S22 CMPE320 Proj 2 Skeleton

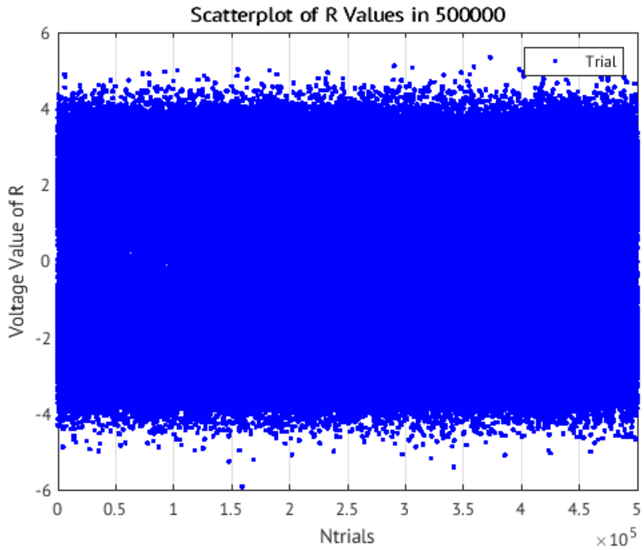
```
close all;
clear;

PrA = 0.5;  % per the project
Ntrials = 500000; % make this as large as you can for your machine.
%  From Project 1, more trials give results closer to the pdf.
A_minusA = (rand(1,Ntrials)<=PrA); % 1  = A, 0 = -A;
A_minusA = 2*(A_minusA-0.5);% convert to  +/-A;

Avalue = 2; % per assignment
sigma2 = 9/16; %per assignment;

N =  sqrt(sigma2)*randn(1,Ntrials); % zero mean variance = sigma2
R =  Avalue*A_minusA+N;  % R = (+/-A)+N;

tenSigma = sqrt(sigma2)*10;
dr=0.05;
rEdge=[-tenSigma-Avalue:dr:tenSigma+Avalue]; % force bin center to zero

% Figure (1) is the scatterplot
% plots each output value of R for each Ntrials value
figure(1)
x = [1:Ntrials];
y = R;
plot(x,y,'b.'); %create the scatterplot use an appropriate x, an appropriate y,
% the 'b.' will plot individual points in blue.
% prettify the graph
title(['Scatterplot of R Values in ', num2str(Ntrials)]);
ylabel('Voltage Value of R');
xlabel('Ntrials');
grid on;
legend('Trial');

% Figure(2) is the histogram
% Now create the histogram, normalized to pdf, as in Project 1.
figure(2)
spdfR = histogram(R, 'BinEdges', rEdge, 'Normalization', 'pdf');

[Vr,Nbinr,r]=unpackHistogram(spdfR);  %I've provided a helper function to assist with histogram management

% Vr is values of the histogram bins
% Nbinr is number of bins
% r is the bin centers

edges = rEdge;
rGivenA = exp(-(edges-Avalue).^2/(2*sigma2))/sqrt(2*pi*sigma2);
rGivenNegA = exp(-(edges-(-Avalue)).^2/(2*sigma2))/sqrt(2*pi*sigma2);
fRr = rGivenA * 0.5 + rGivenNegA * 0.5; % put the equation for your fR(r) here

hold on;
plot(edges, fRr, 'r', 'LineWidth', 3); % plot your fRr
hold off;

% Make the plot look professional
xlabel('Voltage');
ylabel('Probability Density');
grid on;
legend('Random Variable R', 'Theoretical Value of R');
title('Probability Density of R');



figure(3); %Scatterplot for 2.2

%  Method 1
%  Notice the trick here.  (R>=0) will be 1 when true and 0 when false.
%  Multiplying point by point using .* will set all negative values to zero
%  and leave all postive values unchanged, thus creating the S for 2.1

S = (R>=0).*R; %  only accept R>=0;
x = [1:Ntrials];
y = S;
%plot(x,y,'b.'); % scatterplot
plot(R, S, 'b.');

xlabel('Random Variable R');
ylabel('Random Output Variable S');
grid on;
legend('Output Voltage');
title('Voltage Output from Perfect Diode Detector');

% figure(5); % extra scatterplot
```

