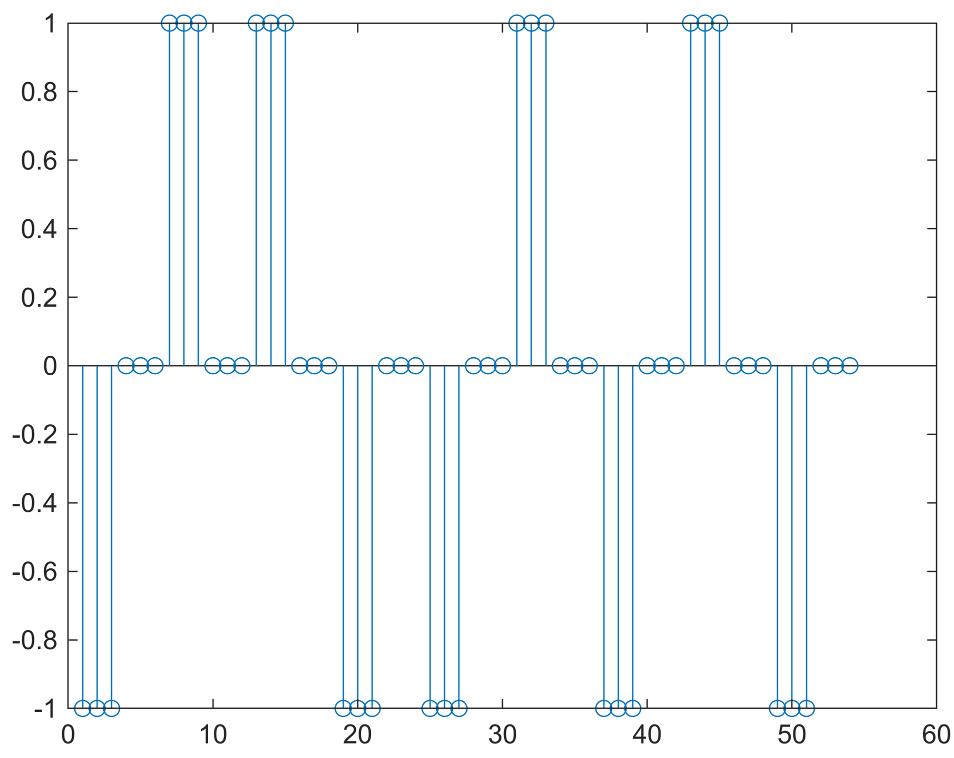
Syed Asghar Abbas Zaidi

Lab3 Task1

```
clc;clear;close all
 A = 1;J = 4;M = 6;T = 1;
 sNt = [];
 Bits = (sign(randn(1,2*J+1))+1)/2
```

```
Bits = 1×9
    0     1     1     0     0     1     0     1     0
```

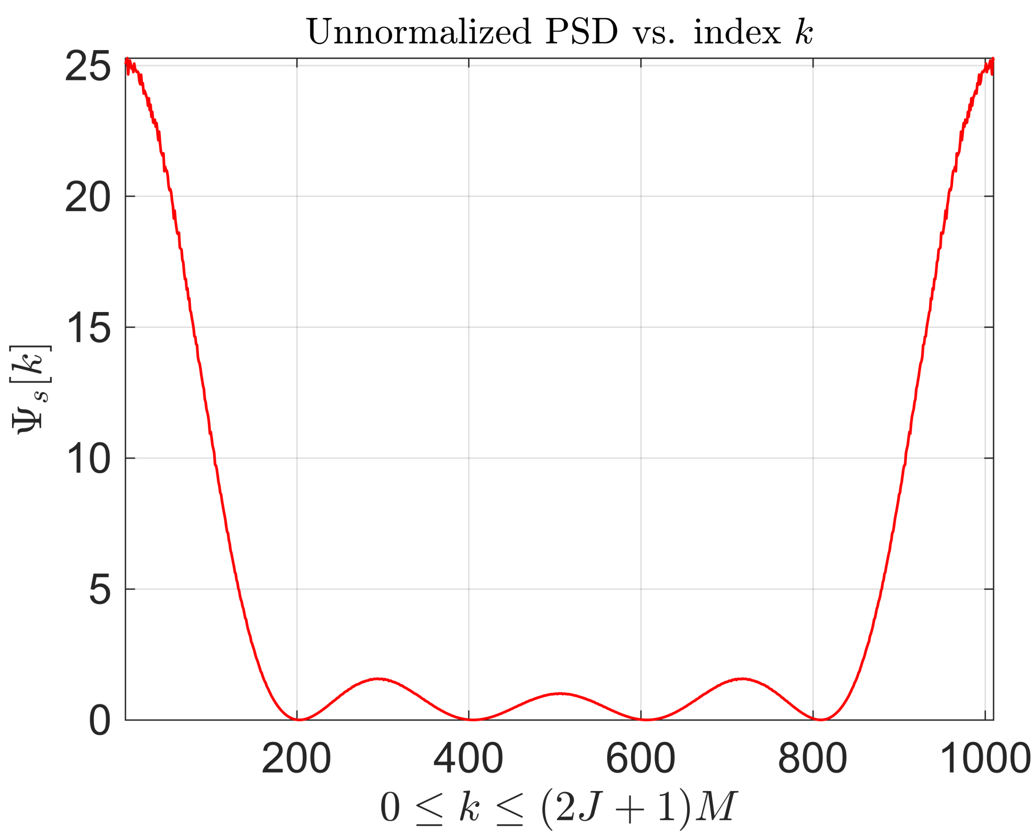
```
 for ii=1:2*J+1
 if Bits(ii)==1
            sNt=[sNt A*ones(1,M/2) A*zeros(1,M/2)];
 else
            sNt=[sNt -A*ones(1,M/2) -A*zeros(1,M/2)];
 end
 end
 stem(sNt)
```



Power Spectral Desntiy for the Polar NZ!

```
 A = 1;J = 50;M = 10;L = 20000;PSD = 0;
 T = 1; Ts=T/M; fs=1/Ts;
 for jj=1:L
 sNt = [];
 Bits = (sign(randn(1,2*J+1))+1)/2;
```

```
    for ii=1:2*J+1
        if Bits(ii)==1
            sNt=[sNt A*ones(1,M/2) A*zeros(1,M/2)];
        else
            sNt=[sNt -A*ones(1,M/2) -A*zeros(1,M/2)];
        end
    end
    PSD = PSD + abs(fft(sNt)).^2;
end
PSD = PSD/L/(2*J+1)/T;
LPSD = length(PSD);
figure(1); plot(PSD,'r-','linewidth',1)
set(gca,'FontSize',16);axis tight;grid on;
xlabel('$0\le k\le (2J+1)M$','FontSize',16,'interpreter','latex')
title('Unnormalized PSD vs. index $k$','FontSize',14,'interpreter','latex')
ylabel('$\Psi_s[k]$','FontSize',16,'interpreter','latex');
h=text(400,50,['$J = ' num2str(J) ', M = ' num2str(M) '$']);
set(h,'fontsize',14,'interpreter','latex');
```

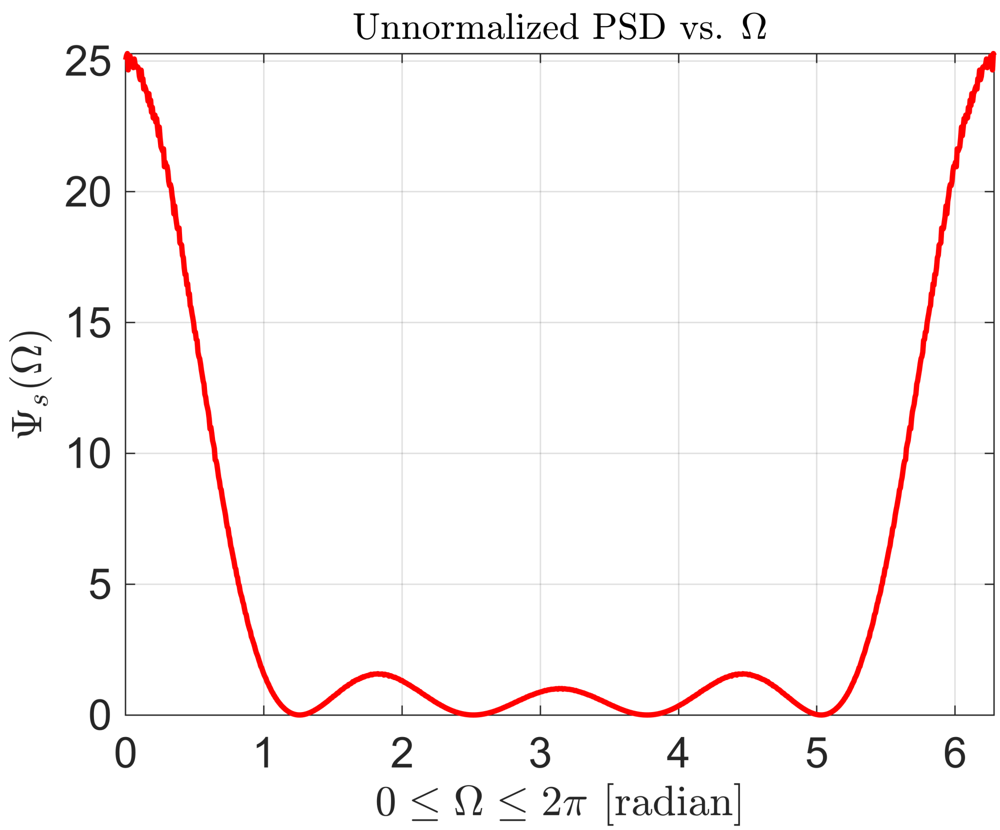



```
drawnow
```

For Unnormalized PSD vs Omega Code:

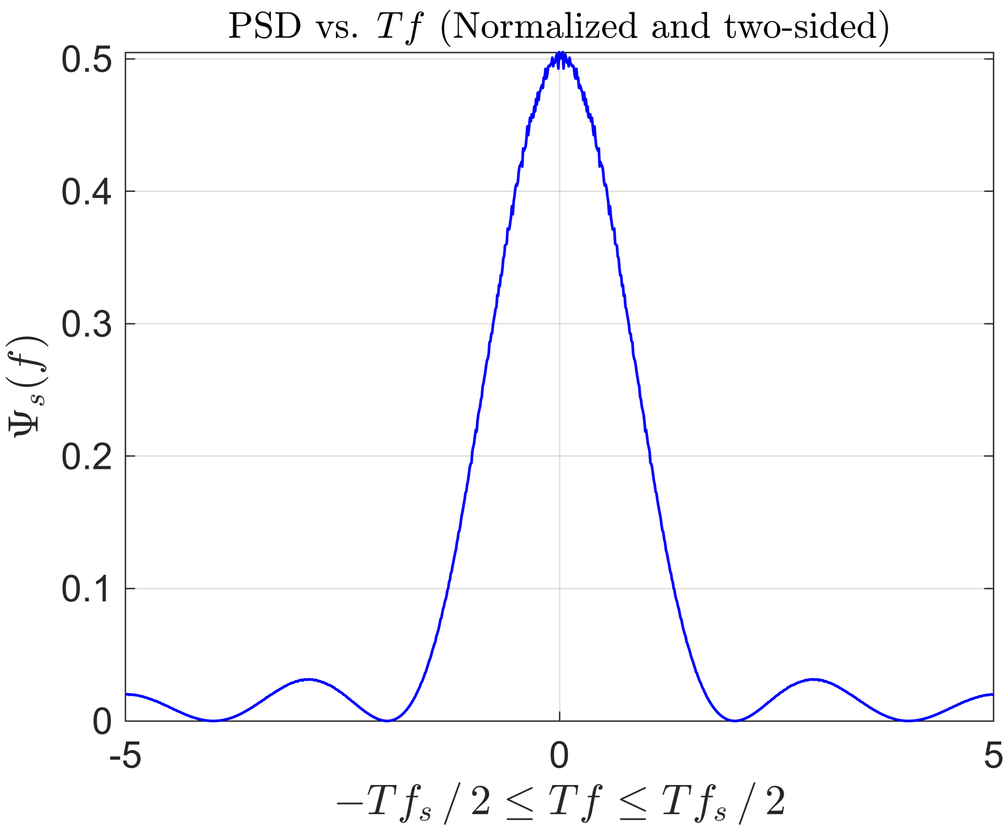
```
DTFT_xaxis = linspace(0,2*pi,LPSD);
figure(2);plot(DTFT_xaxis,PSD,'r-','linewidth',2)
```

```
set(gca,'FontSize',16);axis tight; grid on;
xlabel('$0 \le \Omega\le 2\pi$ [radian]','FontSize',16,'interpreter','latex')
title('Unnormalized PSD vs. $\Omega$','FontSize',14,'interpreter','latex')
ylabel('$\Psi_s(\Omega)$','FontSize',16,'interpreter','latex');drawnow
```



