**d)** Seguem abaixo os resultados da norma das diferenças entre a matriz estimada e a teórica e entre os vetores estimados e teóricos:

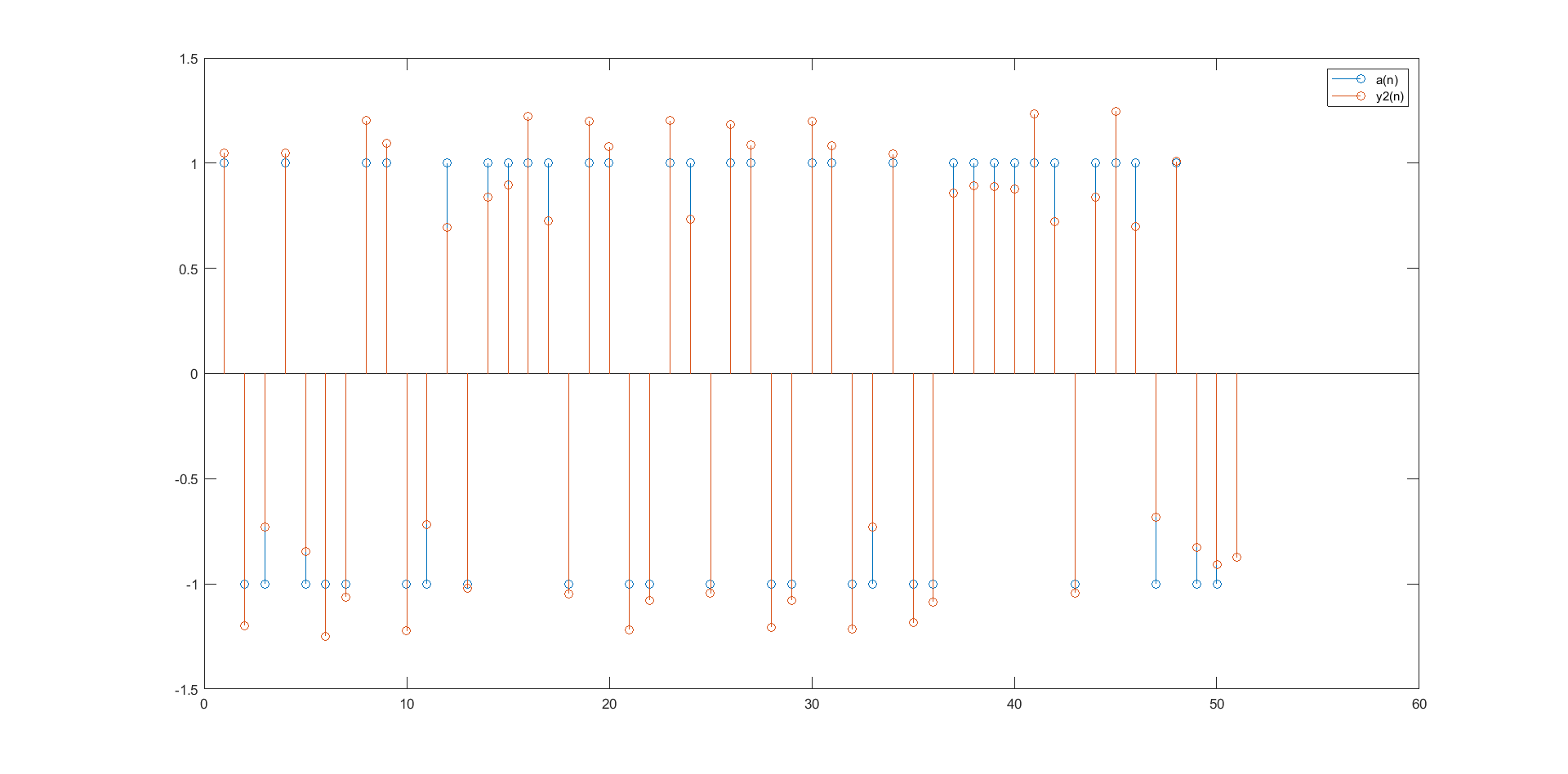
A norma espectral de uma matriz hermetiana retorna o maior autovalor dessa matriz.

