EE 324 HABERLEŞME SİSTEMLERİ LAB-II

Dr. Sadık YILDIZ

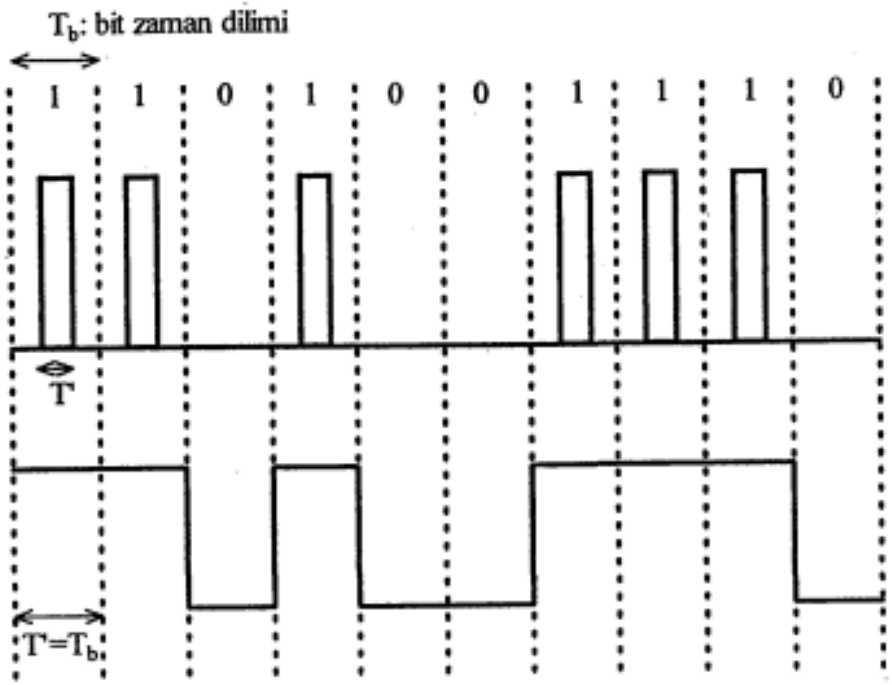
sadikyildiz06@gmail.com

Gazi Üniversitesi
Elektrik Elektronik Mühendisliği
(Teknoloji Fakültesi)

2025– ANKARA

Hat Kodlaması

- Şekil 4.1'de, sayısal bilgi bitlerinin elektriksel darbeler kullanılarak gösterimine örnek verilmektedir.
- İkili iletimde bir bitin iletimi için kullanılabilecek en yüksek süre, *bit zaman dilimi olarak adlandırılır (T_b)*
- *Bit zaman dilimi*, *tipik olarak ikili iletim oranının (r_b)* tersine eşit olmaktadır.
- Örnek olarak, bit oranı r=64 kbit/s olan bir iletim için bit zaman dilimi $T_b = 1/r_b$= 15.625 µsan olarak bulunmaktadır.
- Darbe iletimi için kullanılan süre (T') arttıkça darbenin enerjisi artmakta ve darbenin alıcıda algılanması kolaylaşmaktadır.
- Bu nedenle çoğunlukla *darbe genişliği* üst limit olan *bit (veya sembol) zaman dilimine* eşit alınmaktadır.

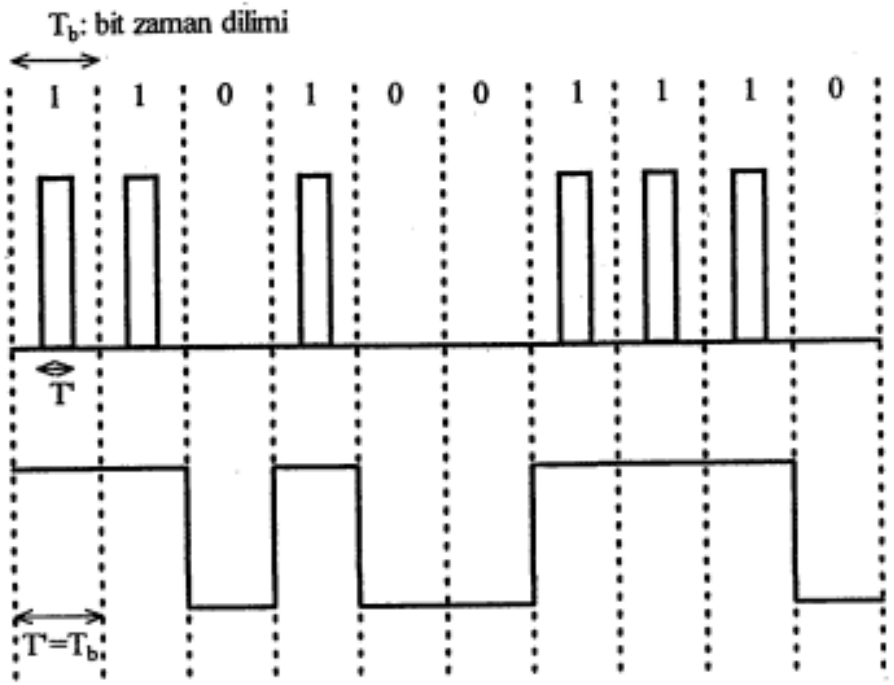


Şekil 4.1. Bitlerin darbelerle gösterimi.