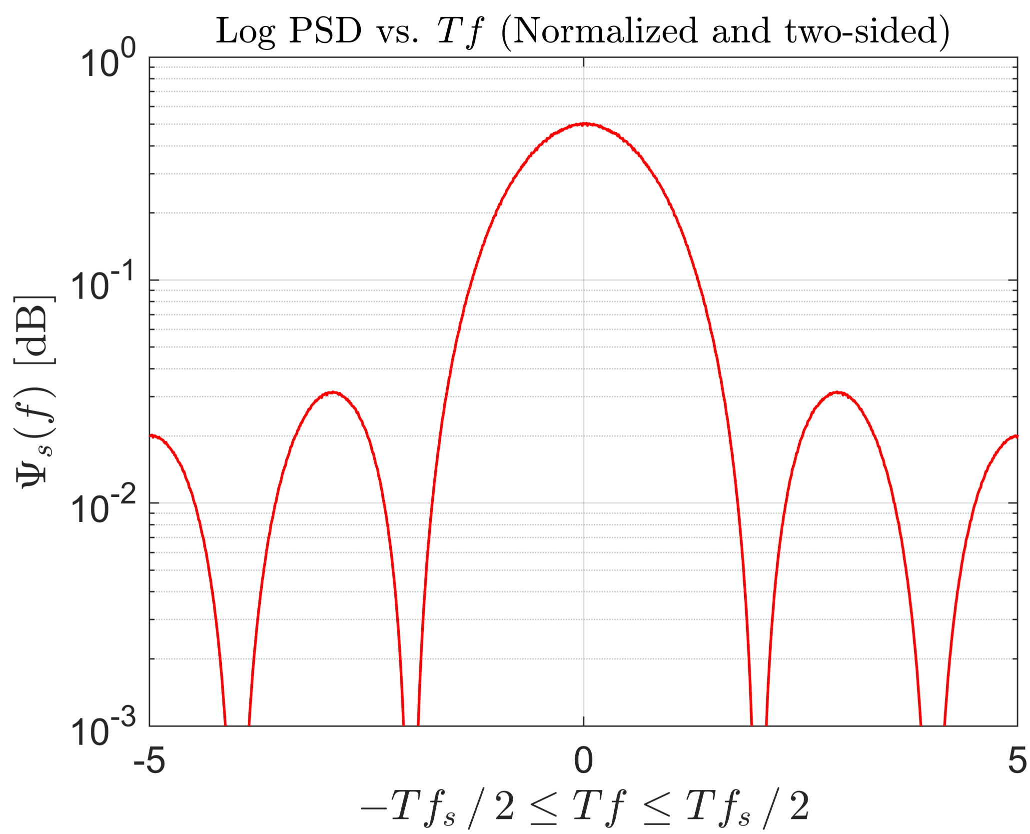
PSD vs T*f* (which is normalized as well as two-sided)

```
 fT = linspace(-T*fs/2,T*fs/2,LPSD); % normalized freq axis
Area = sum(PSD)*(fT(2)-fT(1));
 CTFT_PSD = PSD/Area; % normalized PSD
 CTFT_PSD = [fliplr(CTFT_PSD(end:-1:floor(LPSD/2)))...
    CTFT_PSD(1:floor(LPSD/2)-1)];
 figure(3);plot(fT,CTFT_PSD,'b-','linewidth',1)
 set(gca,'FontSize',14);axis tight; grid on;
 xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$','FontSize',16,'interpreter','latex')
 ylabel('$\Psi_s(f)$','FontSize',16,'interpreter','latex');
 xlim([-fs/2 fs/2]);title('PSD vs. $Tf$ (Normalized and two-sided)',...
 'FontSize',14,'interpreter','latex');drawnow
```



Log PSD vs Tf (Which is now Normalzied and two-sided)

```
figure(4);semilogy(fT,CTFT_PSD,'r-','linewidth',1);
set(gca,'FontSize',14);axis tight; grid on;
xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$','FontSize',16,'interpreter','latex')
ylabel('$\Psi_s(f)$ [dB]','FontSize',16,'interpreter','latex');ylim([1e-3
1]);
xlim(T*[-fs/2 fs/2]);title('Log PSD vs. $Tf$ (Normalized and two-sided)',...
'FontSize',14,'interpreter','latex');drawnow
```



Task 2

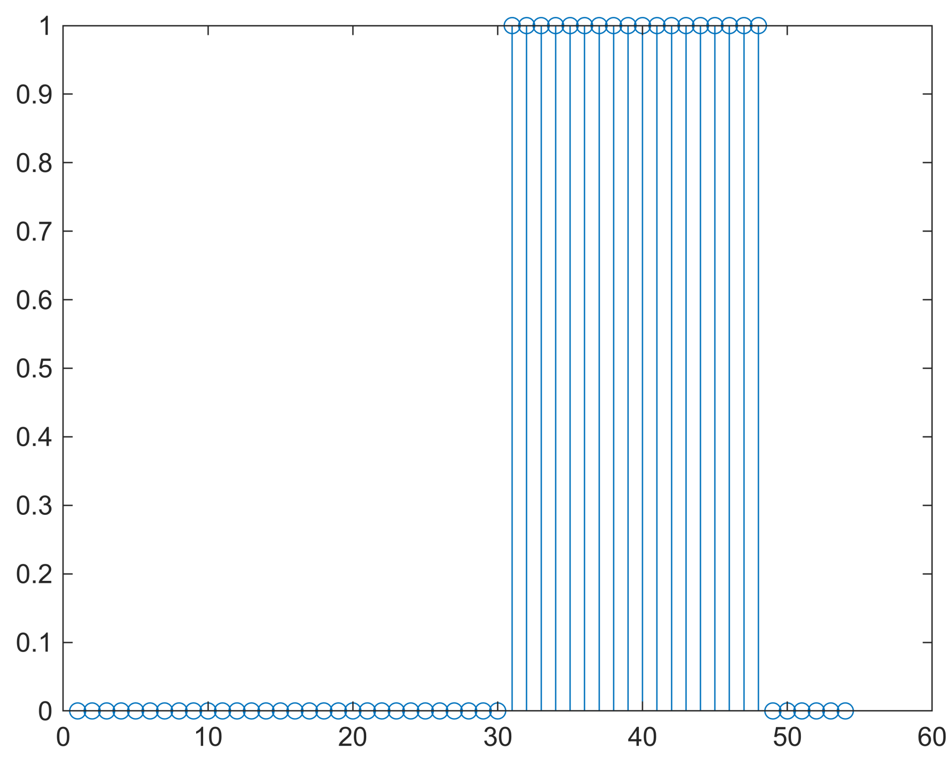
Now, we will try verifying the **Line coding** of Unipolar NRZ

```
%Syed Asghar Abbas Zaidi
clc;clear;close all
A = 1;
J = 4;
M = 6;
T = 1;
sNt = [];
Bits = (sign(randn(1,2*J+1))+1)/2
```

```
Bits = 1×9
     0     0     0     0     0     1     1     1     0
```

```
for ii=1:2*J+1
if Bits(ii)==1
           sNt=[sNt A*ones(1,M)];
else
            sNt=[sNt -A*zeros(1,M)];
```

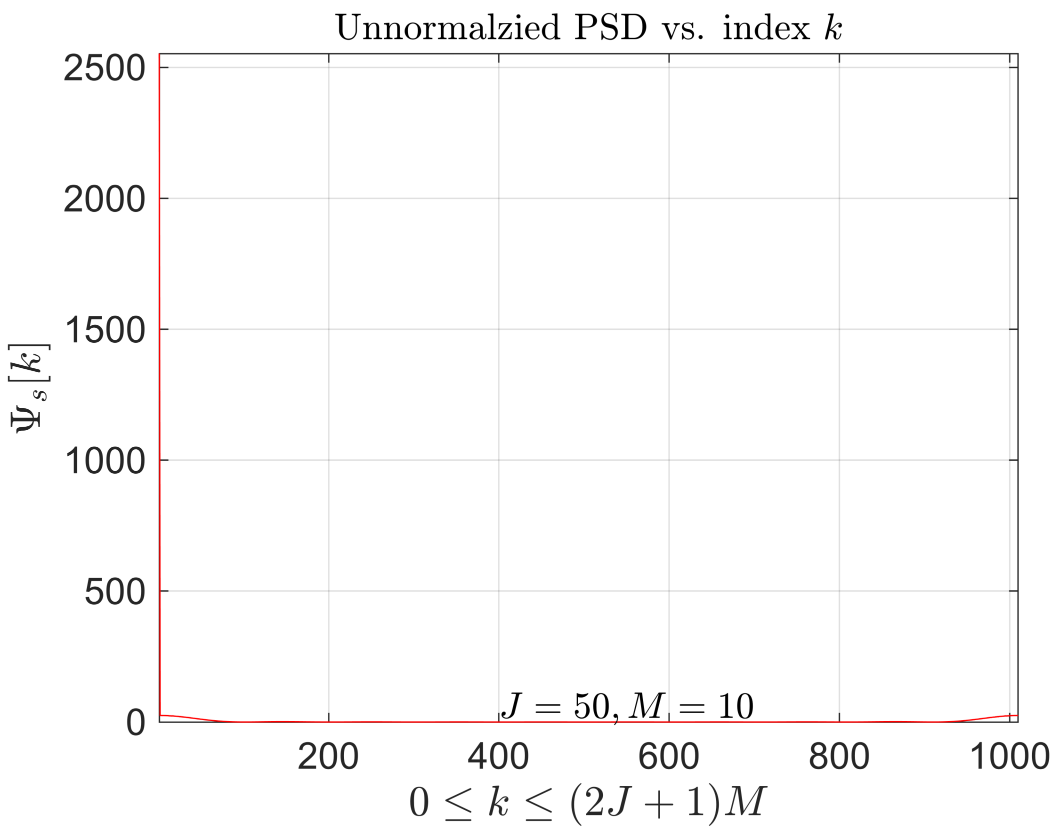
```
end
end
stem(sNt);
```



The Power Spectral Density (PSD) Calculations for Unipolar NRZ

```
A = 1;J = 50;M = 10;L = 20000;PSD = 0;
T = 1; Ts=T/M; fs=1/Ts;
for jj=1:L
sNt = [];
Bits = (sign(randn(1,2*J+1))+1)/2;
   for ii=1:2*J+1
       if Bits(ii)==1
           sNt=[sNt A*ones(1,M)];
        else
            sNt=[sNt -A*zeros(1,M)];
       end
   end
   PSD = PSD + abs(fft(sNt)).^2;
end
PSD = PSD/L/(2*J+1)/T;
LPSD = length(PSD);
figure(1); plot(PSD,'r-')
set(gca,'FontSize',14);axis tight;grid on;
xlabel('$0\le k\le (2J+1)M$','FontSize',16,'interpreter','latex')
```

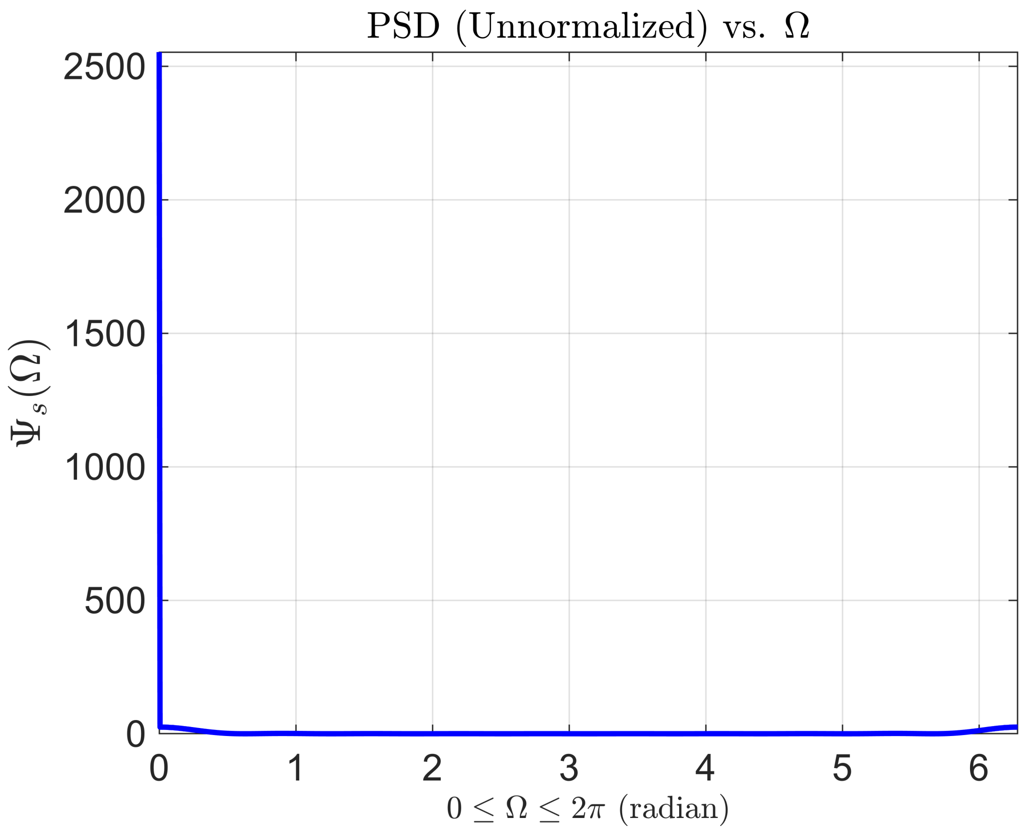
```
title('Unnormalzied PSD vs. index $k$','FontSize',14,'interpreter','latex')
ylabel('$\Psi_s[k]$','FontSize',16,'interpreter','latex');
h=text(400,50,['$J = ' num2str(J) ', M = ' num2str(M) '$']);
set(h,'fontsize',14,'interpreter','latex');
```




```
drawnow
```