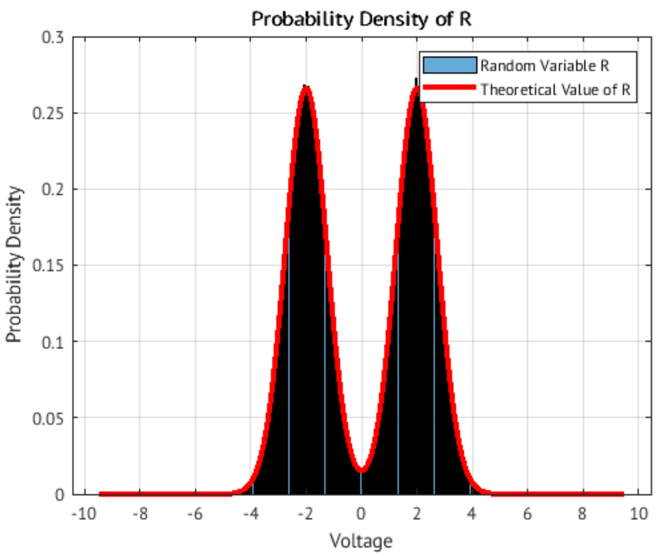
```
ds = dr;

figure(4);

sEdge = rEdge;
% you may use subplots or not, as you desire.  If not, then you'll need new figures
spdfS = histogram(S, 'BinEdges', sEdge, 'Normalization', 'pdf'); %generate normalized histogram as in Project 1

[Values,Ns,s]=unpackHistogram(spdfS); %Use the helper function

i0 = min(find(s>=0)); % locate s nearest to zero
%fSs = Values.*(s>0); % use the trick again
fSs = fRr; % pre editing

% set the value to that of the middle value of R, which is the bin at zero
% times the bin width of that bin
PrS_is_0 = sum(Vr(1:(length(Vr)/2 + 1))); % you will have some value


fSs(i0)=PrS_is_0; % create an effective Dirac Delta function at zero
fSs(1:i0-1) = 0; % sets every value before the dirac delta to 0

hold on
plot(sEdge,fSs,'r','LineWidth',3); % plot in red on top of histogram
hold off

% make the plot look professional
grid on;
xlabel('Value of S');
ylabel('Probability Density');
title('Probability Density Function of Random Variable S')
legend('Measured Probability Density', 'Analytical Probability Density');

% Would a rescaled version make things easier to see?
xlim([-0.5 5]);
ylim([0 .35]);

% old CDF stuff
% figure(5);
%
% % Plot the CDF from the histogram and Theoretical CDF
% CDF_S = histogram(S, 'BinEdges', sEdge, 'Normalization', 'cdf');
% % this is close, but not perfect yet
% % this might now be good
% FSs = (sEdge>=0) .* ((1 - QQ((sEdge - Avalue)/sqrt(sigma2))) * 0.5 + 0.5); % Hint:  Use the QQ helper function to express the integral in terms of Q
%
% hold on
% plot(sEdge, FSs, 'LineWidth', 2); % plot FSs
% hold off
%
% %Make the plot look professional
% grid on;
% xlabel('Value of S');
% ylabel('Cumulative Probability (CDF)');
% title('CDF of S Given Perfect Diode Detector');
% legend('S Normalized as CDF', 'Actual CDF Function');

meanS =  mean(S); % sample mean from S array, not the histogram
varS = var(S);  % sample variance from S array, not the histogram
meanR = mean(R); % sample mean from R array, not the histogram
varR=var(R);  % sample variance from the R array.

% Print the results (example only, do what you want)
disp('--------');
disp('Section 2.1');
disp(['For the input, with A = +/-',num2str(Avalue),' variance ',num2str(sigma2),' and ',int2str(Ntrials),' trials,']);
disp(['the mean of R is ',num2str(meanR),' with variance ',num2str(varR),' =  ',num2str(Avalue^2),'  +  ',num2str(sigma2)]);
disp(['For Method 1 (ideal diode), the mean of S is ',num2str(meanS),' and the variance is ',num2str(varS)])
disp(['For Method 1 (ideal diode), diode(',num2str(meanR),') = ',num2str(meanR*(meanR>=0))]);

%================
% 2.2 uses abs as the function, but the same signal model.
%      We'll retain N and R, and just replace S

%New figures as necessary

S2 = abs(R); % the second method is absolute value.

figure(6);
% Plot the scatter plot
x = [1:Ntrials];
y = S2;
%plot(x,y,'b.');
plot(R, S2, 'b.');

xlabel('Random Variable R');
ylabel('Random Output Variable S');
```

