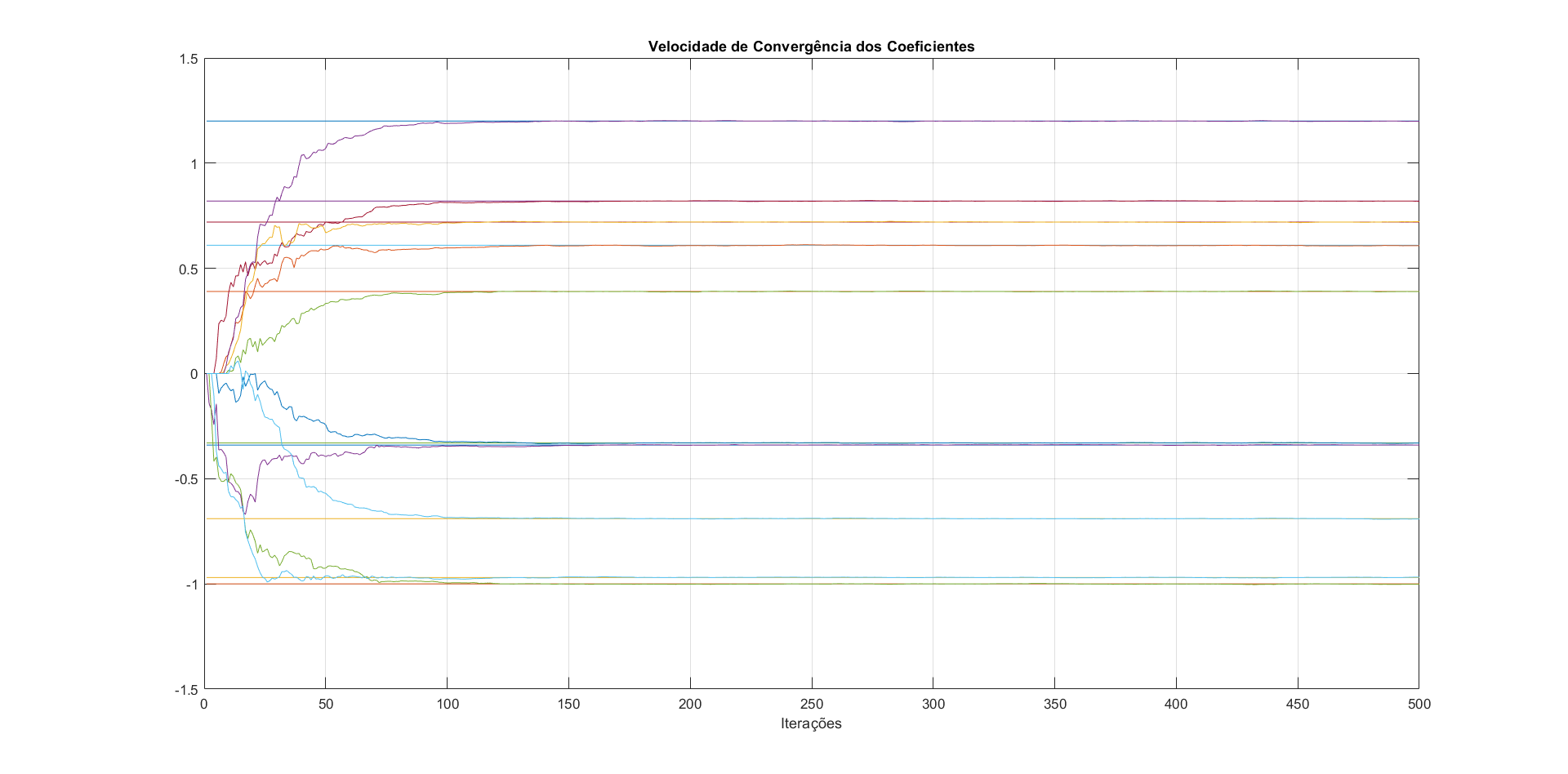
Os *Scripts* de MATLAB usados para alcançar os resultados mostrados abaixo estão nos Anexos.

