

---

```

clear;
clc;
% Generate a random binary sequence
L = 8; % Determine the length of binary sequence
bits = round(rand(1,L));

Ts = 1; % Symbol period
N_sample = 100; % Sample points for single symbol
dt = Ts / N_sample; % Interval of sampling for single symbol
t = 0:dt:(L * N_sample - 1)*dt; % Time during for whole symbol
unipolar_RZ = [ones(1, N_sample/2), zeros(1, N_sample/2)];
polar_NRZ = ones(1, N_sample);

% Problem 1:Plot the unipolar RZ,the bipolar NRZ waveform of the
sequence
unipolar_RZ_wave = []; % Record the digital voltage by unipolar RZ
polar_NRZ_wave = []; % Record the digital voltage by polar NRZ
for i=1:L % For each symbol
    if bits(i) == 1 % If symbol is 1, turn to corresponding waveform
        unipolar_RZ_wave = [unipolar_RZ_wave, unipolar_RZ];
        polar_NRZ_wave = [polar_NRZ_wave, polar_NRZ];
    else % If symbol is 0, turn to 0
        unipolar_RZ_wave = [unipolar_RZ_wave, zeros(1, N_sample)];
        polar_NRZ_wave = [polar_NRZ_wave, -1*ones(1, N_sample)];
    end
end
figure(1)
set(gcf,'position', [250 200 1000 800]);
subplot(211) % Plot the unipolar RZ waveform
plot(t, unipolar_RZ_wave)
grid on;
title("Unipolar RZ waveform");
ylim([-0.25, 1.25]);
xlabel("time (s)")
ylabel("voltage (V)")
subplot(212) % Plot the polar NRZ waveform
plot(t, polar_NRZ_wave)
grid on;
title("Polar NRZ waveform");
ylim([-1.25, 1.25]);
xlabel("time (s)")
ylabel("voltage (V)")

% Problem 2:Write a function to generate the differential code and
HDB3 code.
% Verify differential code function
different_code = Differential_code(bits, N_sample);
figure(2)
set(gcf,'position', [250 200 1000 800]);
subplot(211) % Plot the differential waveform
plot(t, different_code)
grid on;

```

---

---

```

title("Differential code");
ylim([-1.25, 1.25]);
xlabel("time (s)")
ylabel("voltage (V)")
% Verify HDB3 code function
bits = [1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 1];
hdb3_code = HDB3_code(bits, N_sample);
L = length(bits);
t = 0:dt:(L * N_sample - 1)*dt; % Time during for whole symbol
subplot(212) % Plot the HDB3 waveform
plot(t, hdb3_code)
grid on;
title("HDB3 code");
ylim([-1.25, 1.25]);
xlabel("time (s)")
ylabel("voltage (V)")

% Problem 3: Find the theoretic power density and try to plot the curve
% Determine the symbol transmit rate fb and symbol duration Tb
fb = 1;
Tb = 1/fb;
% Equal probabilities
P = 1/2;
% Determine the G1 and G2
f = -10:0.00001:10;
G1 = (Tb^2)*(sinc(f.*Tb/2).^2);
G2 = 0;
% Calculate the Pu(f)
Pu = fb*P*(1-P)*abs(G1-G2).^2;
% Calculate the Pv(f)
m = -10:10;
G1_m = (Tb^2)*(sinc(m.*fb.*Tb/2).^2);
Pv = abs(fb*(P*G1_m)+(1-P)*G2).^2;
% Plot the theoretic power density
figure(3)
plot(f, Pu)
hold on
stem(m, Pv)
xlabel('Frequency (Hz)')
ylabel('Amplitude')
legend('Pu(f)', 'Pv(f)')
title("The theoretic power density")

AMI
# 1 # 13
-1 0 0 0 0 1 0 0 0 0 0 -1
1 0

# 14 # 18
0 0 0 -1 1

hdb3_code step 1