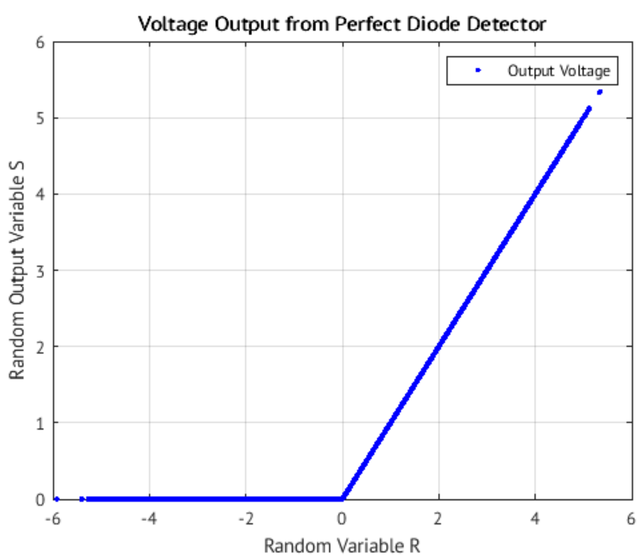
```
grid on;
legend('Output Voltage');
title('Voltage Output from Absolute Value Detector');

% New figure
figure(7);
s2Edge = sEdge;
spdfS2 = histogram(S2, 'BinEdges', s2Edge, 'Normalization', 'pdf');% generate the normalized histogram

[Values2,Ns2,s2]=unpackHistogram(spdfS2); % unpack for ease of use

% old method of finding fS2s
%constantVal2 = abs(s2Edge) ./ (s2Edge * sqrt(2 * pi * sigma2));
constantVal2 = 1 / (sqrt(2 * pi * sigma2));

negAval = exp(-(s2Edge - Avalue) .^ 2 / (2 * sigma2));
posAval = exp(-(s2Edge + Avalue) .^ 2 / (2 * sigma2));

fS2s = constantVal2 .* (negAval + posAval);
PRis0 = 2 * fRr(191); % probability of R = 0, taken from the above function
fS2s(191) = 2 * PRis0; % 2 times the value, due to the +-
fS2s(1:191) = 0;

hold on
plot(s2Edge, fS2s, 'r', 'LineWidth', 3); % your fSs
hold off

% Make your plot professional
grid on;
xlabel('Value of S');
ylabel('Probability Density');
title('Probabilty Density Function of S given an Absolute Value Detector');
legend('Measured Probability Density', 'Analytical Probability Density');

% old CDF stuff
% Compute and plot the CDF and print resultsmodifying lines 74-96 as necessary for this section
% figure(8);
%
% CDF_S2 = histogram(S2, 'BinEdges', s2Edge, 'Normalization', 'cdf');
% FS2s = (1-QQ((s2Edge-Avalue) / sqrt(sigma2)));
%
% hold on;
% plot(s2Edge, FS2s, 'LineWidth',2);
% hold off;

meanS2 =  mean(S2); % sample mean from S array, not the histogram
varS2 = var(S2);  % sample variance from S array, not the histogram

% Print the results (example only, do what you want)
disp('--------');
disp('Section 2.2');
disp(['For the input, with A = +/-',num2str(Avalue),' variance ',num2str(sigma2),' and ',int2str(Ntrials),' trials,']);
disp(['the mean of R is ',num2str(meanR),' with variance ',num2str(varR),' =  ',num2str(Avalue^2),'  +  ',num2str(sigma2)]);
disp(['For Method 2 (absolute value detector), the mean of S is ',num2str(meanS2),' and the variance is ',num2str(varS2)]);
disp(['For Method 2 (absolute value detector), diode(',num2str(meanR),') = ',num2str(meanR*(meanR>=0))]);

%================
% 2.3 uses S = R.^2 as the function, but the same signal model.
%      We'll retain N and R, and just replace S

% New plots
figure(9); % this is the scatterplot
S3 = R.^2;

% scatterplot
% this shows the bounds needed for sxEdge
x = [1:Ntrials];
y = S3;
%plot(x,y,'b.');
plot(R, S3, 'b.');

xlabel('Random Variable R');
ylabel('Random Output Variable S');
grid on;
legend('Output Voltage');
title('Voltage Output from Square Law Detector');

s3Edge = [0:ds:30];
% you may use subplots or not, as you desire.  If not, then you'll need new figures
% this should be the 10th figure
figure(10);
spdfS3 = histogram(S3, 'BinEdges', s3Edge, 'Normalization', 'pdf'); %generate normalized histogram as in Project 1

[Values3,Ns3,s3]=unpackHistogram(spdfS3); %Use the helper function
% this below is done to make the line actually go to where it is supposed to, other than infinity
s3Edge(1) = 0.001;
% this is the section to put the analytical value of the pdf
```

