```
%S21 CMPE320 Project 5 Skeleton for Students
%  EFCL 4/28/2021
%  Updates 4/28/2022 (!)

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.   Make this an odd value, so there is an index. in the middle of
% the array. Regardless of the length you choose, make this value different
 the N

Nt=1001;
Ntd2 =(Nt+mod(Nt,2))/2; %We'll need this to access the "middle" of output of
 xcorr


% N is trials, Nt is times

% Initialize Rxx to zeros;  columns are times, each row is a different trial
Rxx = zeros(N,Nt);

% Initial x to zeros; columns are times, each row is a different trial
%   x  will store the sample functions
x = zeros(N, Nt);

% MATLAB hint, initializing in this way is faster than appending a row to x
% in each iteration!

% Use a loop to generate the N different random sample functions
%     each sample function is 1 x Nt array from N(0,1)

for k = 1:N

    % Generate sample function using randn and store in k-th row of x
    x(k,:)= randn(1, Nt);

    % Each row of Rxx is one sample function of the (time) autocorrelation
 function.
    % Since these are linear combinations of random values, each row is
    % also a random function of time.

    % 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 function creates
 2*Nt-1 points, Nt is a normalizing value
    Rxx(k,:) = junk(Ntd2+1:Ntd2+Nt);% just save the middle Nt points, this
 minimizes "edge effects" ;
```

```
end;

R_XX  = mean(Rxx(:,:)); % "expected value" over Nrows of Rxx (down the
 columns). This will be 1 x Nt
R_XX0 = max(R_XX); % find the max of R_XX. This is a single number

%New Figure

% time array from 0 to Nt; this is somewhat arbitrary
t = [0:Nt-1];

%  create  tau array from [-Ntd2+1:Ntd2-1]
tau = [-Ntd2 + 1:Ntd2 - 1];

% create and title 4 subplots
%  1) t vs x  (all the sample functions of the random process x)
%  2) t vs mean(x) the (sample) mean of the sample function as an estimate
%  of the expected value
%  3) tau vs ensemble Rxx (all the time autocorrelation functions)
%  4) tau vs mean of Rxx computed above; an estimate of the expected value

figure();
subplot(2,2,1);
plot(t, x);
title('Trials of Random Variable x');
grid on;
ylabel("Value of x");
xlabel("Value of t");

subplot(2,2,2);
plot(t, mean(x));
title('Trials of Expected Value of x');
grid on;
ylabel("Value of E[x]");
xlabel("Value of t");

subplot(2,2,3);
plot(tau, Rxx);
title('Autocorrelation for different \tau');
grid on;
ylabel("Value of Rxx");
xlabel("Value of \tau");

subplot(2,2,4);
plot(tau, R_XX);
title('Expected Value of Autocorrelation');
grid on;
ylabel("Value of E[Rxx]");
xlabel("Value of \tau");

sgtitle('Outputs for Random Variable x');

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

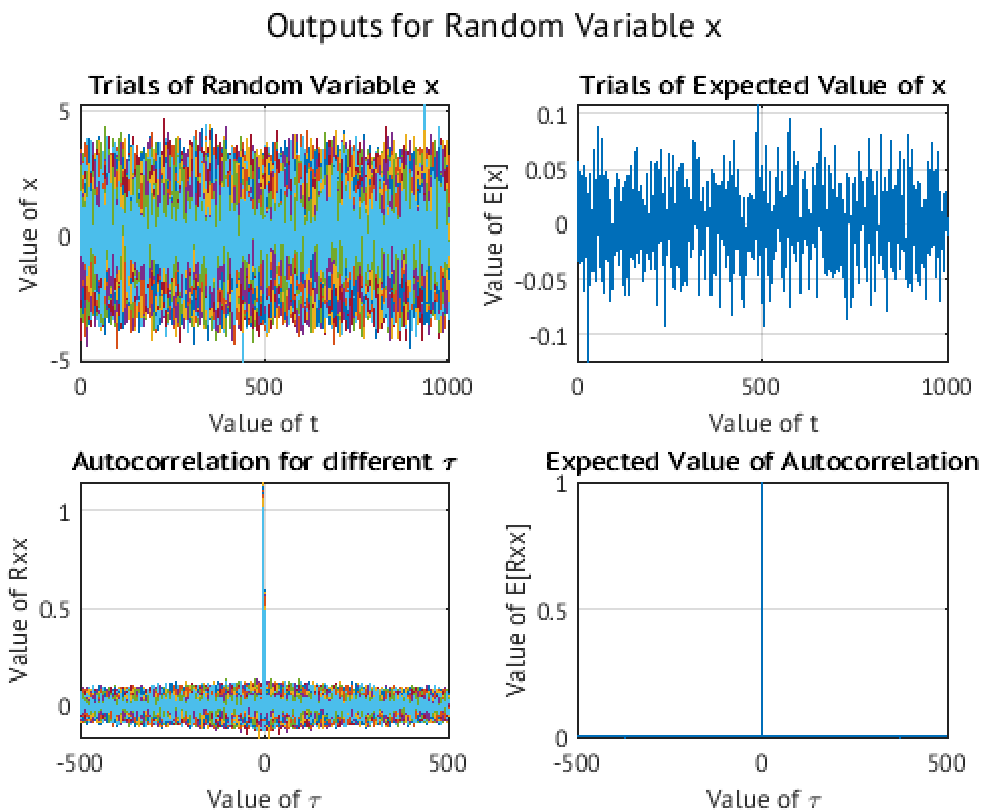
```
% Set the array of FIR filter lengths
L = [10 25 50]; %

