MEMO Number CMPE320-S21-PROJ2 CODE
DATE: 03/31/2021
TO: EFC LaBerge
FROM: Nem Negash
SUBJECT:

1 MATLAB CODE

1.1 Method 1

```
function part2_1()
    figure(1);
    %analytical graph
    syms s
    f_s = piecewise(s < 0, 0, s == 0, 0.5 , s > 0, 0.5*dirac(s)+ (
(0.5) / (sqrt(2*pi * (9/16) ) ) * ( exp (-( (s-2).^2) / (2* (9/16)) )+
( exp (-( (s+2).^2) / (2* (9/16)) )) )));
    fplot(f_s,'Linewidth',3);

    %simulation graph
   %simulation graph
    A = 2; % Known Amplitude
    sigma2 = 9/16; % Known Noise Variance
    N = sqrt(sigma2) * randn(1,100000); % N(0,9/16)
    X = (rand(1,100000)<=0.5)*A; % Create X as a random function less
than 0.5 with amplitude A
    X = 2*(X - A/2); % Set X as 2 * the random function - 1
    R = X + N; % Create the Recieved Signal
    hold on;

    %mean of the fucntion
    T = mean(R);
    disp('Jenson''s Inequality evalution for 2.1');
    fprintf('g(E[x]): %.3f\n',T);
    %setup trials
    S = zeros(1, 100000);
    for i=1:100000
        if R(i) >= 0
            S(i) = R(i);
        end
    end

    %simulation mean
    M = mean(S);
    fprintf('E[g(x)]: %.3f\n',M);
    disp('The Jensen''s inequality rule states E[g(X)]?g(E[X]) which
holds for this trial.');
    %histogram
    histogram(S,'Normalization','pdf');
    %graph detailss
    xlabel('s(Volts)');
    ylabel('f_S(s)');
```

```
    title('PDF f_S(s) S = R,R>=0,0 o/w');
    legend('Analytical','Histogram')
    grid on

end
```

1.2 Method 2

```
function part2_2()
    figure(1);
    %analytical graph
    syms s
    f_s =   piecewise(s <= 0, 0, s > 0,( 1 / (sqrt(2*pi * (9/16) ) ) *
( exp (-( (s-2).^2) / (2* (9/16)) )+ ( exp (-( (s+2).^2) / (2* (9/16))
)) )));
    fplot(f_s,'Linewidth',3);

    %simulation graph
    A = 2; % Known Amplitude
    sigma2 = 9/16; % Known Noise Variance
    N = sqrt(sigma2) * randn(1,100000); % N(0,9/16)
    X = (rand(1,100000)<=0.5)*A; % Create X as a random function less
than 0.5 with amplitude A
    X = 2*(X - A/2); % Set X as 2 * the random function - 1
    R = X + N; % Create the Recieved Signal
    hold on;
    %mean of the function
    T = mean(abs(R(:)));
    disp('Jenson''s Inequality evalution for 2.2');
    fprintf('g(E[x]): %.3f\n',T);
    %setup up trials
    S = [];
    for i=1:100000
            S = [S,abs(R(i))];
    end
    %simulation mean
    M = mean(S);
    fprintf('E[g(x)]: %.3f\n',M);
    disp('The Jensen''s inequality rule states E[g(X)]?g(E[X]) which
holds for this trial.');
    %histogram
    histogram(S,'Normalization','pdf');
    %graph detials
    xlabel('s(Volts)');
    ylabel('f_S(s)');
    title('PDF f_S(s) S = |R|');
    legend('Analytical','Histogram');
    grid on

end
```

1.3 Method 3

```
function part2_3()
    figure(1);
    %analytical graph
```

```
    syms s;
    f_s = ((1/(sqrt(s)))* ( (0.5) / (sqrt(2*pi * (9/16) ) ) * ( exp (-(
(sqrt(s)-2).^2) / (2* (9/16)) )+ ( exp (-( (sqrt(s)+2).^2) / (2*
(9/16)) )) )));
    fplot(f_s,'Linewidth',3);

    %simulation graph
    A = 2; % Known Amplitude
    sigma2 = 9/16; % Known Noise Variance
    N = sqrt(sigma2) * randn(1,100000); % N(0,9/16)
    X = (rand(1,100000)<=0.5)*A; % Create X as a random function less
than 0.5 with amplitude A
    X = 2*(X - A/2); % Set X as 2 * the random function - 1
    R = X + N; % Create the Recieved Signal
    hold on;
    %mean of the function
    T = (mean(R(:).^2));
    disp('Jenson''s Inequality evalution for 2.3');
    fprintf('g(E[x]): %.3f\n',T);
    %set up trials
    S = [];
    for i=1:100000
            S = [S,R(i).^2];
    end

    %simulation mean
    M = mean(S);
    fprintf('E[g(x)]: %.3f\n',M);
    disp('The Jensen''s inequality rule states E[g(X)]?g(E[X]) which
holds for this trial.');
    %histogram
    histogram(S,'Normalization','pdf');
    %graph details
    xlabel('s(Volts)');
    ylabel('f_S(s)');
    title('PDF f_S(s) S = R^2');
    legend('Analytical','Histogram')
    grid on

end
```