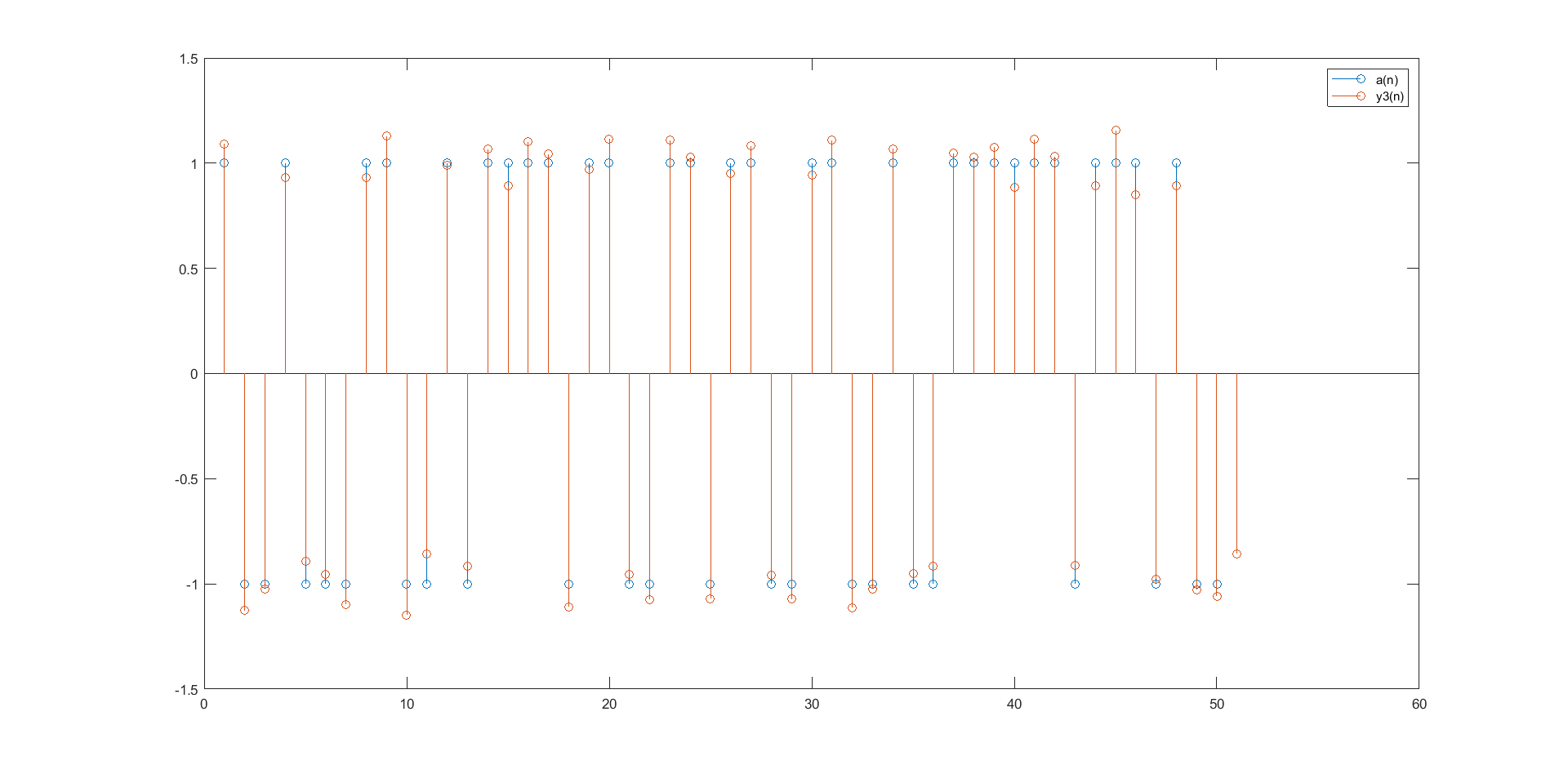
**e)** Sim, a soma dos autovalores é igual ao traço de matriz estimaque por sua vez é bem próximo do valor do traço da matriz teórica **R**.