disp(R_XX0);
% Loop on  the filter lengths
for i = 1:length(L)
    % Set the current filter length
    thisLength = L(i);

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

    b=ones(1,thisLength)/thisLength; % thisLength-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); % 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 (trust me!)

        % Save into the k-th row of y, but only save the (thisLength+1:end)
 columns of
        % ytemp.  This remove the filter transient from beginning of the
 filter
        % output and eliminates its effects on our results
        y(k,:) = ytemp(thisLength+1:Nt + thisLength);

        % 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
        % Only save the middle Nt samples as before

        Ryy(k,:) = junk(Ntd2+1:Ntd2+Nt);% 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

    % for each filter length, repeat the four previous plots, using y and mean
 autocorrelation
```

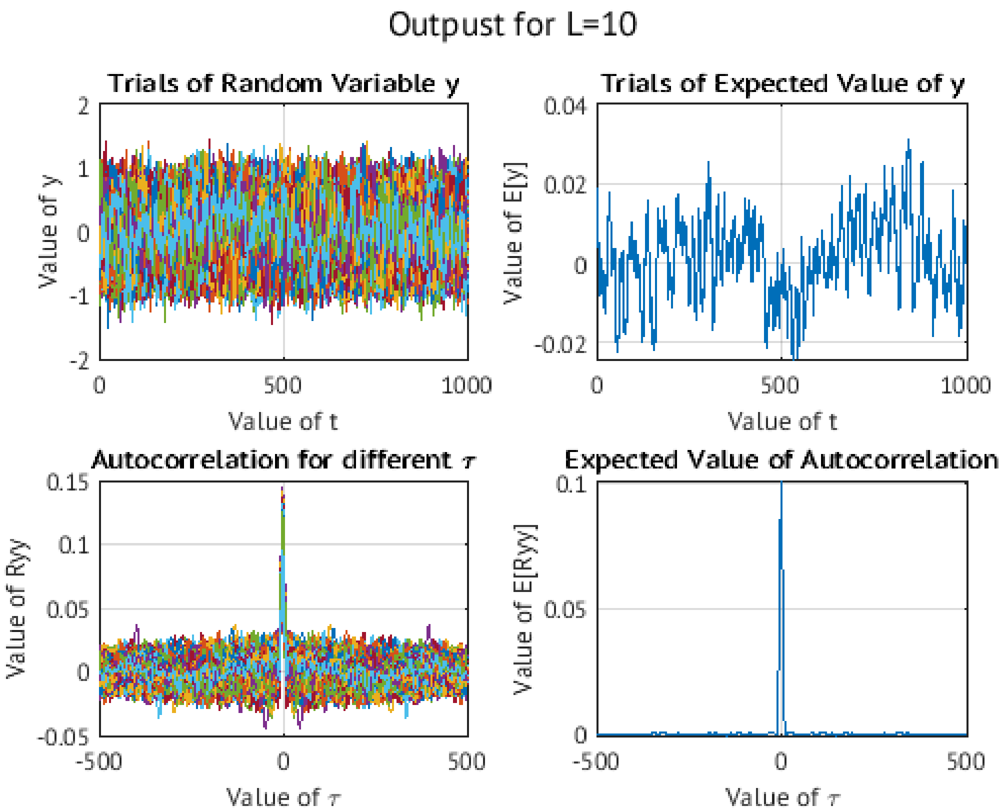
```
    % make sure to use a new figure each time.
    % You should title the plots so you can tell them apart.