```

---

---

```

# 1 # 13
      1      0      0      0      2      1      0      0      0      2      1
1      0

# 14 # 18
      0      0      2      1      1

      5      10     16

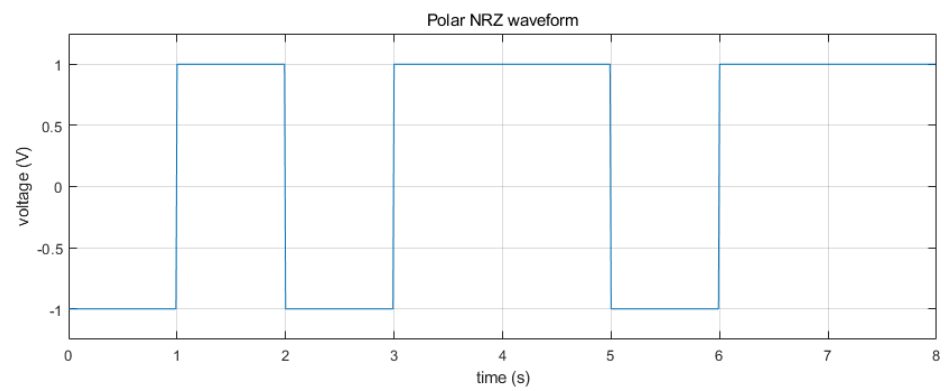
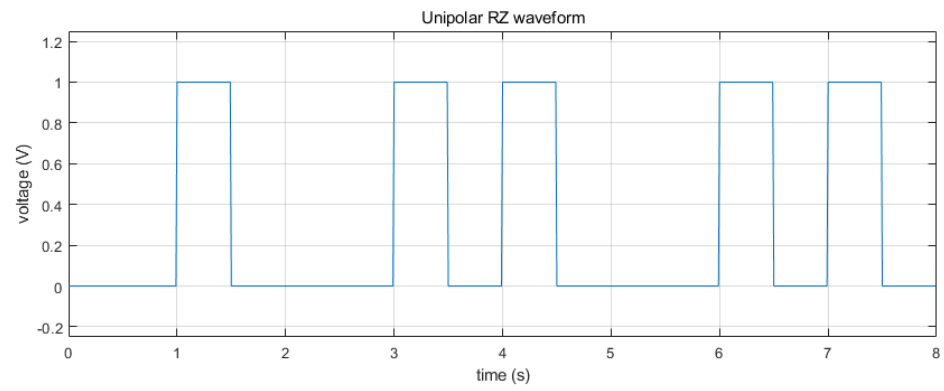
hdb3_code step 2
# 1 # 13
      1      0      0      0      2      1      0      0      0      2      1