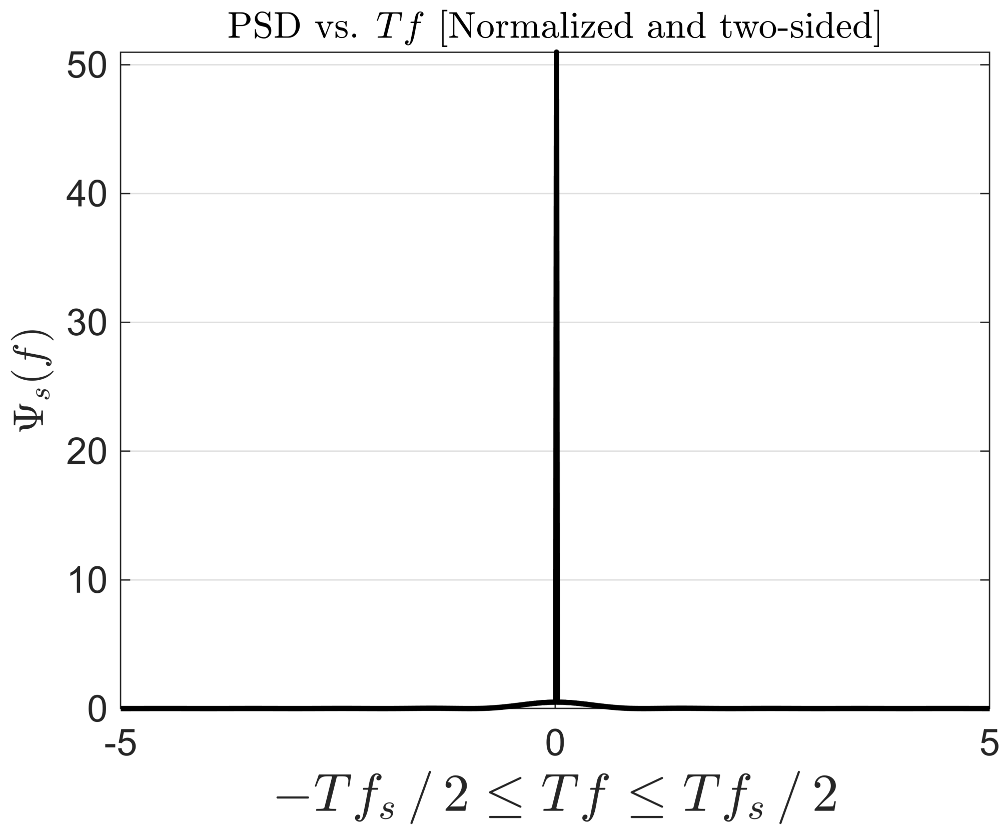
Unnormalized PSD vs Omega

```
 DTFT_xaxis = linspace(0,2*pi,LPSD);
figure(2);plot(DTFT_xaxis,PSD,'b-','linewidth',2)
set(gca,'FontSize',14);axis tight; grid on;
xlabel('$0 \le \Omega\le 2\pi$ (radian)','FontSize',12,'interpreter','latex')
title('PSD (Unnormalized) vs. $\Omega$','FontSize',14,'interpreter','latex')
ylabel('$\Psi_s(\Omega)$','FontSize',16,'interpreter','latex');drawnow
```



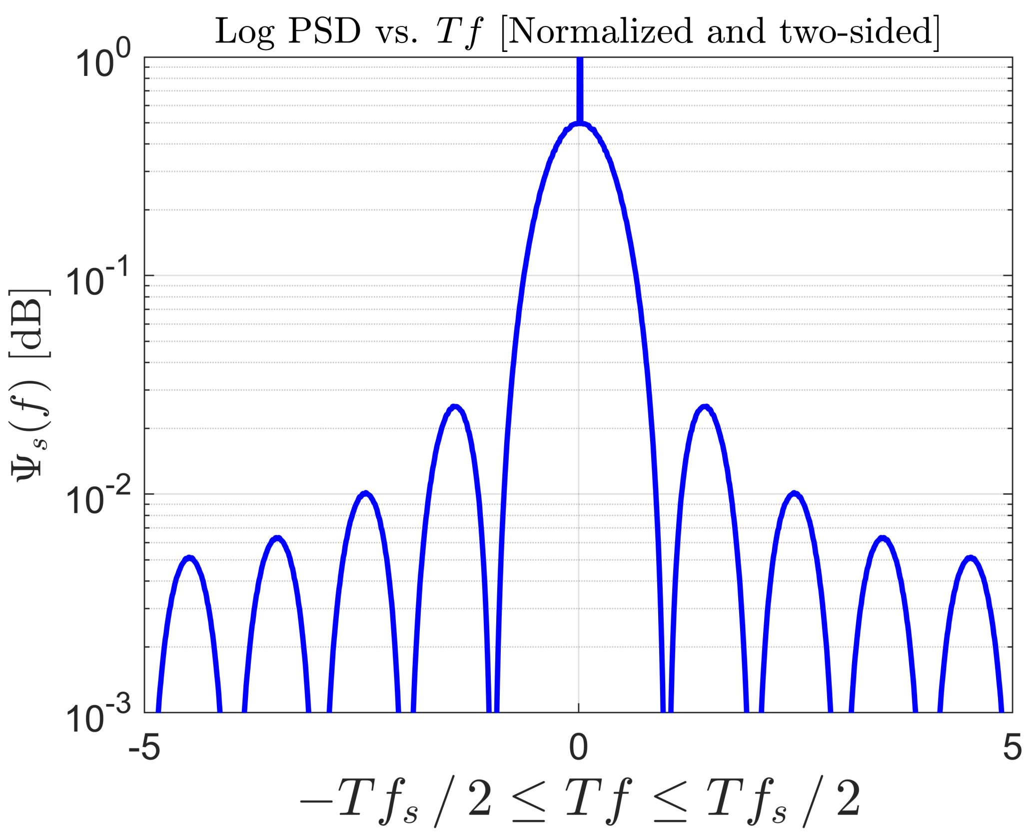
PSD vs T*f* (which is normalized as well as two-sided)

```
 fT = linspace(-T*fs/2,T*fs/2,LPSD); % normalized freq axis
Area = sum(PSD)*(fT(2)-fT(1));
 CTFT_PSD = PSD/Area; % normalized PSD
 CTFT_PSD = [fliplr(CTFT_PSD(end:-1:floor(LPSD/2)))...
    CTFT_PSD(1:floor(LPSD/2)-1)];
 figure(3);plot(fT,CTFT_PSD,'k-','linewidth',2)
 set(gca,'FontSize',14);axis tight; grid on;
 xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$','FontSize',20,'interpreter','latex')
 ylabel('$\Psi_s(f)$','FontSize',16,'interpreter','latex');
 xlim([-fs/2 fs/2]);title('PSD vs. $Tf$ [Normalized and two-sided]',...
 'FontSize',14,'interpreter','latex');drawnow
```



Log Power Spectral Density vs T*f* (Which is normalized as well as two-sided)

```
figure(4);semilogy(fT,CTFT_PSD,'b-','linewidth',2);
 set(gca,'FontSize',14);axis tight; grid on;
 xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$','FontSize',20,'interpreter','latex')
 ylabel('$\Psi_s(f)$ [dB]','FontSize',16,'interpreter','latex');ylim([1e-3
1]);
 xlim(T*[-fs/2 fs/2]);title('Log PSD vs. $Tf$ [Normalized and two-sided]',...
 'FontSize',14,'interpreter','latex');drawnow
```



Task 3

Now trying to verify the line coding of Unipolar RZ

```
clc;clear;close all
 A = 1;J = 4;M = 6;T = 1;
 sNt = [];
 Bits = (sign(randn(1,2*J+1))+1)/2
```

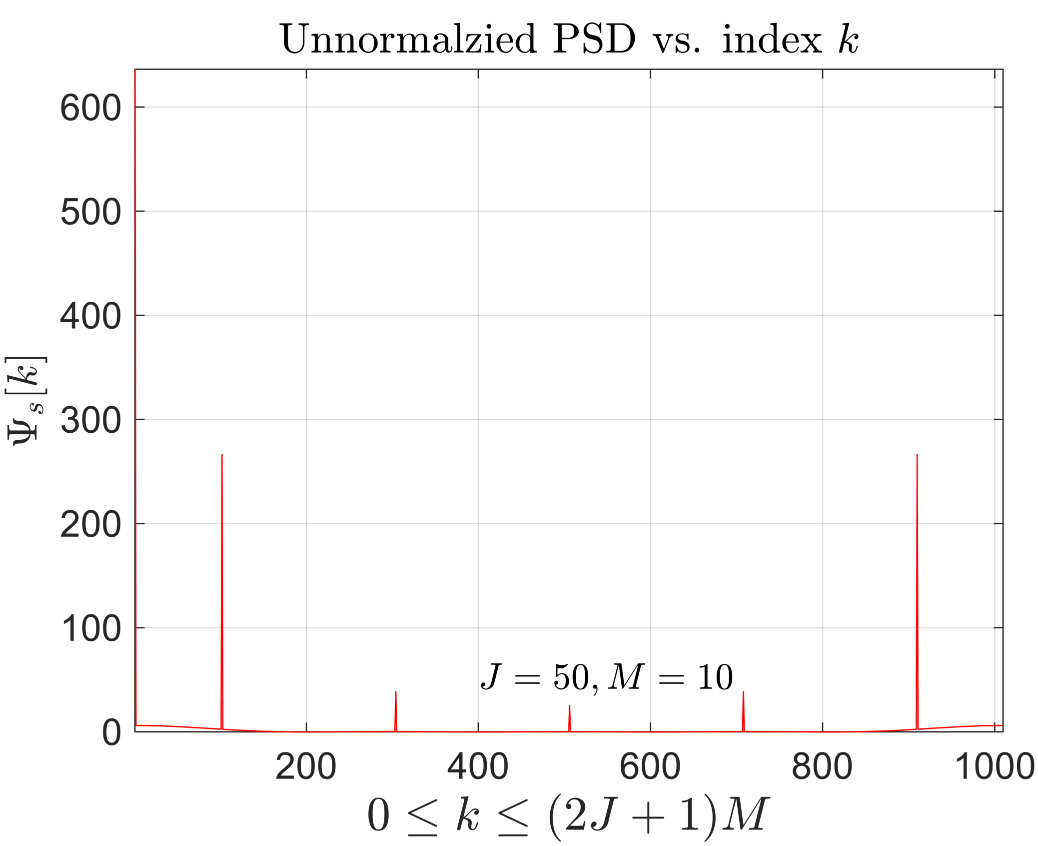
```
Bits = 1×9
     0     0     0     1     1     1     0     1     0
```

```
 for ii=1:2*J+1
 if Bits(ii)==1
            sNt=[sNt A*ones(1,M/2) A*zeros(1,M/2)];
 else
             sNt=[sNt -A*zeros(1,M/2) -A*zeros(1,M/2)];
 end
 end
 stem(sNt);
```