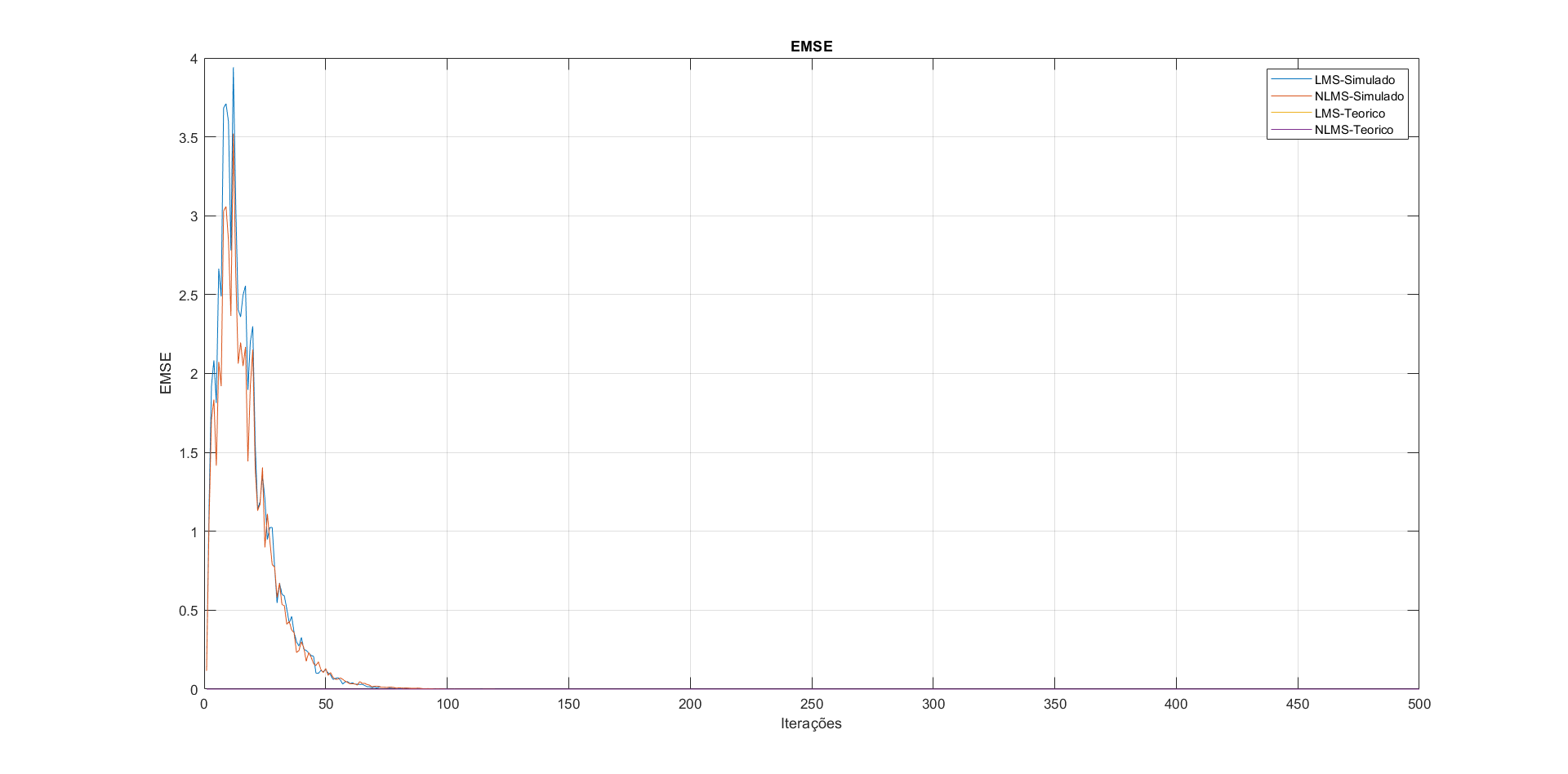
**a)**

**a1.**

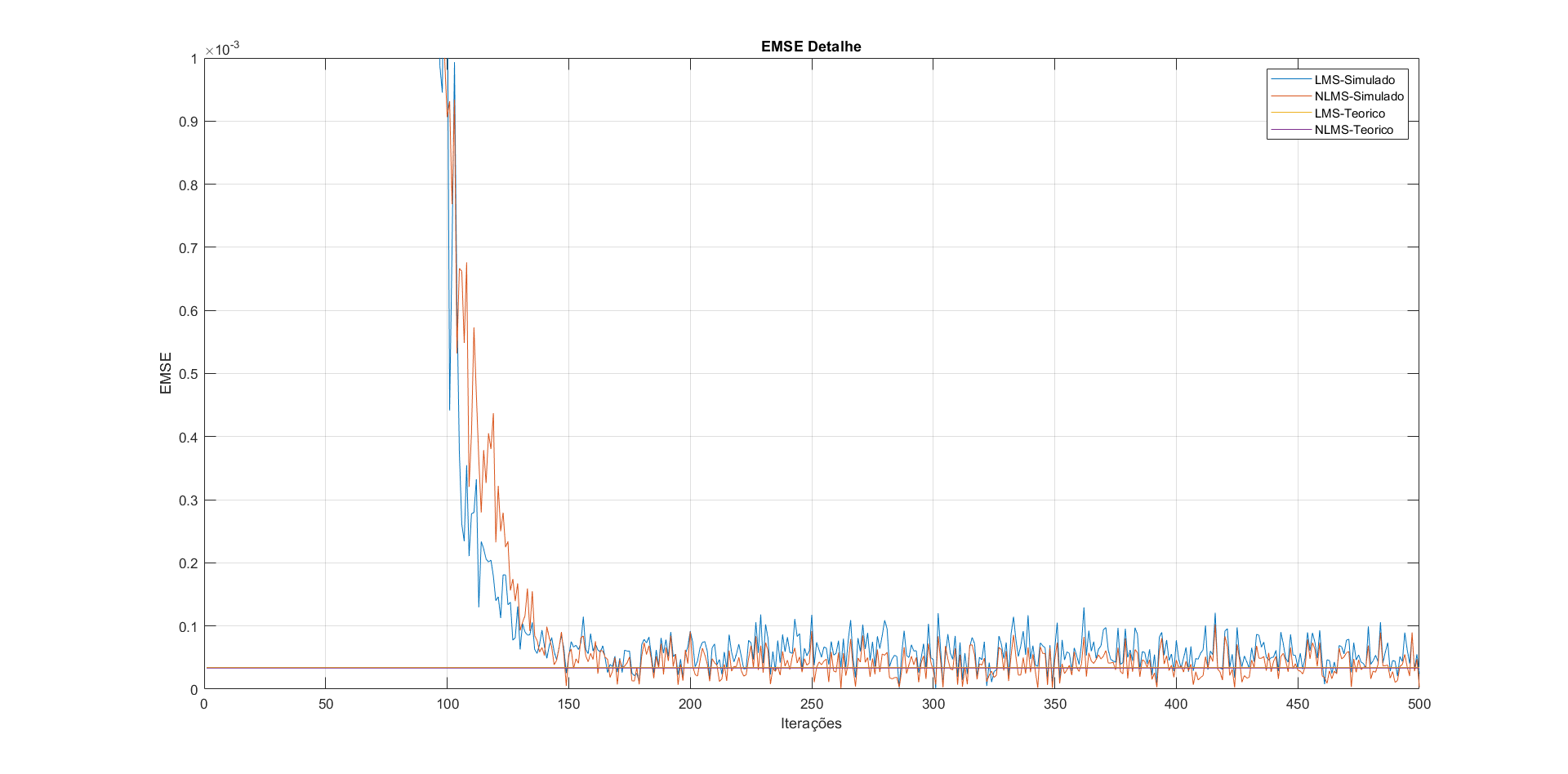


*Fiigura 1 – Comparação da média dos coeficientes adaptados com algoritmo NLMS com coeficientes ótimos.*