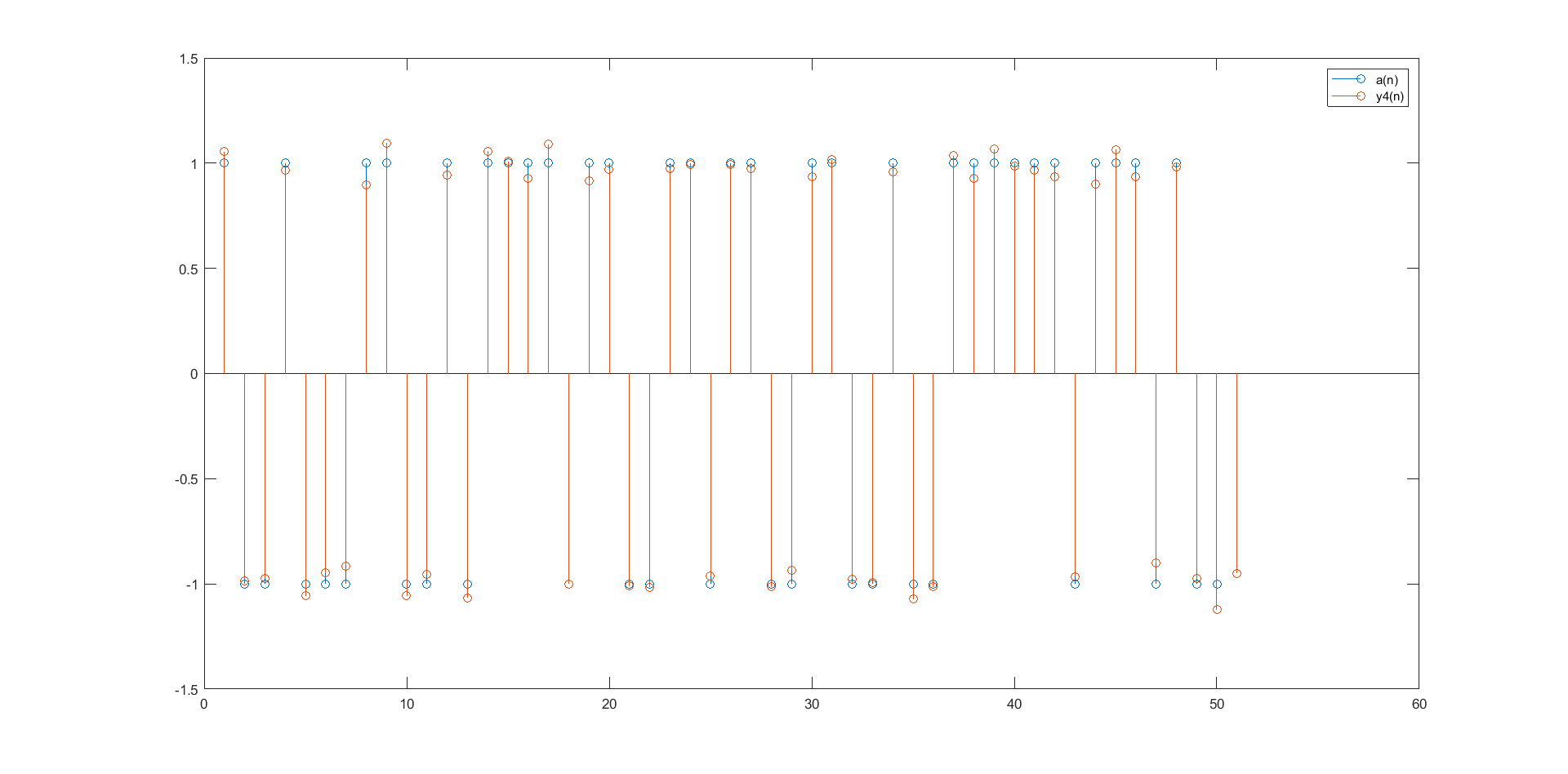
Considerando os resultados acima obtidos podemos considerar a matriz bem dimensionada.