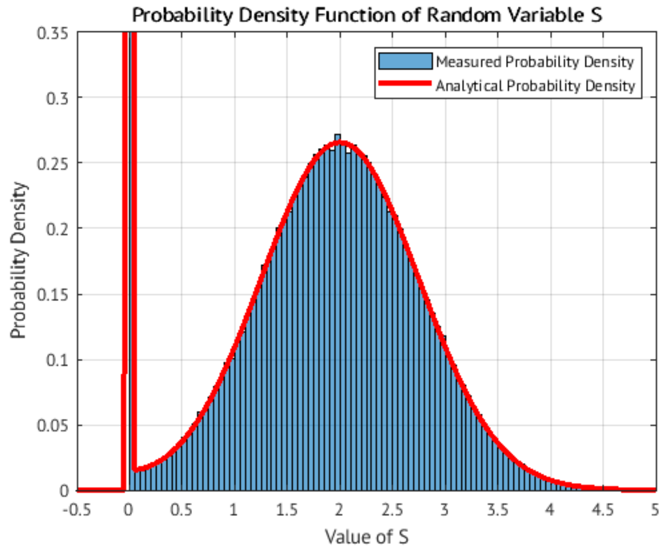
```
% which is found via the math stuff in Appendix B
constantVal = 1 ./ (2 * sqrt(s3Edge * 2 * pi * sigma2));
negAval = exp(-(sqrt(s3Edge) - Avalue) .^ 2 / (2 * sigma2));
posAval = exp(-(sqrt(s3Edge) + Avalue) .^ 2 / (2 * sigma2));
fS3s = (negAval + posAval) .* constantVal;

hold on;
plot(s3Edge, fS3s, 'r', 'LineWidth', 3);
hold off;

% prettiness on the graphs
ylim([0 0.15]);
xlim([0 30]);

grid on;
xlabel('Value of S');
ylabel('Probability Density');
title('Probabilty Density Function of S given a Square Law Detector');
legend('Measured Probability Density', 'Analytical Probability Density');

meanS3 =  mean(S3); % sample mean from S array, not the histogram
varS3 = var(S3);  % sample variance from S array, not the histogram

disp('--------');
disp('Section 2.3');
disp(['For the input, with A = +/-',num2str(Avalue),' variance ',num2str(sigma2),' and ',int2str(Ntrials),' trials,']);
disp(['the mean of R is ',num2str(meanR),' with variance ',num2str(varR),' =  ',num2str(Avalue^2),'  +  ',num2str(sigma2)]);
disp(['For Method 3 (squeare law detector), the mean of S is ',num2str(meanS3),' and the variance is ',num2str(varS3)]);
disp(['For Method 3 (square law detector), diode(',num2str(meanR),') = ',num2str(meanR*(meanR>=0))]);


% Print output table for use in report
%  This table provides the means and variances for the various options all
%  in one place.


% Jensens holds true due to the gER values being less than or equal to ES for all values
% func(E(x)) <= E(func(x)) <- this is the correct Jensens which is shown in the table through the collected data
disp('Output Table');
method =  [1:3]';
ES = [meanS,meanS2,meanS3]';
gER = [meanR*(meanR>=0), abs(meanR), meanR^2]';
table  =  [method ES gER];
%sprintf('%10.5f',table)
disp(table);
```