Power Spectral Calculations for the Unipolar RZ

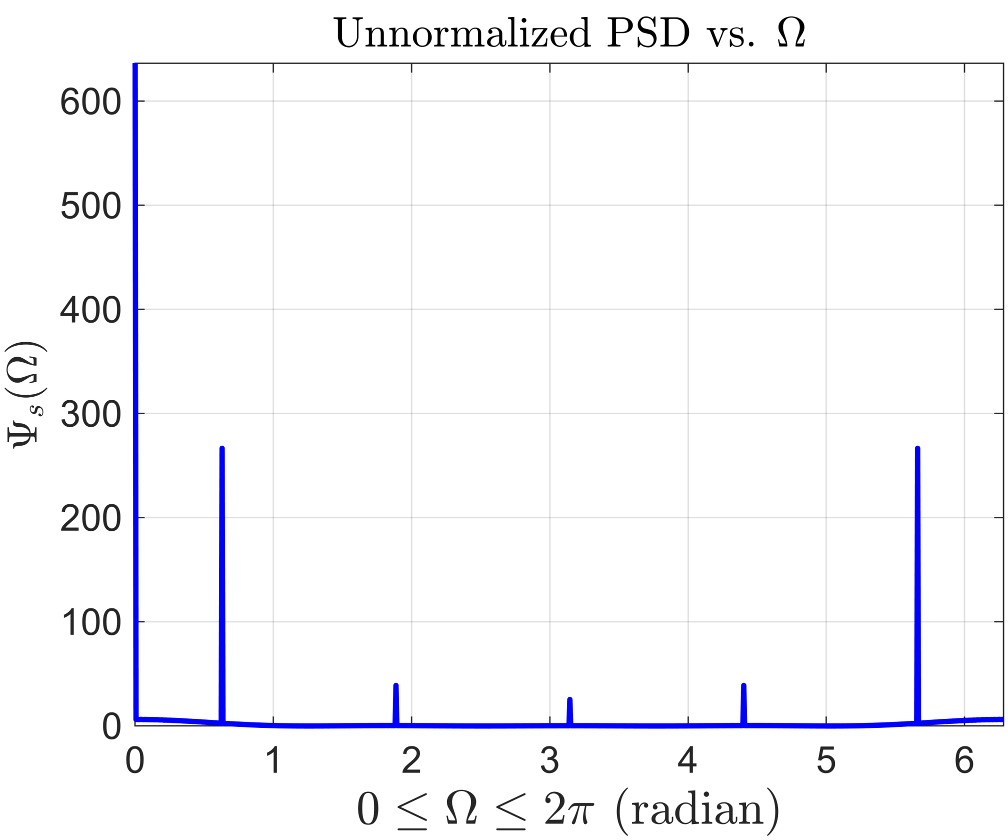
```
 clc;clear;close all
 A = 1;
```

```
J = 50;
M = 10;
L = 20000;
PSD = 0;
T = 1; Ts=T/M; fs=1/Ts;
for jj=1:L
sNt = [];
Bits = (sign(randn(1,2*J+1))+1)/2;
   for ii=1:2*J+1
       if Bits(ii)==1
           sNt=[sNt A*ones(1,M/2) A*zeros(1,M/2)];
        else
            sNt=[sNt -A*zeros(1,M/2) -A*zeros(1,M/2)];
       end
   end
   PSD = PSD + abs(fft(sNt)).^2;
end
PSD = PSD/L/(2*J+1)/T;
LPSD = length(PSD);
figure(1); plot(PSD,'r-');
set(gca,'FontSize',14);axis tight;grid on;
xlabel('$0\le k\le (2J+1)M$','FontSize',18,'interpreter','latex')
title('Unnormalzied PSD  vs. index $k$','FontSize',16,'interpreter','latex')
ylabel('$\Psi_s[k]$','FontSize',16,'interpreter','latex');
h=text(400,50,['$J = ' num2str(J) ', M = ' num2str(M) '$']);
set(h,'fontsize',14,'interpreter','latex');drawnow
```



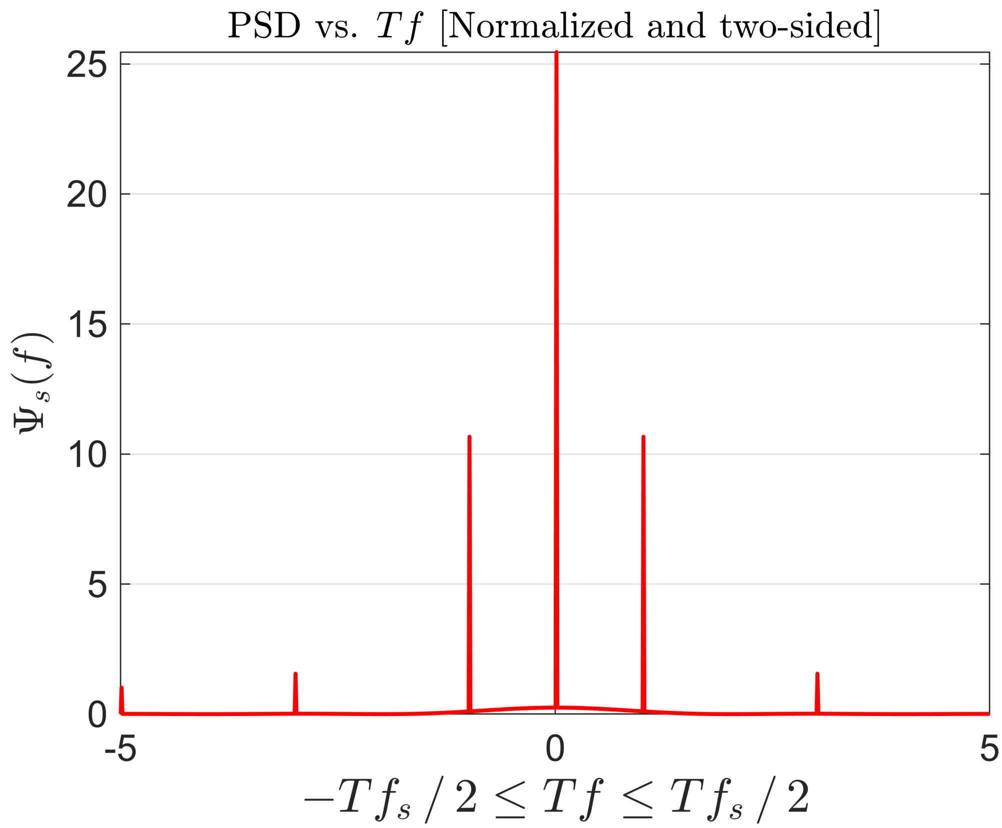
Unnormalized PSD vs Omega!

```
 DTFT_xaxis = linspace(0,2*pi,LPSD);
figure(2);plot(DTFT_xaxis,PSD,'b-','linewidth',2)
set(gca,'FontSize',14);axis tight; grid on;
xlabel('$0 \le \Omega\le 2\pi$ (radian)','FontSize',18,'interpreter','latex')
title('Unnormalized PSD vs. $\Omega$','FontSize',16,'interpreter','latex')
ylabel('$\Psi_s(\Omega)$','FontSize',16,'interpreter','latex');drawnow
```



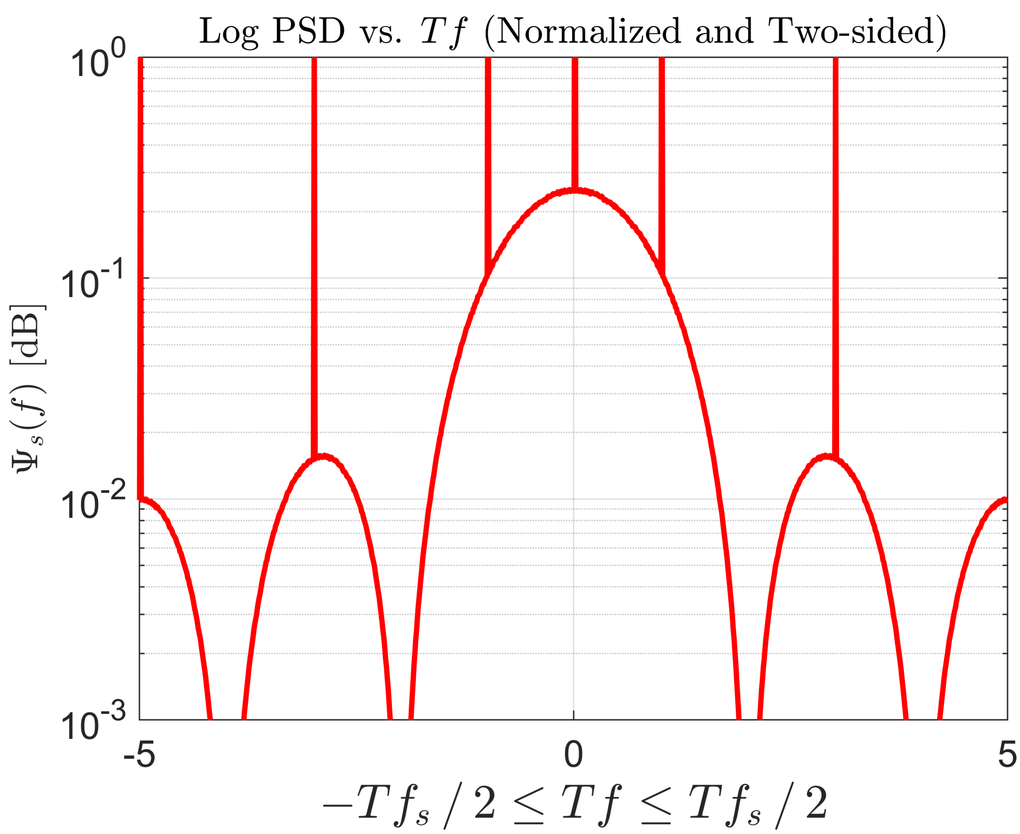
Power Spectral Density vs T*f* (Normalized as well as two-sided)

```
 fT = linspace(-T*fs/2,T*fs/2,LPSD); % normalized freq axis
Area = sum(PSD)*(fT(2)-fT(1));
 CTFT_PSD = PSD/Area; % normalized PSD
 CTFT_PSD = [fliplr(CTFT_PSD(end:-1:floor(LPSD/2)))...
    CTFT_PSD(1:floor(LPSD/2)-1)];
 figure(3);plot(fT,CTFT_PSD,'r-','linewidth',1.5)
 set(gca,'FontSize',14);axis tight; grid on;
 xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$','FontSize',18,'interpreter','latex')
 ylabel('$\Psi_s(f)$','FontSize',16,'interpreter','latex');
 xlim([-fs/2 fs/2]);title('PSD vs. $Tf$ [Normalized and two-sided]',...
 'FontSize',14,'interpreter','latex');drawnow
```



Log Power Spectral Density vs T*f* (Normalized as well as two sided)

```
figure(4);semilogy(fT,CTFT_PSD,'r-','linewidth',2);
set(gca,'FontSize',14);axis tight; grid on;
xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$','FontSize',18,'interpreter','latex')
ylabel('$\Psi_s(f)$ [dB]','FontSize',14,'interpreter','latex');ylim([1e-3
1]);
xlim(T*[-fs/2 fs/2]);title('Log PSD vs. $Tf$ (Normalized and Two-sided)',...
'FontSize',14,'interpreter','latex');drawnow
```



Task 4

Verifying the Line coding for the Bipolar NRZ!

```
clc;clear;close all
A = 1;J = 6;M = 4;a = 0;
T = 1; Ts=T/M; fs=1/Ts;
sNt = [];
Bits = (sign(randn(1,2*J+1))+1)/2
```

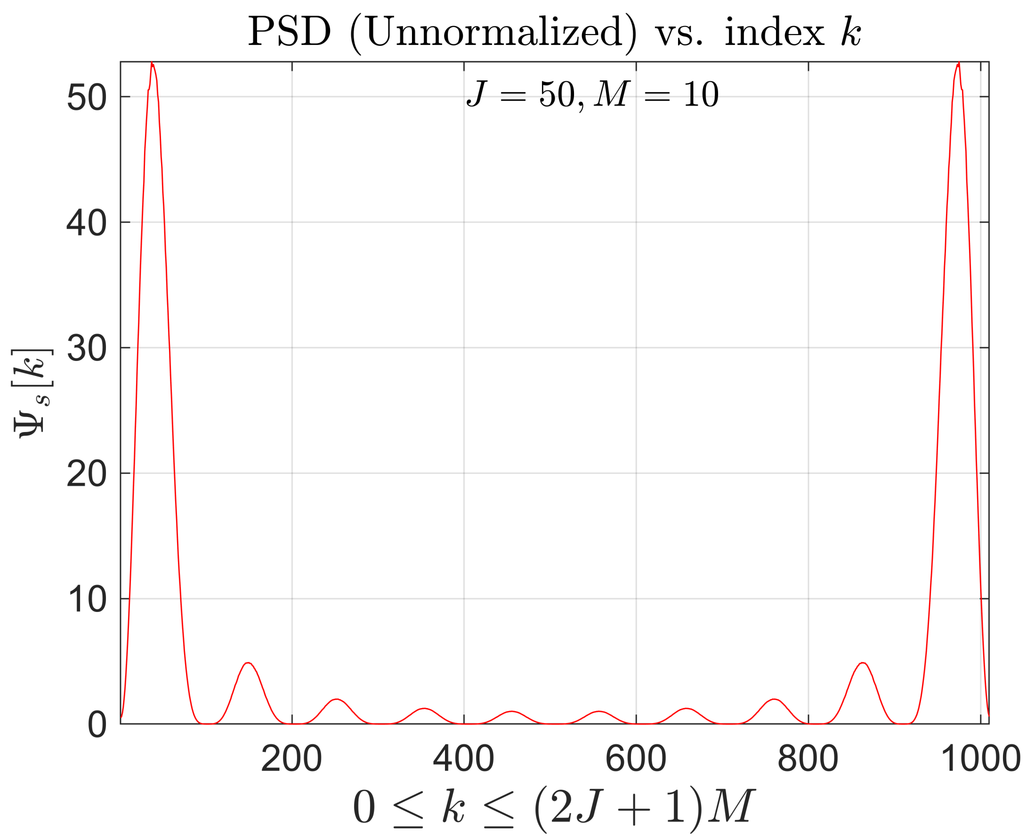
```
Bits = 1×13
     1     1     0     1     1     1     0     0     0     0     0     0     1
```

```
for ii = 1:2*J+1
   if Bits(ii) == 1
       a = a + 1;
       if (mod(a, 2) == 0)
           sNt = [sNt A*ones(1, M)];
       else
           sNt = [sNt -A*ones(1, M)];
       end
   else
```

```
        sNt = [sNt 0*ones(1, M)];
    end
end
stem(sNt);
```