**f)** Abaixo estão os vetores de coeficientes ótimos para cada um dos atrasos nd = 2, 3, 4 e 7.

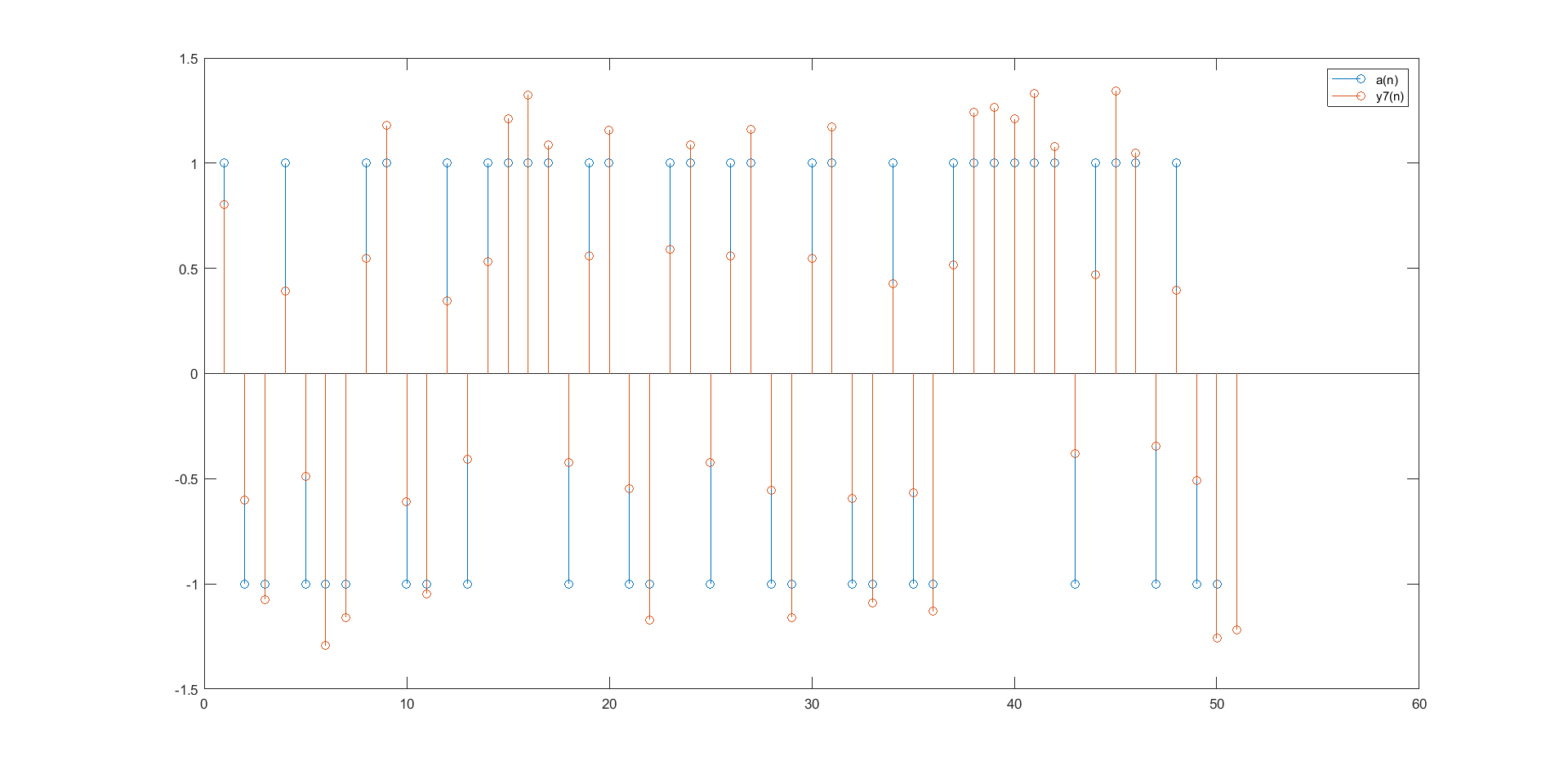
**f)** As figuras a seguir apresentam 50 amostras da sequência de entrada *a(n)* sobreposta pelas sequências na saída do filtro *y(n)* para os atrasos nd = 2, 3, 4 e 7. A título de comparação nd amostras foram removidas do início da sequência *ynd(n).*