**a4.**



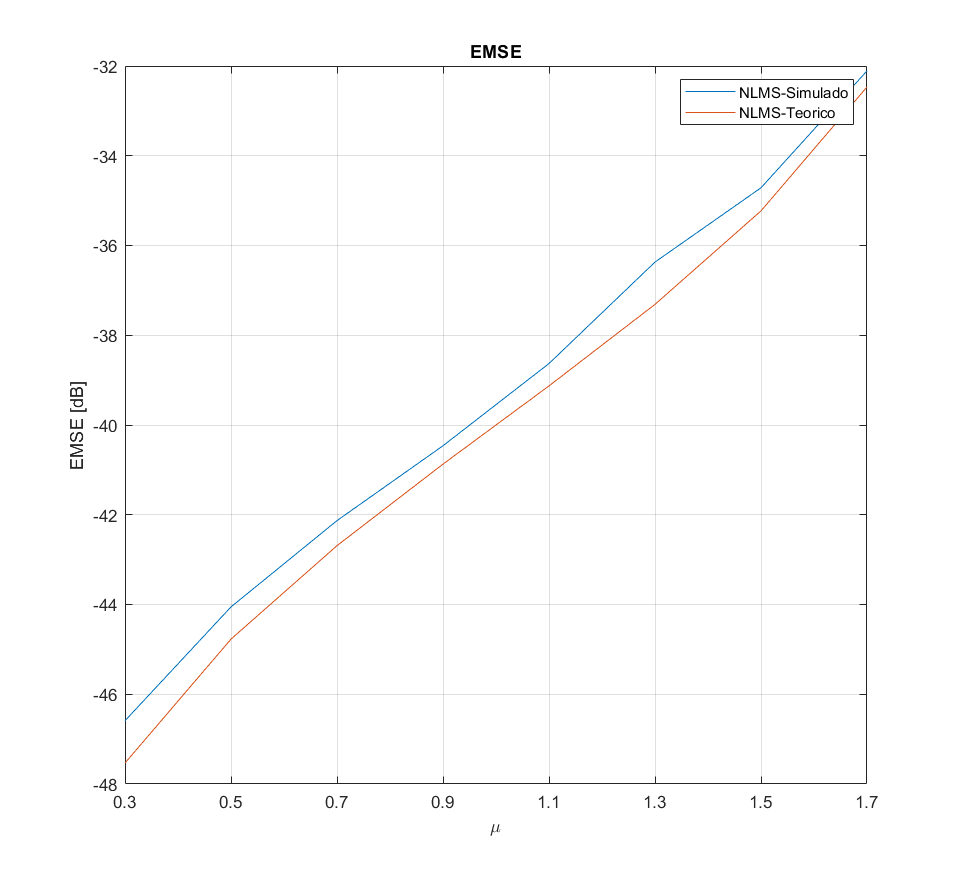
*F**igura 2 – Curvas de EMSE para o LMS com μ=0,0667 e para o NLMS com =0,5.*

**

*Figura 3 – Detalhes das Curvas apresentadas na Figura 2.*

**a5.** O NLMS continua convergindo em todos os valores assumidos de, entretanto o LMS diverge quando conforme demostrado no item a3, sendo

**b)**O desajuste aumenta com o aumento do passo de adaptação do algoritmo. Conforme apresentado na Figura 4.

*Figura 4 – Curvas de EMSE teórico e simulado, em dB, para o algoritmo NLMS.*

**Anexo A – Script de MATLAB para os itens a)**

%PTC 5890 - 2019

%Prof: Magno T. M. Silva e Maria D. Miranda

%Aluno: Stéfano Albino Vilela Rezende (Ouvinte)

%P3 - Parte Computacional

clear all

close all

clc

n = 501

H = [-0.34; -1; -0.97; 0.82; -0.33; 0.61; 0.72; 1.2; 0.39; -0.69]

M = 10

N = 500

mutil =0.5 %Usado no NLMS

mu=2\*mutil/(10\*(2-mutil)) %Usado no LMS

delta = 10^-5

sigmav = 10^-4

WNmean=zeros(n,10)

WLmean=zeros(n,10)

EMSEL\_Sim=zeros(n-1,1)

EMSEN\_Sim=zeros(n-1,1)

for rpt=1:100

u = randn(1,n) %sinal de entrada

x=filter(H,[1],u)

v=sqrt(sigmav)\*randn(1,length(x)) %ruído de medida

d = x+v %sinal desejado

[yn,en,WN] = NLMS (u,d,M,mutil,N,delta)

[y,el,WL] = LMS (u,d,M,mu,N)

WNmean =(WN+WNmean)/2 %Cálculo da média dos coeficientes do filtro NLMS

WLmean =(WL+WLmean)/2 %Cálculo da média dos coeficientes do filtro LMS

EMSEL\_Sim(:,rpt)= el.^2-sigmav

EMSEN\_Sim(:,rpt)= en.^2-sigmav

end

%a1.

figure(1)

plot(H'.\*ones(500,10))

hold on

plot(WNmean)

xlim([0,500])

grid;title('Velocidade de Convergência dos Coeficientes')

xlabel('Iterações')

%a2. e a3. resolvidas a mão.