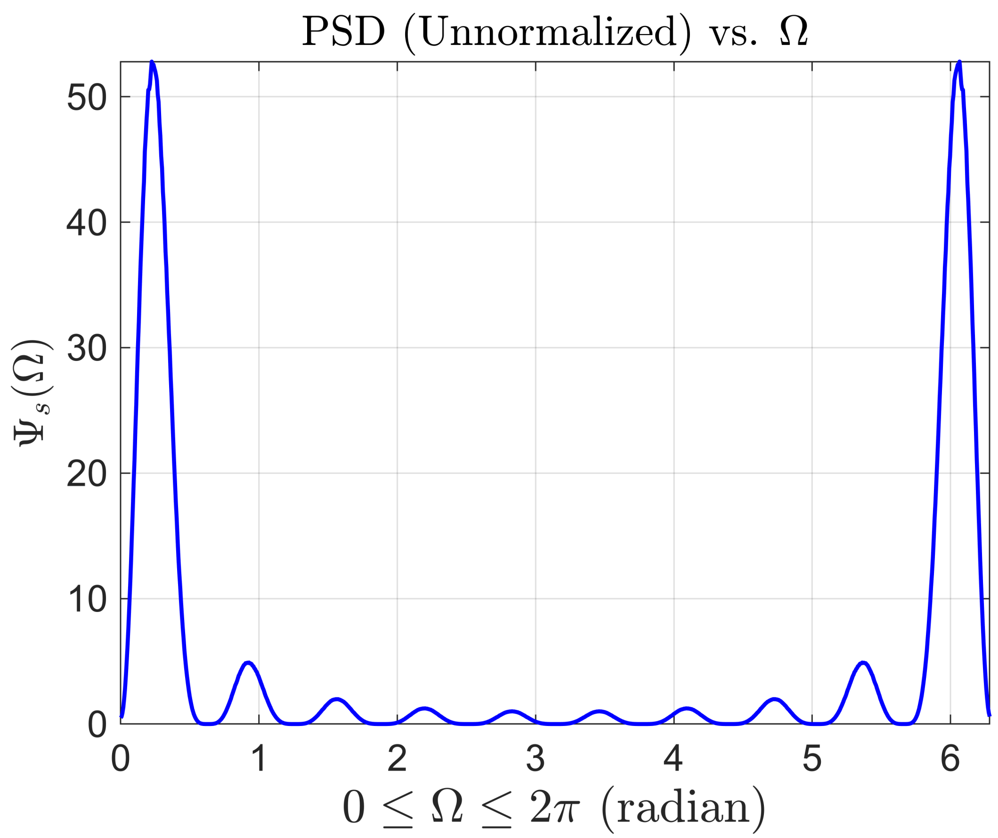
Power Spectral Density Calculations for Bipolar NRZ

```
clc;clear;close all
A = 1;J = 50;M = 10;L = 20000;PSD = 0;a = 0;
T = 1; Ts=T/M; fs=1/Ts;
for jj=1:L
sNt = [];
Bits = (sign(randn(1,2*J+1))+1)/2;
for ii = 1:2*J+1
    if Bits(ii) == 1
        a = a + 1;
        if (mod(a, 2) == 0)
            sNt = [sNt A*ones(1, M)];
        else
            sNt = [sNt -A*ones(1, M)];
        end
    else
        sNt = [sNt 0*ones(1, M)];
    end
end
    PSD = PSD + abs(fft(sNt)).^2;
end
PSD = PSD/L/(2*J+1)/T;
LPSD = length(PSD);
figure(1); plot(PSD,'r-')
set(gca,'FontSize',14);axis tight;grid on;
xlabel('$0\le k\le (2J+1)M$','FontSize',18,'interpreter','latex')
title('PSD (Unnormalized) vs. index $k$','FontSize',16,'interpreter','latex')
ylabel('$\Psi_s[k]$','FontSize',16,'interpreter','latex');
h=text(400,50,['$J = ' num2str(J) ', M = ' num2str(M) '$']);
set(h,'fontsize',14,'interpreter','latex');drawnow
```



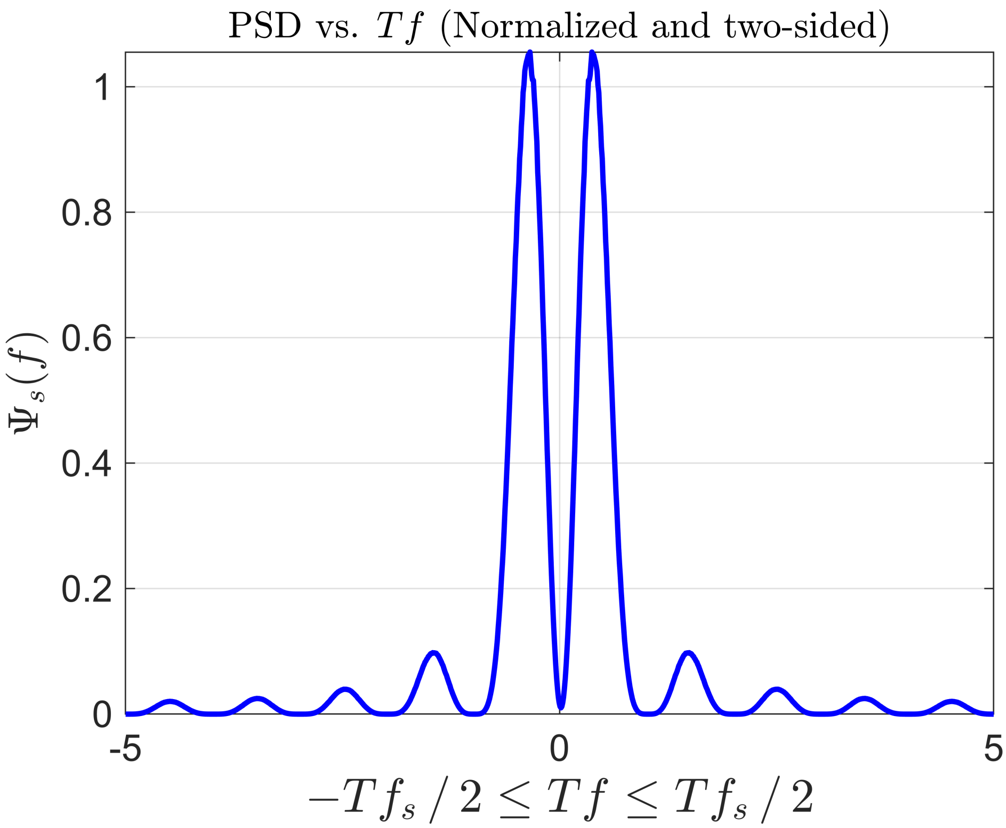
Power Spectral Density (Unnormalized) vs Omega

```
 DTFT_xaxis = linspace(0,2*pi,LPSD);
figure(2);plot(DTFT_xaxis,PSD,'b-','linewidth',1.5)
set(gca,'FontSize',14);axis tight; grid on;
xlabel('$0 \le \Omega\le 2\pi$ (radian)','FontSize',18,'interpreter','latex')
title('PSD (Unnormalized) vs. $\Omega$','FontSize',16,'interpreter','latex')
ylabel('$\Psi_s(\Omega)$','FontSize',16,'interpreter','latex');drawnow
```



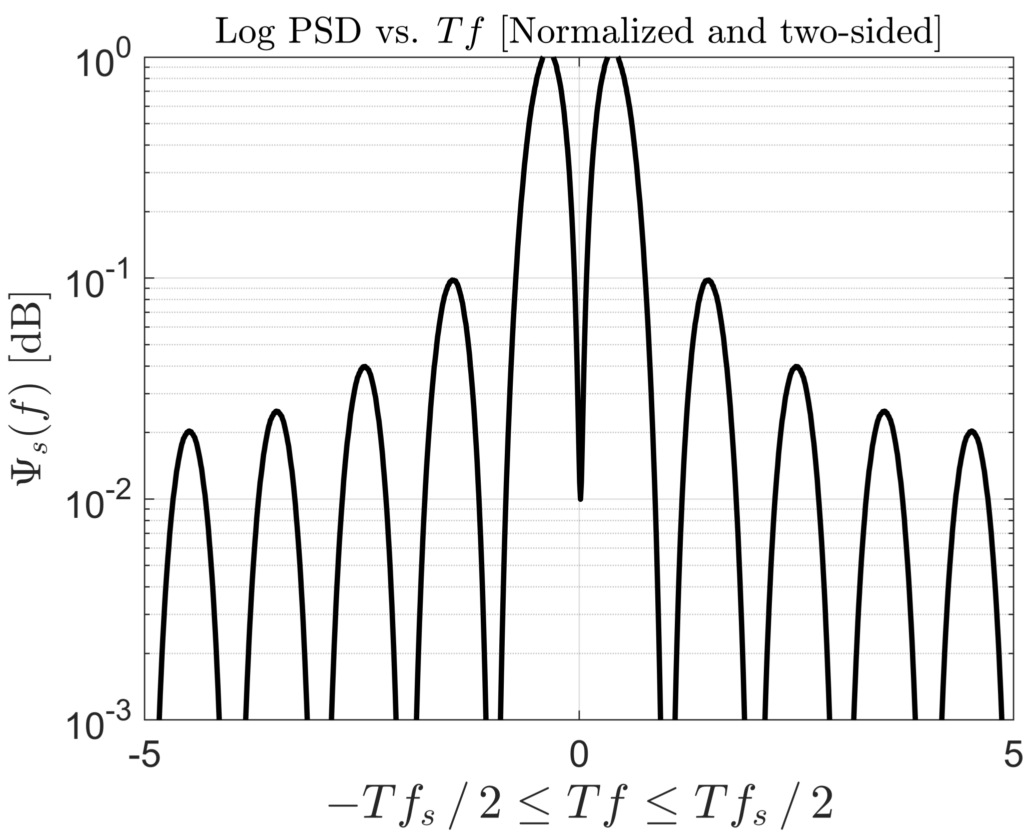
PSD vs T*f* (normalized and two-sided)

```
 fT = linspace(-T*fs/2,T*fs/2,LPSD); % normalized freq axis
Area = sum(PSD)*(fT(2)-fT(1));
 CTFT_PSD = PSD/Area; % normalized PSD
 CTFT_PSD = [fliplr(CTFT_PSD(end:-1:floor(LPSD/2)))...
    CTFT_PSD(1:floor(LPSD/2)-1)];
 figure(3);plot(fT,CTFT_PSD,'b-','linewidth',2)
 set(gca,'FontSize',14);axis tight; grid on;
 xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$','FontSize',18,'interpreter','latex')
 ylabel('$\Psi_s(f)$','FontSize',16,'interpreter','latex');
 xlim([-fs/2 fs/2]);title('PSD vs. $Tf$ (Normalized and two-sided)',...
 'FontSize',14,'interpreter','latex');drawnow
```



Log PSD vs T*f* (Normalized as well as Two-sided)

```
figure(4);semilogy(fT,CTFT_PSD,'k-','linewidth',2);
set(gca,'FontSize',14);axis tight; grid on;
xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$','FontSize',18,'interpreter','latex')
ylabel('$\Psi_s(f)$ [dB]','FontSize',16,'interpreter','latex');ylim([1e-3
1]);
xlim(T*[-fs/2 fs/2]);title('Log PSD vs. $Tf$ [Normalized and two-sided]',...
'FontSize',14,'interpreter','latex');drawnow
```



Task 5

Checking and verifying for the Line coding of Bipolar RZ

```
clc;clear;close all
A = 1;J = 6;M = 4;a = 0; T = 1;
sNt = [];
Bits = (sign(randn(1,2*J+1))+1)/2
```