*Figura 1 – Sequência de entrada a(n) sobreposta pela sequência de saída do filtro y(n) para o atraso nd = 2.*

*Figura 2 – Sequência de entrada a(n) sobreposta pela sequência de saída do filtro y(n) para o atraso nd = 3.*

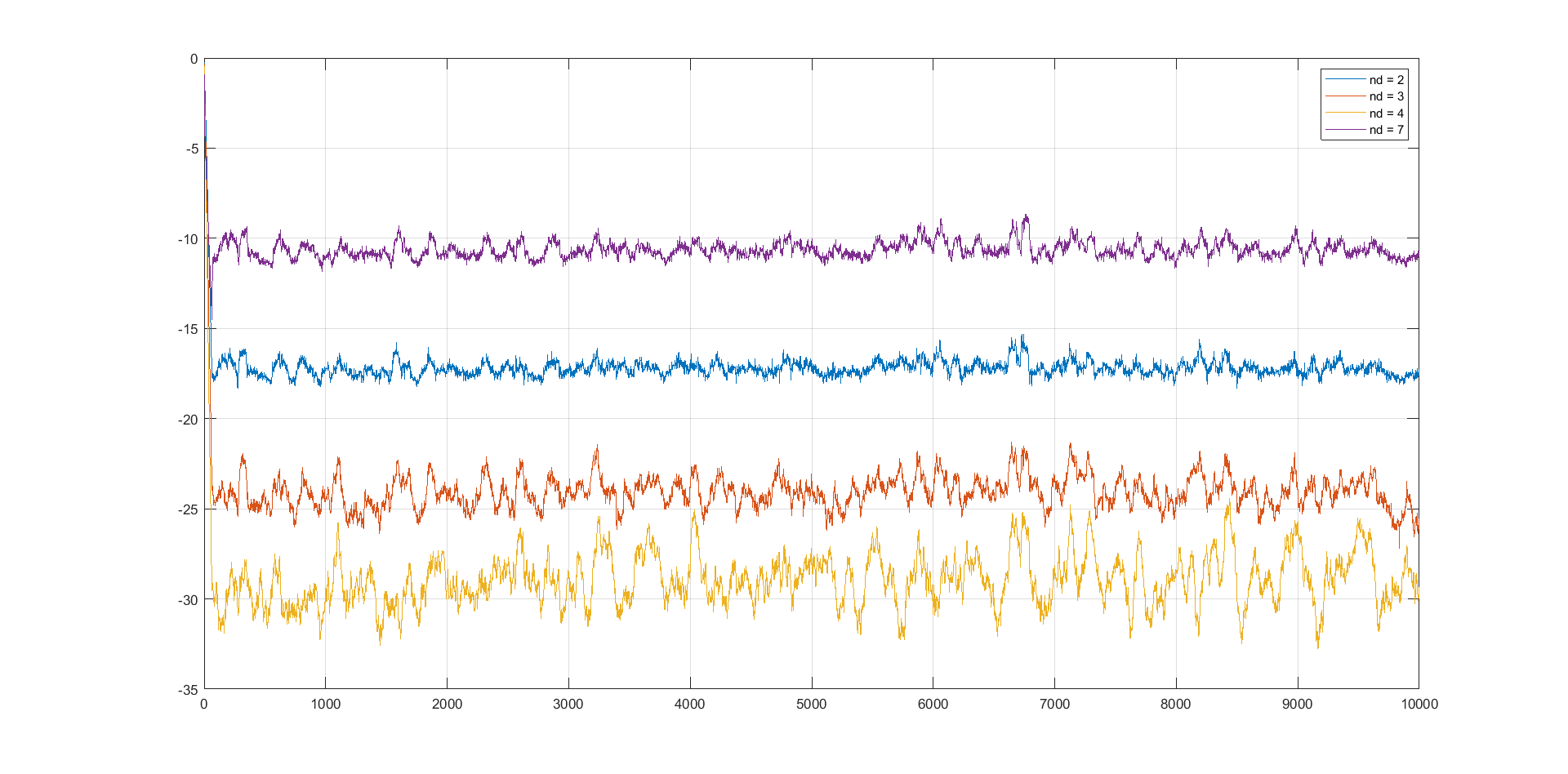


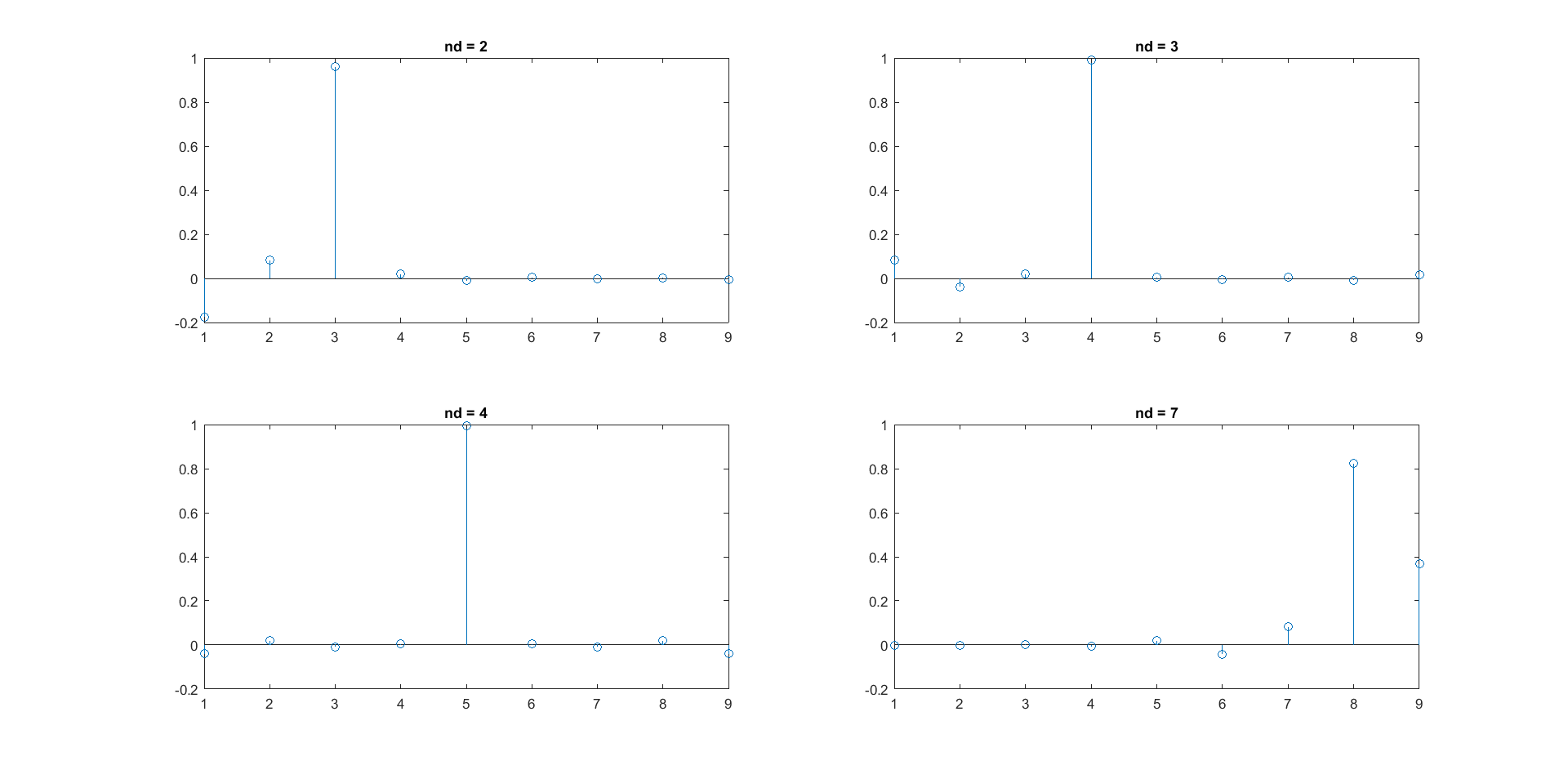
*Figura 3 – Sequência de entrada a(n) sobreposta pela sequência de saída do filtro y(n) para o atraso nd = 4.*