1      3

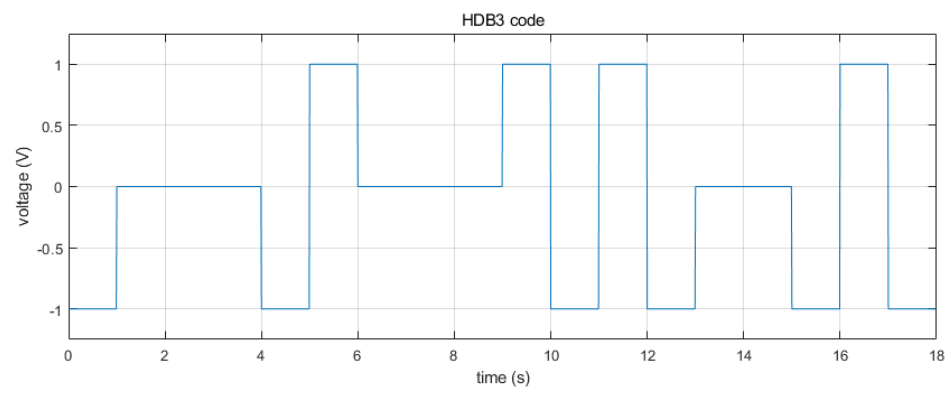
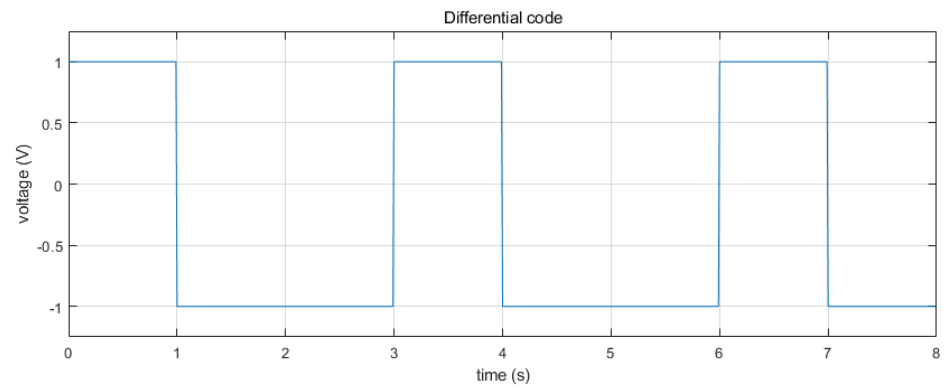
# 14 # 18
      0      0      2      1      1

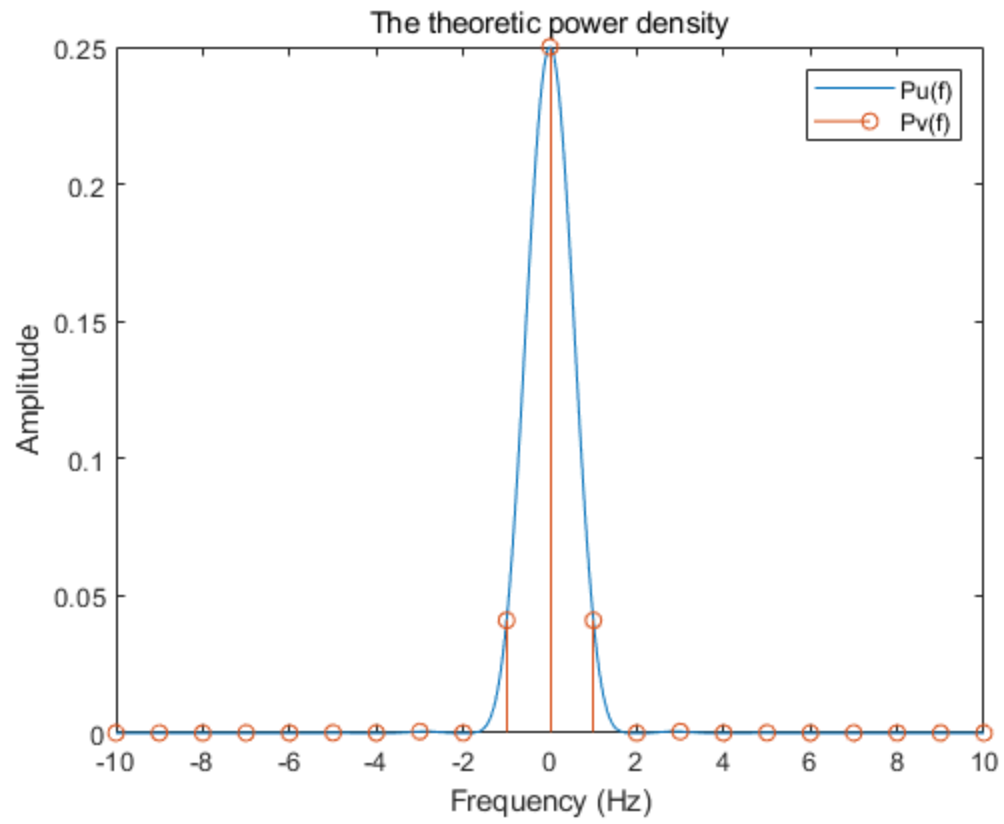
hdb3_code
# 1 # 13
     -1      0      0      0     -1      1      0      0      0      1     -1
1     -1

# 14 # 18
      0      0     -1      1     -1

```







*Published with MATLAB® R2021a*