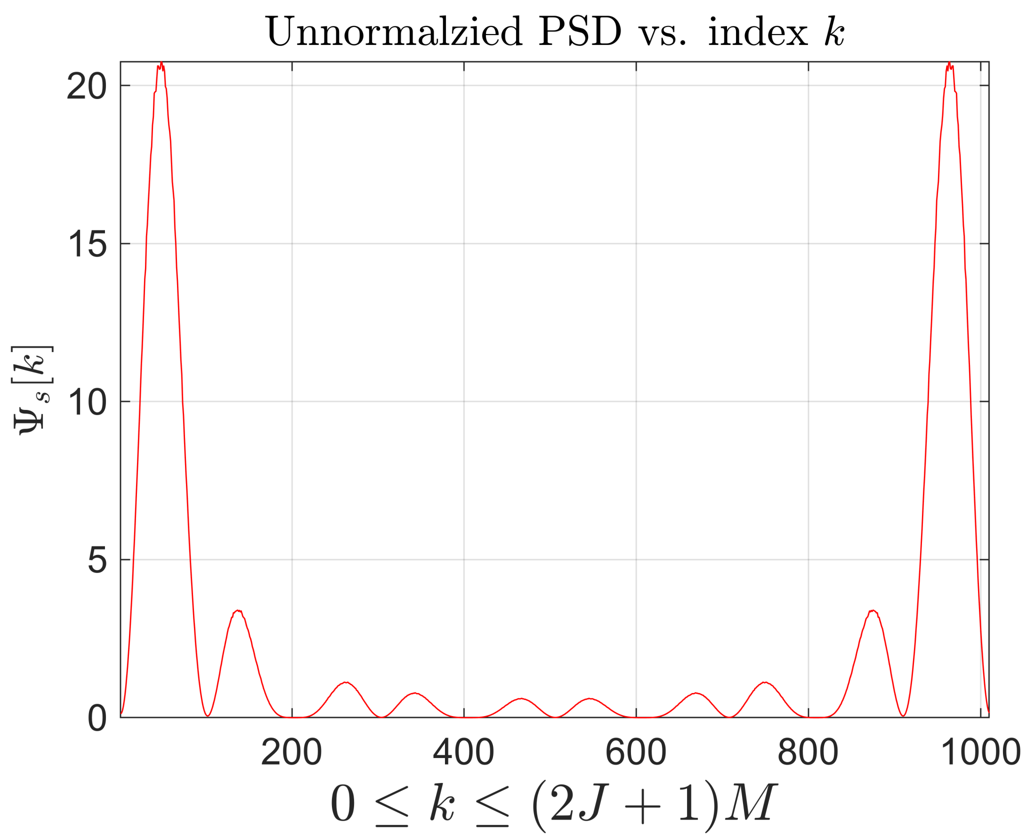
```
Bits = 1×13
     0     1     1     1     0     0     0     0     1     1     1     1     0
```

```
for ii = 1:2*J+1
   if Bits(ii) == 1
       a = a + 1;
       if (mod(a, 2) == 0)
           sNt = [sNt A*ones(1, M/2) A*zeros(1,M/2)];
       else
           sNt = [sNt -A*ones(1, M/2) -A*zeros(1,M/2)];
       end
   else
       sNt = [sNt 0*ones(1, M)];
   end
end
```

```
 stem(sNt);
```

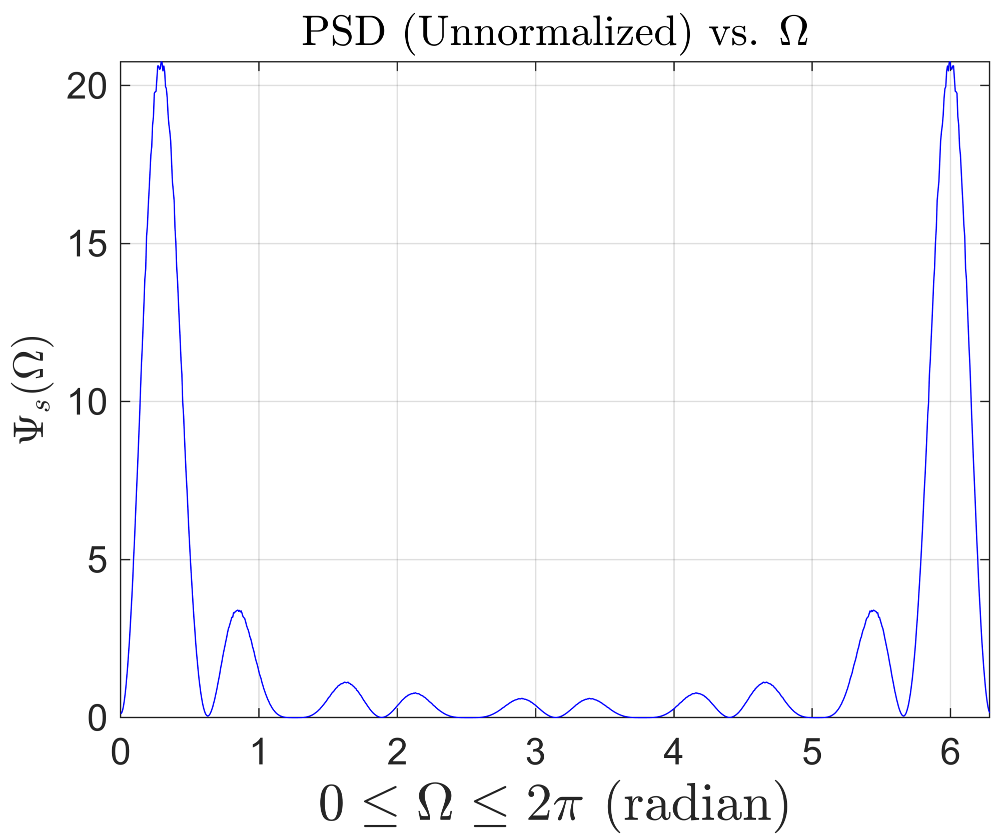
Power Spectral Density Calculations for Bipolar RZ

```
clc;clear;close all
 A = 1;
 J = 50;
 M = 10;
 L = 20000;
 PSD = 0;
 a = 0;
 T = 1; Ts=T/M; fs=1/Ts;
 for jj=1:L
 sNt = [];
 Bits = (sign(randn(1,2*J+1))+1)/2;
 for ii = 1:2*J+1
    if Bits(ii) == 1
        a = a + 1;
        if (mod(a, 2) == 0)
            sNt = [sNt A*ones(1, M/2) A*zeros(1,M/2)];
        else
            sNt = [sNt -A*ones(1, M/2) -A*zeros(1,M/2)];
        end
           else
        sNt = [sNt 0*ones(1, M)];
    end
 end
    PSD = PSD + abs(fft(sNt)).^2;
 end
 PSD = PSD/L/(2*J+1)/T;
LPSD = length(PSD);
 figure(1); plot(PSD,'r-')
 set(gca,'FontSize',14);axis tight;grid on;
 xlabel('$0\le k\le (2J+1)M$','FontSize',20,'interpreter','latex')
 title('Unnormalzied PSD vs. index $k$','FontSize',16,'interpreter','latex')
 ylabel('$\Psi_s[k]$','FontSize',16,'interpreter','latex');
 h=text(400,50,['$J = ' num2str(J) ', M = ' num2str(M) '$']);
 set(h,'fontsize',14,'interpreter','latex');drawnow
```



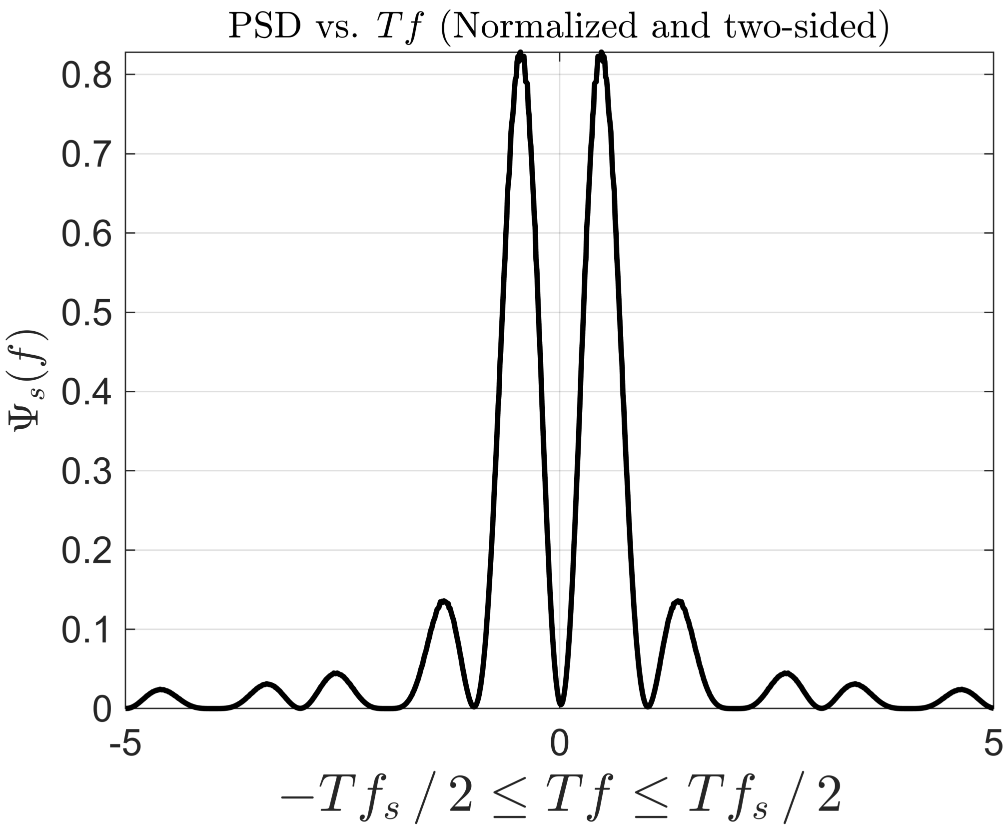
Unnormalized Power Spectral Density vs Omega

```
 DTFT_xaxis = linspace(0,2*pi,LPSD);
figure(2);plot(DTFT_xaxis,PSD,'b-')
set(gca,'FontSize',14);axis tight; grid on;
xlabel('$0 \le \Omega\le 2\pi$ (radian)','FontSize',20,'interpreter','latex')
title('PSD (Unnormalized) vs. $\Omega$','FontSize',16,'interpreter','latex')
ylabel('$\Psi_s(\Omega)$','FontSize',16,'interpreter','latex');drawnow
```



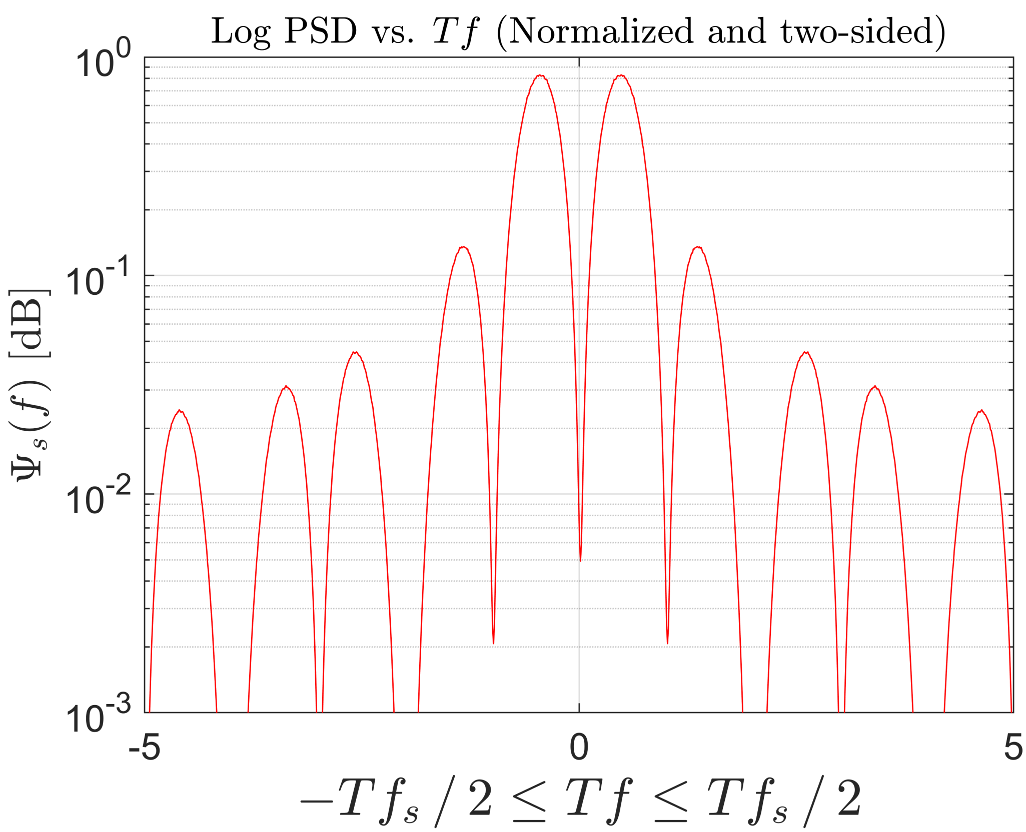
Power Spectral Density vs T*f* (Normalized as well as two sided)

```
 fT = linspace(-T*fs/2,T*fs/2,LPSD); % normalized freq axis
Area = sum(PSD)*(fT(2)-fT(1));
 CTFT_PSD = PSD/Area; % normalized PSD
 CTFT_PSD = [fliplr(CTFT_PSD(end:-1:floor(LPSD/2)))...
    CTFT_PSD(1:floor(LPSD/2)-1)];
 figure(3);plot(fT,CTFT_PSD,'k-','linewidth',2)
 set(gca,'FontSize',14);axis tight; grid on;
 xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/ \,2$','FontSize',20,'interpreter','latex')
 ylabel('$\Psi_s(f)$','FontSize',16,'interpreter','latex');
 xlim([-fs/2 fs/2]);title('PSD vs. $Tf$ (Normalized and two-sided)',...
 'FontSize',14,'interpreter','latex');drawnow
```