*Figura 4 – Sequência de entrada a(n) sobreposta pela sequência de saída do filtro y(n) para o atraso nd = 7.*

**h)** Abaixo segue os gráficos do erro quadrático em dB e da resposta combinada:

*Figura 5 – Erro Quadrático em dB para cada atraso.*

*Figura 6 – Resposta combinada para cada atraso.*

Um filtro de média móvel de 64 coeficientes foi usado para melhor visualização do gráfico do erro quadrático, pois esse filtro normaliza os 64 elementos mais próximos, eliminando ruídos brancos do sinal.

Além dessas análises foi calculado também a interferência intersimbólica residual para cada atraso, tendo como resultados:

ISI2 = -13,7907 dB

ISI3 = -20,1711 dB

ISI4 = -23,8757 dB

ISI7 = -6,7073 dB

Portanto, baseado nos resultados obtidos, o atraso que tem melhor desempenho é **nd = 4.**

# Anexo A – Scripts MATLAB

%PTC 5890 - 2019

%Prof: Magno T. M. Silva

%Aluno: Stéfano Albino Vilela Rezende (Ouvinte)

%P1 - Parte Computacional

%itens a) e b) foram feitos a mão e serão entregues ao professor.

clear

clc

%c)

%Atribuindo os valores aos coeficientes da resposta do canal H(z).

h0=0.3887

h2=0.3887

h1=1

sgm=0.001

%Cálculo dos elementos da matriz R e seu traço

ru0 = h0^2 + h1^2 + h2^2 + sgm

ru1 = h1\*h0 + h1\*h2

ru2 = h2\*h0

ru3 = 0

ru4 = 0

ru5 = 0

ru6 = 0

r = [ru0 ru1 ru2 ru3 ru4 ru5 ru6]

Rteo = toeplitz(r)

trRteo= trace(Rteo)

%Cálculo dos elementos dos vetores p

rud0 = h2

rud1 = h1

rud2 = h0

p2teo = [rud0; rud1; rud2; 0; 0; 0; 0]

p3teo = [0; rud0; rud1; rud2; 0; 0; 0]

p4teo = [0; 0; rud0; rud1; rud2; 0; 0]

p7teo = [0; 0; 0; 0; 0; rud0; rud1]

%d)

%Geração do sinal a(n)

M = 7n = 10000a = 2\*randi ([0 1], n, 1)-1

%Geração do sinal d(n) para cada atraso nd

nd2 = [0; 0; 1]

nd3 = [0; 0; 0; 1]

nd4 = [0; 0; 0; 0; 1]

nd7 = [0; 0; 0; 0; 0; 0; 0; 1]

d2 = filter(nd2,[1],a)

d3 = filter(nd3,[1],a)

d4 = filter(nd4,[1],a)

d7 = filter(nd7,[1],a)

%Resposta do Canal

h = [0.3887; 1; 0.3887]

a1 = filter(h,[1],a)

noise = 0.001\*randn(size(a,1),1)

u = a1 + noise

%Cálculo da Matriz de Correlação (R) Estimada e

%Vetores de Correlação Cruzada Estimado(p)

%A função Wiener retorna também os vetores coeficientes ótimos Wo para o

%item f)

[R,p2,Wo2] = Wiener(u,d2,M)

[R,p3,Wo3] = Wiener(u,d3,M)

[R,p4,Wo4] = Wiener(u,d4,M)

[R,p7,Wo7] = Wiener(u,d7,M)

%Cálculo da norma da diferença entre a matriz R teórica e estimada

%e dos vetores p teóricos e estimados.