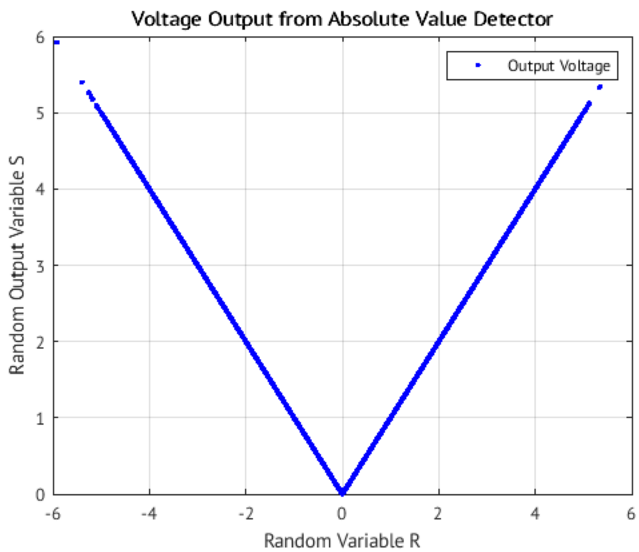
```
--------
Section 2.1
For the input, with A = +/-2 variance 0.5625 and 500000 trials,
the mean of R is -0.003123 with variance 4.5636 =  4  +  0.5625
For Method 1 (ideal diode), the mean of S is 0.99941 and the variance is 1.2797
For Method 1 (ideal diode), diode(-0.003123) = 0
--------
Section 2.2
For the input, with A = +/-2 variance 0.5625 and 500000 trials,
the mean of R is -0.003123 with variance 4.5636 =  4  +  0.5625
For Method 2 (absolute value detector), the mean of S is 2.0019 and the variance is 0.5558
For Method 2 (absolute value detector), diode(-0.003123) = 0
--------
Section 2.3
For the input, with A = +/-2 variance 0.5625 and 500000 trials,
the mean of R is -0.003123 with variance 4.5636 =  4  +  0.5625
For Method 3 (squeare law detector), the mean of S is 4.5636 and the variance is 9.6509
For Method 3 (square law detector), diode(-0.003123) = 0
Output Table
    1.0000    0.9994         0
    2.0000    2.0019    0.0031
    3.0000    4.5636    0.0000
```