Log Power Spectral Density vs T*f* (Normalized and two-sided)

```
figure(4);semilogy(fT,CTFT_PSD,'r-');
set(gca,'FontSize',14);axis tight; grid on;
xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/ \,2$','FontSize',20,'interpreter','latex')
ylabel('$\Psi_s(f)$ [dB]','FontSize',16,'interpreter','latex');ylim([1e-3
1]);
xlim(T*[-fs/2 fs/2]);title('Log PSD vs. $Tf$ (Normalized and two-sided)',...
'FontSize',14,'interpreter','latex');drawnow
```

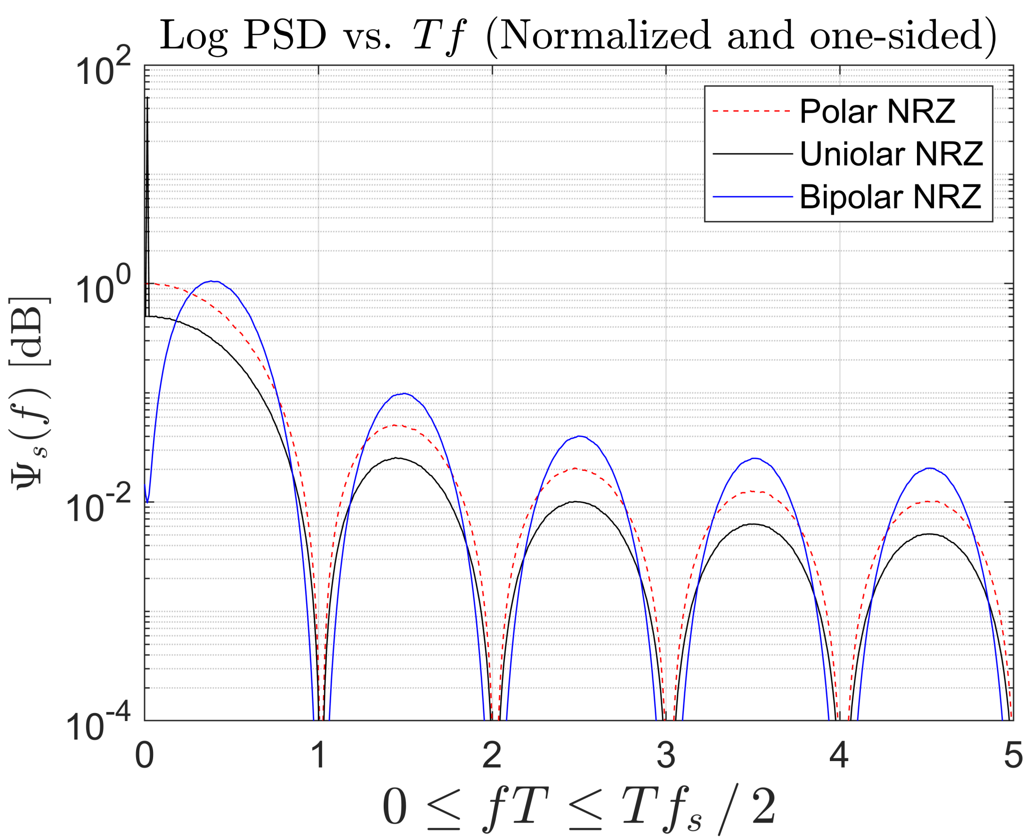


Task 6

```
%Syed Asghar Abbas Zaidi
clc;clear;close all
A = 1;J = 50;M = 10;L = 20000;
PSD_1 = 0;PSD_2 = 0;PSD_3 = 0;a = 0;
T = 1; Ts=T/M; fs=1/Ts;
%polar, and Unipolar and Bipolar NRZ
for jj=1:L
sNt_1 = [];sNt_2 = [];sNt_3 = [];
Bits_1 = (sign(randn(1,2*J+1))+1)/2;
Bits_2 = (sign(randn(1,2*J+1))+1)/2;
Bits_3 = (sign(randn(1,2*J+1))+1)/2;
   for ii=1:2*J+1
       if Bits_1(ii)==1
           sNt_1=[A*ones(1,M) sNt_1];
else
           sNt_1=[-A*ones(1,M) sNt_1];
end
end
for ii=1:2*J+1
```

```
 if Bits_2(ii)==1
            sNt_2=[sNt_2 A*ones(1,M)];
 else
             sNt_2=[sNt_2 -A*zeros(1,M)];
 end
 end
 for ii = 1:2*J+1
 if Bits_3(ii) == 1
            a = a + 1;
 if (mod(a, 2) == 0)
                sNt_3 = [A*ones(1, M) sNt_3];
 else
                sNt_3 = [-A*ones(1, M) sNt_3];
 end
 else
            sNt_3 = [0*ones(1, M) sNt_3];
 end
 end
    PSD_1 = PSD_1 + abs(fft(sNt_1)).^2;
    PSD_2 = PSD_2 + abs(fft(sNt_2)).^2;
    PSD_3 = PSD_3 + abs(fft(sNt_3)).^2;
 end
 PSD_1 = PSD_1/L/(2*J+1)/T;
LPSD_1 = length(PSD_1);
 fT_1 = linspace(-T*fs/2,T*fs/2,LPSD_1); % normalized freq axis
Area_1 = sum(PSD_1)*(fT_1(2)-fT_1(1));
 CTFT_PSD_1 = PSD_1/Area_1; % normalized PSD
 CTFT_PSD_1 = [fliplr(CTFT_PSD_1(end:-1:floor(LPSD_1/2)))...
 CTFT_PSD_1(1:floor(LPSD_1/2)-1)];
 PSD_2 = PSD_2/L/(2*J+1)/T;
LPSD_2 = length(PSD_2);
 fT_2 = linspace(-T*fs/2,T*fs/2,LPSD_2); % normalized freq axis
Area_2 = sum(PSD_2)*(fT_2(2)-fT_2(1));
 CTFT_PSD_2 = PSD_2/Area_2; % normalized PSD
 CTFT_PSD_2 = [fliplr(CTFT_PSD_2(end:-1:floor(LPSD_2/2)))...
 CTFT_PSD_2(1:floor(LPSD_2/2)-1)];
 PSD_3 = PSD_3/L/(2*J+1)/T;
LPSD_3 = length(PSD_3);
 fT_3 = linspace(-T*fs/2,T*fs/2,LPSD_3); % normalized freq axis
Area_3 = sum(PSD_3)*(fT_3(2)-fT_3(1));
 CTFT_PSD_3 = PSD_3/Area_3; % normalized PSD
 CTFT_PSD_3 = [fliplr(CTFT_PSD_3(end:-1:floor(LPSD_3/2)))...
 CTFT_PSD_3(1:floor(LPSD_3/2)-1)];
 % Plot polar nrz
 semilogy(fT_1, CTFT_PSD_1, 'r--');
 hold on; %
 % Plot unipolar nrz
 semilogy(fT_2, CTFT_PSD_2, 'k-');
 % Plot bipolar nrz
 semilogy(fT_3, CTFT_PSD_3, 'b-');
```

```
 set(gca,'FontSize',14); % Set font size for the current axis
 axis tight; grid on;
 xlabel('$0 \le fT\le Tf_s\,/\,2$', 'FontSize', 20, 'interpreter', 'latex')
 ylabel('$\Psi_s(f)$ [dB]', 'FontSize', 16, 'interpreter', 'latex');
 ylim([1e-4 1e2]);
 % Use xlim based on the range of T, fs, as per your requirements
 xlim(T * [0 fs/2]);
 title('Log PSD vs. $Tf$ (Normalized and one-sided)', 'FontSize', 16,
'interpreter', 'latex');
 legend('Polar NRZ', 'Uniolar NRZ', 'Bipolar NRZ'); % Add legend if needed
 hold off; % Release the hold on the plot
```



Task 7

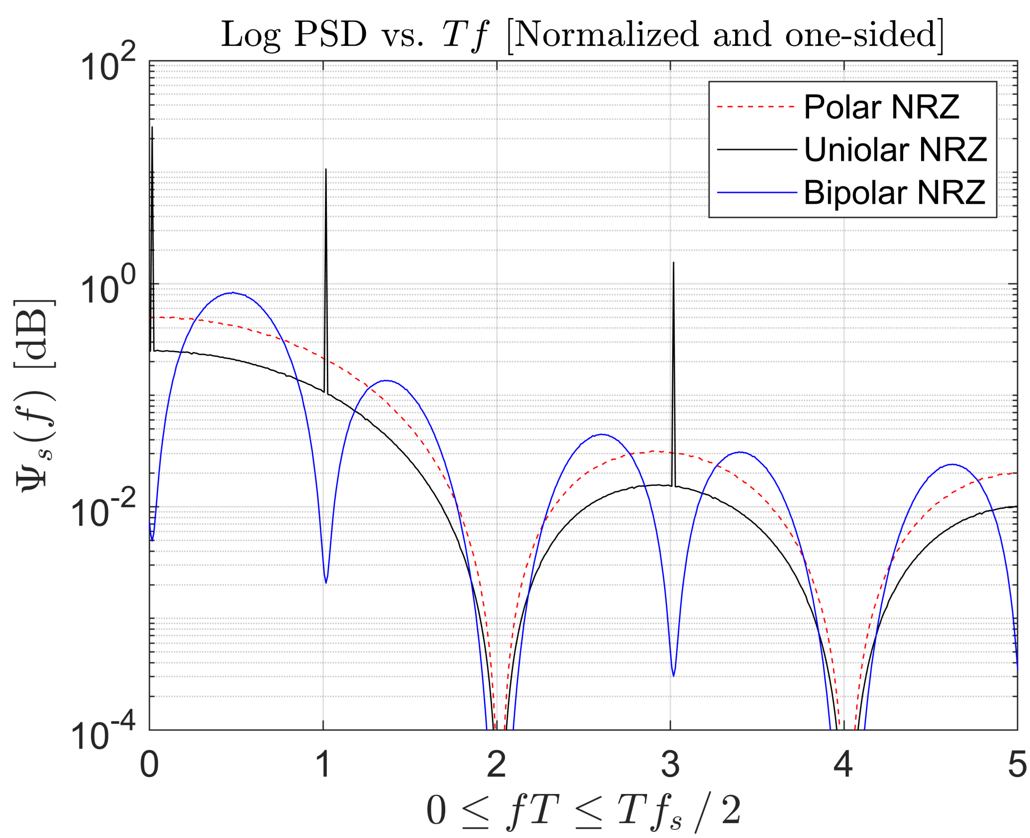
```
clc;clear;close all
 A = 1;J = 50;M = 10;L = 20000;
 PSD_1 = 0;PSD_2 = 0;PSD_3 = 0;
 a = 0;T = 1; Ts=T/M; fs=1/Ts;
 %polar, unipolar, bipolar RZ
 for jj=1:L
 sNt_1 = [];sNt_2 = [];sNt_3 = [];
 Bits_1 = (sign(randn(1,2*J+1))+1)/2;Bits_2 = (sign(randn(1,2*J+1))+1)/2;Bits_3 =
(sign(randn(1,2*J+1))+1)/2;
    for ii=1:2*J+1
```

```
 if Bits_1(ii)==1
            sNt_1=[sNt_1 A*ones(1,M/2) A*zeros(1,M/2)];
 else
            sNt_1=[sNt_1 -A*ones(1,M/2) -A*zeros(1,M/2)];
 end
 end
 for ii=1:2*J+1
 if Bits_2(ii)==1
            sNt_2=[sNt_2 A*ones(1,M/2) A*zeros(1,M/2)];
 else
             sNt_2=[sNt_2 -A*zeros(1,M/2) -A*zeros(1,M/2)];
 end
 end
 for ii = 1:2*J+1
 if Bits_3(ii) == 1
            a = a + 1;
 if (mod(a, 2) == 0)
                sNt_3 = [A*ones(1, M/2) A*zeros(1,M/2) sNt_3];
 else
                sNt_3 = [-A*ones(1, M/2) A*zeros(1,M/2) sNt_3];
 end
 else
            sNt_3 = [0*ones(1, M) sNt_3];
 end
 end
    PSD_1 = PSD_1 + abs(fft(sNt_1)).^2;
    PSD_2 = PSD_2 + abs(fft(sNt_2)).^2;
    PSD_3 = PSD_3 + abs(fft(sNt_3)).^2;
 end
 PSD_1 = PSD_1/L/(2*J+1)/T;
LPSD_1 = length(PSD_1);
 fT_1 = linspace(-T*fs/2,T*fs/2,LPSD_1); % normalized freq axis
Area_1 = sum(PSD_1)*(fT_1(2)-fT_1(1));
 CTFT_PSD_1 = PSD_1/Area_1; % normalized PSD
 CTFT_PSD_1 = [fliplr(CTFT_PSD_1(end:-1:floor(LPSD_1/2)))...
 CTFT_PSD_1(1:floor(LPSD_1/2)-1)];
 PSD_2 = PSD_2/L/(2*J+1)/T;
LPSD_2 = length(PSD_2);
 fT_2 = linspace(-T*fs/2,T*fs/2,LPSD_2); % normalized freq axis
Area_2 = sum(PSD_2)*(fT_2(2)-fT_2(1));
 CTFT_PSD_2 = PSD_2/Area_2; % normalized PSD
 CTFT_PSD_2 = [fliplr(CTFT_PSD_2(end:-1:floor(LPSD_2/2)))...
 CTFT_PSD_2(1:floor(LPSD_2/2)-1)];
 30
```

```
ans = 30
```

```
 PSD_3 = PSD_3/L/(2*J+1)/T;
 LPSD_3 = length(PSD_3);
 fT_3 = linspace(-T*fs/2,T*fs/2,LPSD_3); % normalized freq axis
Area_3 = sum(PSD_3)*(fT_3(2)-fT_3(1));
```

```
CTFT_PSD_3 = PSD_3/Area_3; % normalized PSD
CTFT_PSD_3 = [fliplr(CTFT_PSD_3(end:-1:floor(LPSD_3/2)))...
CTFT_PSD_3(1:floor(LPSD_3/2)-1)];
% Plot polar nrz
semilogy(fT_1, CTFT_PSD_1, 'r--');
hold on; %
% Plot unipolar nrz
semilogy(fT_2, CTFT_PSD_2, 'k-');
% Plot bipolar nrz
semilogy(fT_3, CTFT_PSD_3, 'b-');
set(gca,'FontSize',14); % Set font size for the current axis
axis tight; grid on;
xlabel('$0 \le fT\le Tf_s\,/\,2$', 'FontSize', 16, 'interpreter', 'latex')
ylabel('$\Psi_s(f)$ [dB]', 'FontSize', 16, 'interpreter', 'latex');
ylim([1e-4 1e2]);
xlim(T * [0 fs/2]);
title('Log PSD vs. $Tf$ [Normalized and one-sided]', 'FontSize', 14,
'interpreter', 'latex');
legend('Polar NRZ', 'Uniolar NRZ', 'Bipolar NRZ');
hold off;
```

