bc

**MEMO Number** : CMPE320\_S21\_PROJ5\_CODE

**DATE: 5/17/2021**

**TO: EFC LaBerge**

**FROM: Nem Negash**

**SUBJECT: MATLAB Code**

# MATLAB COde

function proj5()

%S21 CMPE320 Project 5 Skeleton for Students

% EFCL 4/28/2021

close all

clear

% Set the number of layers or number of trials

N=1000; % you might have to adjust this for memory. Make it as big as you can up to about 10,000

% Larger than 10,000 won't have much effect.

% Set the length of the time array this is somewhat arbitrary, longer is

% better

Nt=1000;

Ntd2 =(Nt+mod(Nt,2))/2; %We'll need this to access the "middle" of output of xcorr

% Initialize Rxx to zeros; columns are times, each row is a different trial

% Initial x to zeros; columns are times, each row is a different trial

% x will store the sample functions

Rxx = zeros(N,Nt);

x = zeros(N,Nt);

% Use a loop to generate the N different random sample functions

% each sample function is 1 x Nt array from randn N(0,1)

for k = 1:N

% Generate sample function and store in k-th row of x

x(k,:)= randn(1,Nt);

% Temporarily store the output of xcorr( )/Nt using the k-th row of x as both inputs

junk = xcorr(x(k,:),x(k,:))/Nt; %Matlab cross-correlation creates 2\*Nt-1 points

Rxx(k,:) = junk(Ntd2+1:Ntd2+Nt);% just save the middle Nt;

end

R\_XX = mean(Rxx); % "expected value" over Nrows of Rxx (down the columns)

R\_XX0 = max(R\_XX); % fine the max of Rxx

x\_mean = mean(x);

x\_var = var(R\_XX);

%New Figure

% time array from 0 to Nt

t = [0:Nt-1];

% tau array from [-Ndt/2+1:Ntd/2]

tau = [-Ntd2+1:Ntd2];

% create and title 4 subplots

% 1) t vs x (all the functions)

% 2) t vs mean(x)

% 3) tau vs ensemble Rxx (all the functions)

% 4) tau vs mean of Rxx computed above.

figure(1)

subplot(4,1,1)

plot(t, x);

xlabel('t(time)');

ylabel('x');

title(['10000 random variables vs 1000 time units']);

subplot(4,1,2)

plot(t, x\_mean);

xlabel('t(time)');

ylabel('mean(x)');

title(['Average of 10000 random variables over 1000 time units']);

subplot(4,1,3)

plot(tau, Rxx);

xlabel('tau');

ylabel('R\_{xx}');

title(['Cross correlation of y vs tau']);

subplot(4,1,4)

plot(tau, R\_XX);

xlabel('tau');

ylabel('mean(R\_{xx})');

title(['Average cross correlation of y vs tau']);

% Now do the entire thing three more times using a sliding window filter

% Set the array of FIR filter lengths

L =[10 20 40]; %

y\_var = [0,0,0];

% Loop on the filter lengths

for i=1:3

% Set the current filter length

thisLength = L(i);

tau1 = [-Ntd2+1:Ntd2+thisLength];

% Set the coefficients for this FIR filter for MATLAB function filter (trust me!)

b=ones(1,thisLength)/thisLength; % L point sliding window

a=1; %See MATLAB routine filter(b,a,x)

% Initialize Ryy and y, as you did with Rxx and x

Ryy = zeros(N,Nt+thisLength); % more points to accommodate transient

y = zeros(N,Nt);

%Loop on the sample functions as we did with Rxx

for k = 1:N

%Generate xin as the iid Gaussian, as above, but this time with

% Nt+thisLength columns (extra columns)

xin = randn(1,Nt+thisLength); % iid Gaussian variance 1 mean zero

% create a temporary output for the filter output

ytemp = filter(b,1,xin); % filter with the sliding window

% Save into the k-th row of y, but only save the LL+1:end columns of

% ytemp. This remove the filter transient from beginning of the filter

% output

y(k,:) = ytemp(thisLength+1:end);

% Create the temporary output of xcorr using the k-th row of y for both

% inputs; then scale by Nt as before

junk = xcorr(y(k,:),y(k,:))/Nt;

% Store this output in k-th row of Ryy save the middle Nt samples as

% before

Ryy(k,:) = junk(Ntd2+1:Ntd2+Nt+thisLength);% just save the middle Nt, as before;

end % loop on sample functions

R\_YY = mean(Ryy);% create the mean down the columns, as before

R\_YY0 = max(R\_YY) ; % find the max R\_YY, as before

y\_mean = mean(y);

y\_var = var(R\_YY);

% repeat the four previous plots, using y and mean autocorrelation

% make sure to use a new figure each time.

% You should title the plots so you can tell them apart.

figure

subplot(4,1,1)

ylim([-1 1]);

plot(t, y);

xlabel('t(time)');

ylabel('y');

title(['1000 random variables vs 1000 time units for L = ',num2str(thisLength)]);

subplot(4,1,2)

ylim([-.01 .01]);

plot(t, y\_mean);

xlabel('t(time)');

ylabel('mean(y)');

title(['Average of 1000 random variables over 1000 time units for L = ',num2str(thisLength)]);

subplot(4,1,3)

plot(tau1, Ryy);

xlabel('tau');

ylabel('R\_{yy}');

title(['Cross correlation of y vs tau for L = ',num2str(thisLength)]);

subplot(4,1,4)

plot(tau1, R\_YY);

xlabel('tau');

ylabel('mean(R\_{yy})');

title(['Average cross correlation of y vs tau for L = ',num2str(thisLength)]);

disp(['The variance reduction factor for L = ',num2str(thisLength)]);

g = (R\_XX0./R\_YY0)

%disp(g);

end

% display the ratio of the peak autocorrelations RXX/RYY

end %Loop on the filter length