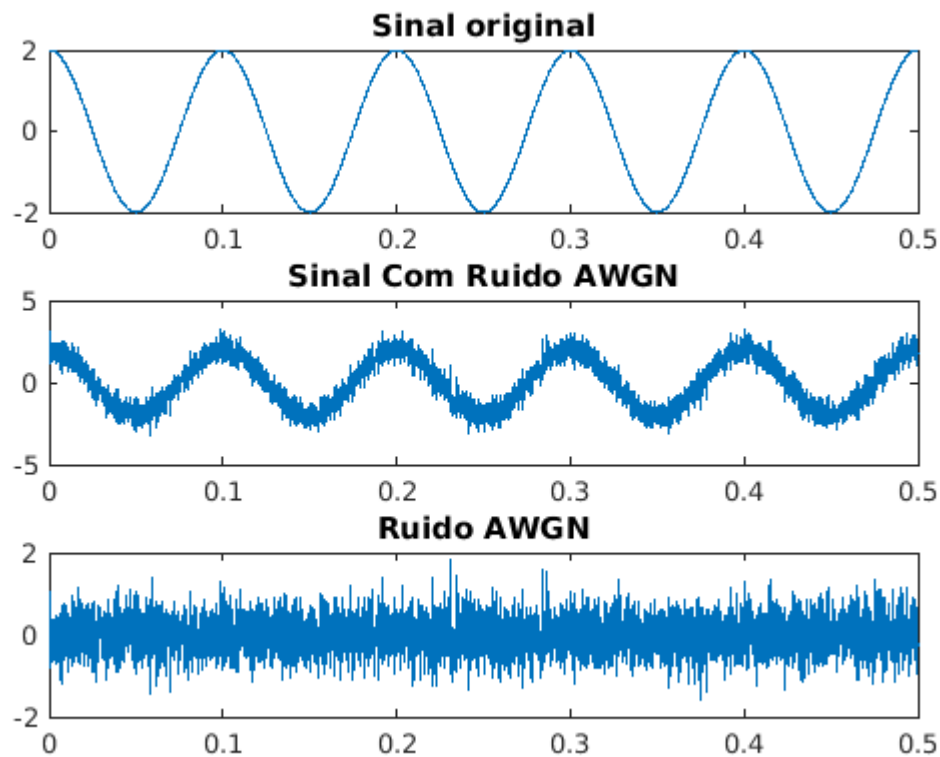
$$Y(t) = x(t) + \text{Rudo}(t)$$

O script abaixo mostra uma implementação de ruído para um sinal qualquer $x(t)$:

```
In [2]: % AWGN_Real.m
        clc;clear all;close all;
        %% Parâmetros
        SNR_dB= 10; % Determina o valor da
        t=0:0.0001:0.5; % Eixo do tempo
        x=2*cos(2*pi*10*t); % Sinal qualquer x(t)

        %% Montagem do vetor Ruído Real
        L=length(x); % Calcula o comprimento
        Es= sum(abs(x).^2)/L; % Calcula a potência do
        SNR= 10^(SNR_dB/10); % Calcula a SNR linear
        D=Es/SNR; % Calcula a densidade espectral
        noiseSigma = sqrt(D); % Derivação padrão para
        n = noiseSigma* randn(1,L); % Ruído real calculado
        y =x+n; % Sinal Ruidoso

        %% Plotting
        subplot(3,1,1)
        plot(t,x);
        title('Sinal original')
        subplot(3,1,2)
        plot(t,y);
        title('Sinal Com Ruído AWGN')
        subplot(3,1,3)
        plot(t,n);
        title('Ruído AWGN')
```