    % display the ratio of the peak autocorrelations RXX_0/RYY_0
    disp(R_YY0);
    disp(R_XX0 / R_YY0);

    figure();
    subplot(2,2,1);
    plot(t, y);
    title('Trials of Random Variable y');
    grid on;
    ylabel("Value of y");
    xlabel("Value of t");

    subplot(2,2,2);
    plot(t, mean(y));
    title('Trials of Expected Value of y');
    grid on;
    ylabel("Value of E[y]");
    xlabel("Value of t");

    subplot(2,2,3);
    plot(tau, Ryy);
    title('Autocorrelation for different \tau');
    grid on;
    ylabel("Value of Ryy");
    xlabel("Value of \tau");

    subplot(2,2,4);
    plot(tau, R_YY);
    title('Expected Value of Autocorrelation');
    ylabel("Value of E[Ryy]");
    xlabel("Value of \tau");
    grid on;

    sgtitle(['Outpust for L=', num2str(L(i))]);

end; %Loop on the filter lengths
```

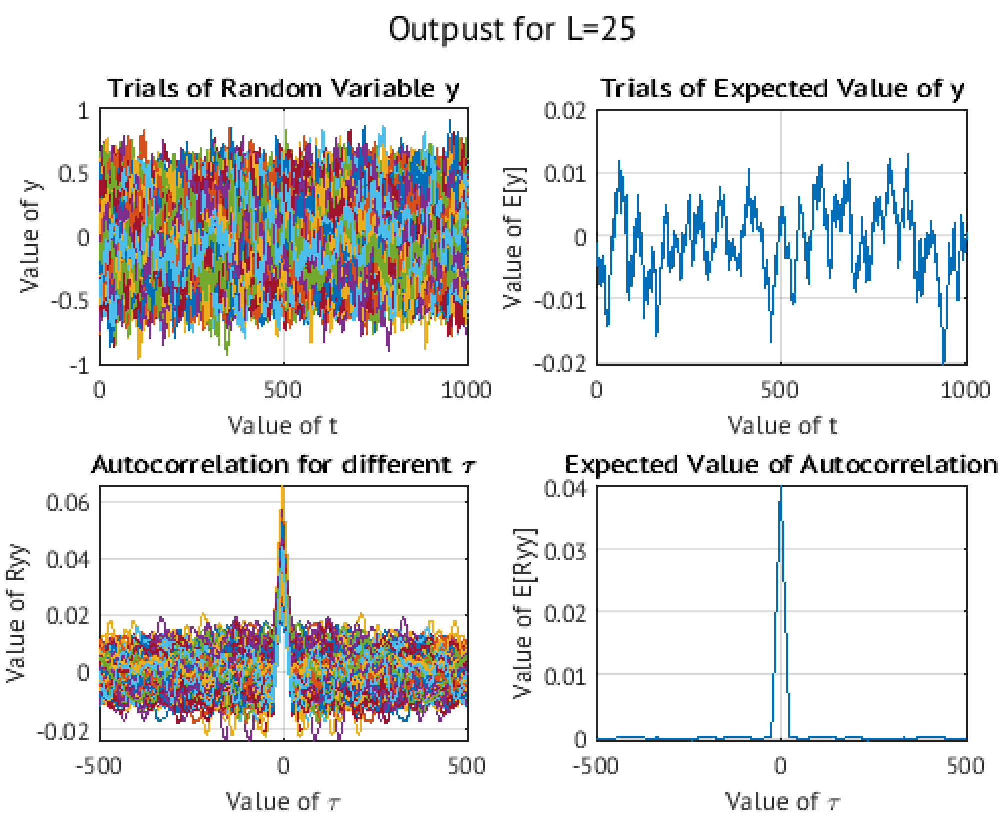
```
%}
```

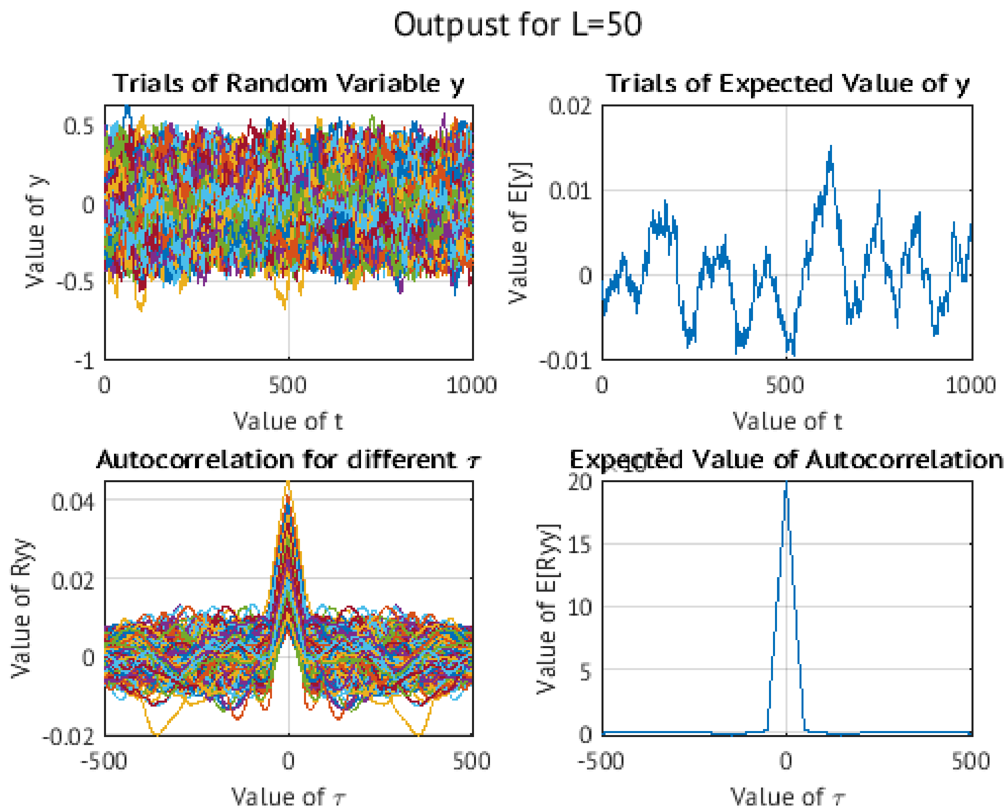
```
    0.9995
```

```
 0.1008

 9.9139

 0.0402

24.8850

 0.0199

50.1351
```




Outpust for L=10
Trials of Random Variable y
Value of y
Value of t
Trials of Expected Value of y
Value of E[y]
Value of t
Autocorrelation for different τ
Value of Ryy
Value of τ
Expected Value of Autocorrelation
Value of E[Ryy]
Value of τ


Outpust for L=25
Trials of Random Variable y
Value of y
Value of t
Trials of Expected Value of y
Value of E[y]
Value of t
Autocorrelation for different τ
Value of Ryy
Value of τ
Expected Value of Autocorrelation
Value of E[Ryy]
Value of τ



Published with MATLAB® R2021b