%a4.

[R,p,Wo] = Wiener(u,d,M)

trR = trace(R)

EMSEL\_Teo = mu\*sigmav \*trR/2 %Cálculo do EMSE teórico para o LMS

EMSEN\_Teo = mutil\*sigmav /(2-mutil) %Cálculo do EMSE teórico para o NLMS

figure(2)

plot(abs(mean(EMSEL\_Sim,2)))

hold on

plot(abs(mean(EMSEN\_Sim,2)))

plot(EMSEL\_Teo.\*ones(500,1))

plot(EMSEN\_Teo.\*ones(500,1))

grid;title('EMSE');legend('LMS-Simulado','NLMS-Simulado','LMS-Teorico','NLMS-Teorico')

xlabel('Iterações');ylabel('EMSE')

figure(3)

plot(abs(mean(EMSEL\_Sim,2)))

hold on

plot(abs(mean(EMSEN\_Sim,2)))

plot(EMSEL\_Teo.\*ones(500,1))

plot(EMSEN\_Teo.\*ones(500,1))

grid;title('EMSE Detalhe');legend('LMS-Simulado','NLMS-Simulado','LMS-Teorico','NLMS-Teorico')

xlabel('Iterações');ylabel('EMSE');ylim([0 1\*10^-3])

**Anexo B – Script de MATLAB para o b)**

%PTC 5890 - 2019

%Prof: Magno T. M. Silva e Maria D. Miranda

%Aluno: Stéfano Albino Vilela Rezende (Ouvinte)

%P3 - Parte Computacional item b

clear all

close all

clc

n = 501

H = [-0.34; -1; -0.97; 0.82; -0.33; 0.61; 0.72; 1.2; 0.39; -0.69]

M = 10

N = 500

mutil = [0.3 0.5 0.7 0.9 1.1 1.3 1.5 1.7]

h=waitbar(0,'Computando...');

for j=1:length(mutil)

delta = 10^-5

sigmav = 10^-4

for rpt=1:100

u = randn(1,n) %sinal de entrada

x=filter(H,[1],u)

v=sqrt(sigmav)\*randn(1,n) %ruído de medida

d = x+v

[yn,en,WN] = NLMS (u,d,M,mutil(j),N,delta)

EMSE\_Sim(:,rpt)= en.^2-sigmav

end

EMSE\_Teo(j) = mutil(j)\*sigmav /(2-mutil(j))

EMSEmean(j) = mean(mean(EMSE\_Sim(0.9\*N:N,:)))

waitbar(j/length(mutil))

end

close(h)

figure(1)

plot(mutil,10\*log10(EMSEmean))

hold on

plot (mutil,10\*log10(EMSE\_Teo))

grid;title('EMSE');legend('NLMS-Simulado','NLMS-Teorico')

xlim([0.3 1.7]);xlabel('\mu');ylabel('EMSE[dB]')

**Anexo C – Função de MATLAB para a solução de Wiener**

function [R,p,Wo] = Wiener(u,d,M)

r=xcorr (u,M-1,'biased')

ru=r(M:end);

R=toeplitz(ru)

rdu=xcorr(d,u,M-1,'biased')

p=rdu(M:end)

Wo=R\p

**Anexo E – Função de MATLAB para o algoritmo NLMS**

function [y,e,W] = NLMS (u,d,M,mu,N,delta)

uM = zeros(M,1)

y = zeros(N,1)

e = zeros(N,1)

W = zeros(N+1,M)

for n=1:N

uM=[u(n);uM(1:M-1)];

y(n)=W(n,:)\*uM;

e(n)= d(n)-y(n);

W(n+1,:) = W(n,:)+mu/(norm(uM)^2+delta)\*e(n)\*uM';

end

end

**Anexo D – Função de MATLAB para o algoritmo LMS**

function [y,e,W] = LMS (u,d,M,mu,N)

uM = zeros(M,1)

y = zeros(N,1)

e = zeros(N,1)

W = zeros(N+1,M)

for n=1:N

uM=[u(n);uM(1:M-1)];

y(n)=W(n,:)\*uM;

e(n)= d(n)-y(n);

W(n+1,:) = W(n,:)+mu\*e(n)\*uM';

end

end