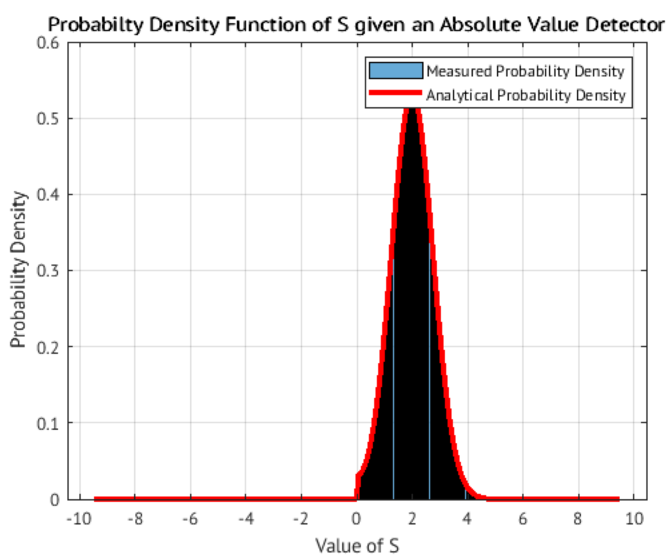

Scatterplot of R Values in 500000
Trial
Voltage Value of R
Ntrials
×10^5

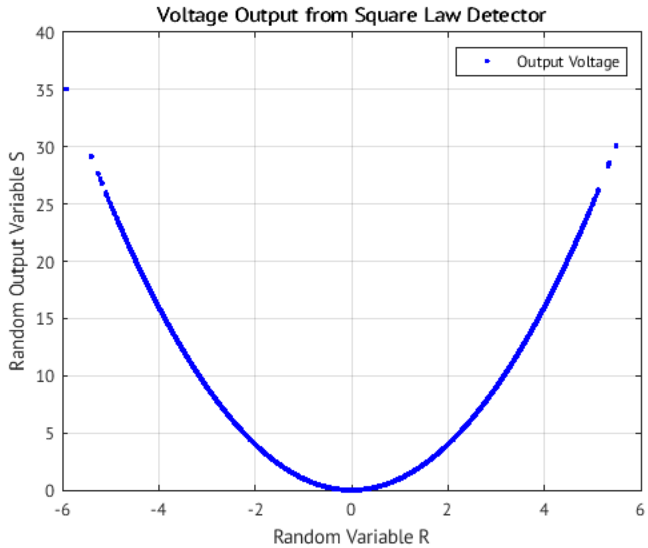

Probability Density of R
Random Variable R
Theoretical Value of R
Probability Density
Voltage

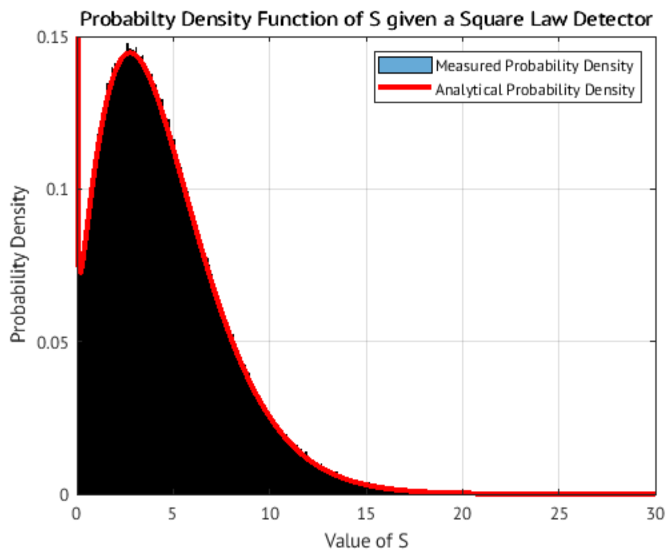

Voltage Output from Perfect Diode Detector
Output Voltage
Random Output Variable S
Random Variable R


Probability Density Function of Random Variable S
Measured Probability Density
Analytical Probability Density
Probability Density
Value of S


Voltage Output from Absolute Value Detector
Output Voltage
Random Output Variable S
Random Variable R


Probabilty Density Function of S given an Absolute Value Detector
Measured Probability Density
Analytical Probability Density
Probability Density
Value of S





Published with MATLAB® R2021b