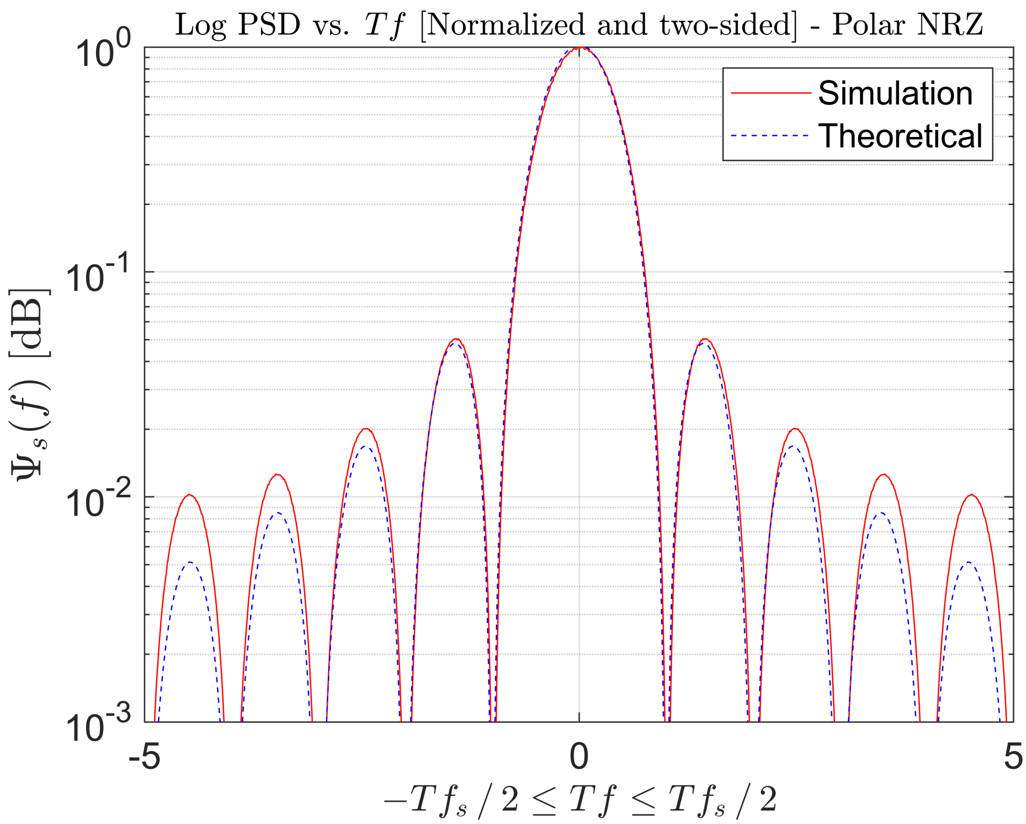


Task 8

Polar NRZ

```
 clc;clear;close all
A = 1;J = 50;M = 10;L = 20000;PSD = 0;
T = 1; Ts=T/M; fs=1/Ts;
for jj=1:L
sNt = [];
Bits = (sign(randn(1,2*J+1))+1)/2;
   for ii=1:2*J+1
       if Bits(ii)==1
           sNt=[A*ones(1,M) sNt];
       else
           sNt=[-A*ones(1,M) sNt];
       end
   end
   PSD = PSD + abs(fft(sNt)).^2;
end
PSD = PSD/L/(2*J+1)/T;
LPSD = length(PSD);
DTFT_xaxis = linspace(0,2*pi,LPSD);
fT = linspace(-T*fs/2,T*fs/2,LPSD); % normalized freq axis
Area = sum(PSD)*(fT(2)-fT(1));
CTFT_PSD = PSD/Area; % normalized PSD
CTFT_PSD = [fliplr(CTFT_PSD(end:-1:floor(LPSD/2)))...
   CTFT_PSD(1:floor(LPSD/2)-1)];
% Polar NRZ - Theoretical PSD
T_theoretical = 1;  % Symbol duration (adjust as needed)
f_theoretical = linspace(-fs/2, fs/2, LPSD);
% Calculate theoretical PSD
PSD_theoretical = (sin(pi*f_theoretical*T_theoretical)).^2 ./
(pi*f_theoretical*T_theoretical).^2;
PSD_theoretical(isnan(PSD_theoretical)) = 1;
Area_theoretical = sum(PSD_theoretical) * (fT(2) - fT(1));
CTFT_PSD_theoretical = PSD_theoretical / Area_theoretical;
figure;
semilogy(fT, CTFT_PSD, 'r-', 'DisplayName', 'Simulated');
hold on;
semilogy(f_theoretical, CTFT_PSD_theoretical, 'b--', 'DisplayName', 'Theoretical');
set(gca, 'FontSize', 14);
axis tight; grid on;
xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$', 'FontSize', 14,
'interpreter','latex');
ylabel('$\Psi_s(f)$ [dB]', 'FontSize', 16, 'interpreter', 'latex');
ylim([1e-3 1]);
xlim(T * [-fs/2 fs/2]);
title('Log PSD vs. $Tf$ [Normalized and two-sided] - Polar NRZ', 'FontSize', 12,
'interpreter', 'latex');
legend('Simulation', 'Theoretical');
```

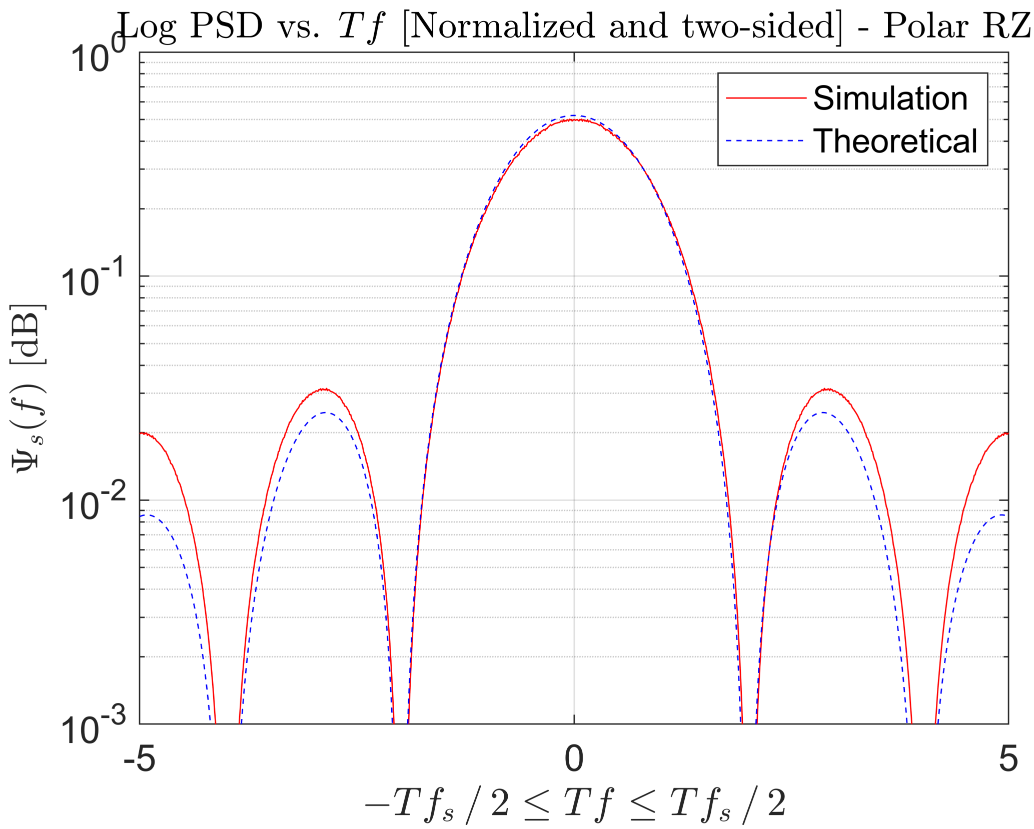
```
hold off;
```



POLAR RZ

```
A = 1;
J = 50;
M = 10;
L = 20000;
PSD = 0;
T = 1; Ts=T/M; fs=1/Ts;
for jj=1:L
sNt = [];
Bits = (sign(randn(1,2*J+1))+1)/2;
   for ii=1:2*J+1
       if Bits(ii)==1
           sNt=[sNt A*ones(1,M/2) A*zeros(1,M/2)];
       else
           sNt=[sNt -A*ones(1,M/2) -A*zeros(1,M/2)];
       end
   end
   PSD = PSD + abs(fft(sNt)).^2;
end
PSD = PSD/L/(2*J+1)/T;
LPSD = length(PSD);
```

```
DTFT_xaxis = linspace(0,2*pi,LPSD);
 fT = linspace(-T*fs/2,T*fs/2,LPSD); % normalized freq axis
Area = sum(PSD)*(fT(2)-fT(1));
 CTFT_PSD = PSD/Area; % normalized PSD
 CTFT_PSD = [fliplr(CTFT_PSD(end:-1:floor(LPSD/2)))...
    CTFT_PSD(1:floor(LPSD/2)-1)];
 % Polar NRZ - Theoretical PSD
 T_theoretical = 1;  % Symbol duration (adjust as needed)
 f_theoretical = linspace(-fs/2, fs/2, LPSD);
 % Calculate theoretical PSD
 PSD_theoretical = (T_theoretical^2/4) * (sinc(f_theoretical*T_theoretical/2)).^2;
 PSD_theoretical(isnan(PSD_theoretical)) = 1;
 % Normalize the theoretical PSD
 Area_theoretical = sum(PSD_theoretical) * (fT(2) - fT(1));
 CTFT_PSD_theoretical = PSD_theoretical / Area_theoretical;
 % Plot the simulated and theoretical PSD
 figure;
 semilogy(fT, CTFT_PSD, 'r-','DisplayName', 'Simulation');
 hold on;
 semilogy(f_theoretical, CTFT_PSD_theoretical, 'b--', 'DisplayName', 'Theoretical');
 set(gca, 'FontSize', 14);
 axis tight; grid on;
 xlabel('$-Tf_s\,/\,2 \le Tf\le Tf_s\,/\,2$', 'FontSize', 15, 'interpreter',
'latex');
 ylabel('$\Psi_s(f)$ [dB]', 'FontSize', 14, 'interpreter', 'latex');
 ylim([1e-3 1]);
 xlim(T * [-fs/2 fs/2]);
 title('Log PSD vs. $Tf$ [Normalized and two-sided] - Polar RZ', 'FontSize', 14,
'interpreter', 'latex');
 legend('Simulation', 'Theoretical');
 hold off;
```


Log PSD vs. Tf [Normalized and two-sided] - Polar RZ
Simulation
Theoretical
$\Psi_s(f)$ [dB]
10^0
10^{-1}
10^{-2}
10^{-3}
-5
0
5
$-Tf_s\,/\,2 \le Tf \le Tf_s\,/\,2$