Simularemos também para um sinal complexo, gerando também um ruído complexo segundo a equação dada acima:

```
In [3]: % AWGN_Complexo.m
        clc;clear all;close all;
        %% Parâmetros
        SNR_dB= 10;                                     % Determina o valor da
        t=0:0.0001:0.5;                                 % Eixo do tempo
        x=2*cos(2*pi*10*t)+i*0.2*cos(2*pi*10*t);        % Sinal qualquer x(t)

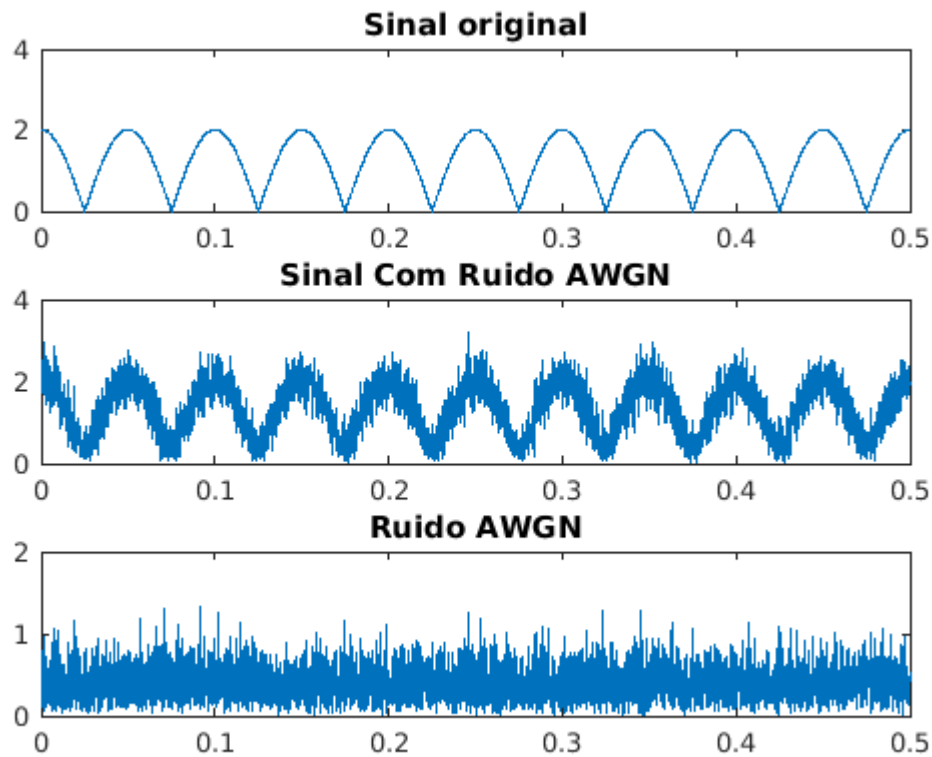
        %% Montagem do vetor Ruído Complexo
        L=length(x);                                    % Calcula o comprimento
        Es= sum(abs(x).^2)/L;                             % Calcula a potência do
        SNR= 10^(SNR_dB/10);                             % Calcula a SNR linear
        D=Es/SNR;                                         % Calcula a densidade de
        noiseSigma= sqrt(D/2);                           % Derivação padrão para
        n = noiseSigma*(randn(1,L)+i* randn(1,L));       % Ruído complexo calculado
        y =x+n;                                           % Sinal Ruidoso

        %% Plotting
        subplot(3,1,1)
        plot(t,abs(x));
```

```

title('Sinal original')
subplot(3,1,2)
plot(t,abs(y));
title('Sinal Com Ruído AWGN')
subplot(3,1,3)
plot(t,abs(n));
title('Ruído AWGN')

```



Desafio: Estimar a SNR (dB) de um sinal $x = \cos(2\pi f_m t)$ real e um complexo do arquivo DesafioR_08.mat e DesafioC_08.mat, respectivamente a partir do sinal recebido $y(t)$, da amplitude do sinal A_m e sua frequência f_m .

Obs: O sinal complexo é dado como a soma de dois cossenos de amplitudes diferentes, sendo um complexo e outro real.

- a) Cálculo do valor da SNR.
- b) Justificar o porquê de ao colocar um ruído de $\text{snr} = 10$, na medição é comum ter valores de snr próximos ao original.