Rres= Rteo-R

p2res= p2teo-p2

p3res= p3teo-p3

p4res= p4teo-p4

p7res= p7teo-p7

Rnorm = norm(Rres)

p2norm = norm(p2res)

p3norm = norm(p3res)

p4norm = norm(p4res)

p7norm = norm(p7res)

%e)

[Q,A]= eig(R)

Rdec = Q\*A\*Q'

trA = trace (A)

trR = trace (R)

%f)

%Coeficientes ótimos Wo calculados pela função Wiener no item d)

%g)

%Filtrando os sinais u(n) considerando os quatro vetores de coeficientes

%ótimos Wo.

y2 = filter(Wo2,[1],u)

y3 = filter(Wo3,[1],u)

y4 = filter(Wo4,[1],u)

y7 = filter(Wo7,[1],u)

%Gráficos das Sequência de entrada (a(n)) x Saídas do Filtro (y(n))

figure(1)

stem (a(1:50))

hold on

stem (y2(3:53))

legend ('a(n)','y2(n)')

figure(2)

stem (a(1:50))

hold on

stem (y3(4:54))

legend ('a(n)','y3(n)')

figure(3)

stem (a(1:50))

hold on

stem (y4(5:55))

legend ('a(n)','y4(n)')

figure(4)

stem (a(1:50))

hold on

stem (y7(8:58))

legend ('a(n)','y7(n)')

%h)

%Cálculo do erro quadrático em dB

e2=10\*log10((d2-y2).^2)

e3=10\*log10((d3-y3).^2)

e4=10\*log10((d4-y4).^2)

e7=10\*log10((d7-y7).^2)

%Filtragem dos erros por um filtro de média móvel, como sugerido no

%exercício

Hmm = 1/64.\*ones(1,64)

fe2 = filter(Hmm,[1],e2)

fe3 = filter(Hmm,[1],e3)

fe4 = filter(Hmm,[1],e4)

fe7 = filter(Hmm,[1],e7)

figure(5)

plot(fe2)

hold on

grid on

plot(fe3)

plot(fe4)

plot(fe7)

legend('nd = 2','nd = 3','nd = 4','nd = 7')

%Cálculo da ISI para nd = 2

s2=conv(Wo2,h)

s2o=max(s2)

Som2 = 0

for i=1:length(s2)

Som2=s2(i)^2+Som2

end

ISI2=10\*log10(Som2/(s2o^2)-1)

%Cálculo da ISI para nd = 3

s3=conv(Wo3,h)

s3o=max(s3)

Som3 = 0

for i=1:length(s3)

Som3=s3(i)^2+Som3

end

ISI3=10\*log10(Som3/(s3o^2)-1)

%Cálculo da ISI para nd = 4

s4=conv(Wo4,h)

s4o=max(s4)

Som4 = 0

for i=1:length(s4)

Som4=s4(i)^2+Som4

end

ISI4=10\*log10(Som4/(s4o^2)-1)

%Cálculo da ISI para nd = 7

s7=conv(Wo7,h)

s7o=max(s7)

Som7 = 0

for i=1:length(s7)

Som7=s7(i)^2+Som7

end

ISI7=10\*log10(Som7/(s7o^2)-1)

%Gráficos das respostas combinada para cada atraso

figure (5)

subplot(221); stem (s2);title('nd = 2')

subplot(222); stem (s3);title('nd = 3')

subplot(223); stem (s4);title('nd = 4')

subplot(224); stem (s7);title('nd = 7')

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