Hat Kodlaması

- *Darbe genişliğinin bit zaman dilimine* eşit olduğu durumda, iletilen elektriksel dalga şekli belirli darbelerin varlığı/yokluğu yerine, *seviyeler arasında geçiş olarak da düşünülebilmektedir.*

- *Sayısal bilginin elektriksel darbeler ile iletimi darbe biçimine göre dört grupta sınıflandırılmaktadır.*
  1. *Sıfıra dönmeyen (NRZ) darbeler*
  2. *Sıfıra dönen (RZ) darbeler*
  3. *Faz Kodlanmış darbeler*
  4. *Çok-seviyeli darbeler*
- *Tabanbant iletiminde, sayısal bilginin iletimi için kullanılacak darbe biçimi, yani hat kodu, haberleşme sisteminin özellik ve gereksinimlerine göre belirlenebilmektedir.*



Şekil 4.1. Bitlerin darbelerle gösterimi.

Hat Kodlaması

- ***DC seviyesinin varlığı:*** *Bazı sistemlerde ortalama değeri sıfırdan farklı işaretler kullanıldığında problemler meydana gelebilmektedir. Bu nedenle, bu tür sistemlerde ortalama değeri sıfır olan darbe biçimleri kullanılmaktadır.*

- ***Güç spektral yoğunluğu:*** *Güç spektral yoğunluğu iletim için kullanılan gücün frekans bandına göre dağılımını göstermektedir. Kullanılması gereken güç miktarı çoğunlukla bant genişliği, hata olasılığı veya sistem karmaşıklığı ile ters orantılı olmaktadır.*

- ***Spektrum kullanımı (bant genişliği):*** *Tabanbant iletimi elektriksel darbeler kullanılarak gerçekleştirildiğinde, darbelerin bant genişliği doğrudan haberleşme sistemi için gerekli bant genişliğini belirlemektedir.*

- ***Bit hata oranı performansı:*** *Haberleşme sisteminin başarımı doğrudan bit hata oranı ile ölçüldüğünden bant genişliği, güç kullanımı ve sistem karmaşıklığı göz önünde bulundurularak bit hata oranının olabildiğince düşük olması istenmektedir.*

Hat Kodlaması

- ***Sembol senkronizasyonu için zamanlama bilgisinin elde edilebilmesi:*** *Sayısal haberleşme sistemlerinde çoğunlukla eşzamanlamaya (senkronizasyona) ihtiyaç duyulmaktadır*

- ***Hata algılama özelliklerine duyulan ihtiyaç:*** *Kullanılan dalga biçimlerine bağlı olarak alıcıya hata algılayabilme özelliği kazandırılabilmektedir. Bu özellik, çoğunlukla daha karmaşık bir sistem yapısı gerektirmektedir. İletim için kullanılan darbe biçimleri sayesinde elde edilen hata algılama özelliğinin en önemli faydası, ek hata kontrol verisine ihtiyaç duyulmamasıdır.*

- ***Saydamlık:*** *İletilen sayısal işarette arka arkaya gelen sembollere bağlı olarak sistem performansının değişmediği sistemler saydam olarak adlandırılmaktadır. Örneğin, arka arkaya gönderilen 0 ve 1 sembollerinin sayısından bağımsız olarak sistem sürekli aynı başarımda çalışmayı sürdürüyorsa sistem saydamdır.*

Hat Kodlaması

1. Tek-kutuplu İşaretleşme
2. Kutuplu İşaretleşme.
3. Faz-kodlanmış İşaretleşme
4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi).
5. Yüksek Yoğunluklu Çift-Kutuplu İşaretleşme (HDBn).
6. Kod İm Değişimi İşaretleşme (CMI).
7. İkili Sembollerin Üçlü Kodlanması (nBmT).
8. Çok-Seviyeli İşaretleşme.

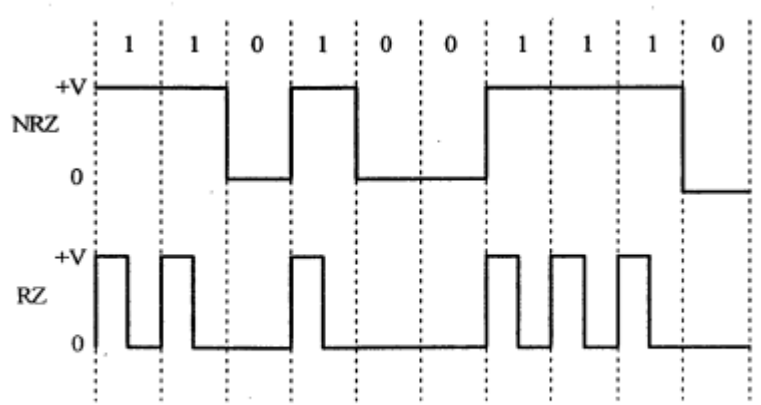
Hat Kodlaması

1. Tek-kutuplu İşaretleşme

- Tek-kutuplu işaretleşme kullanıldığında, ikili sembollerden birisi (genelde 1 biti) için bir darbe iletilmekte, diğer sembol (genelde 0 biti) için ise darbe iletilmemektedir.
- İletim için kullanılan darbe *p(t)* ile gösterildiği taktirde, tek-kutuplu işaretleşmede 1 biti için *p(t)* gönderilirken, 0 biti için *p(t)* gönderilmemektedir.
- 0 biti için sıfır seviyesinden iletim yapıldığı düşünülebilmektedir.

1. Tek-kutuplu İşaretleşme

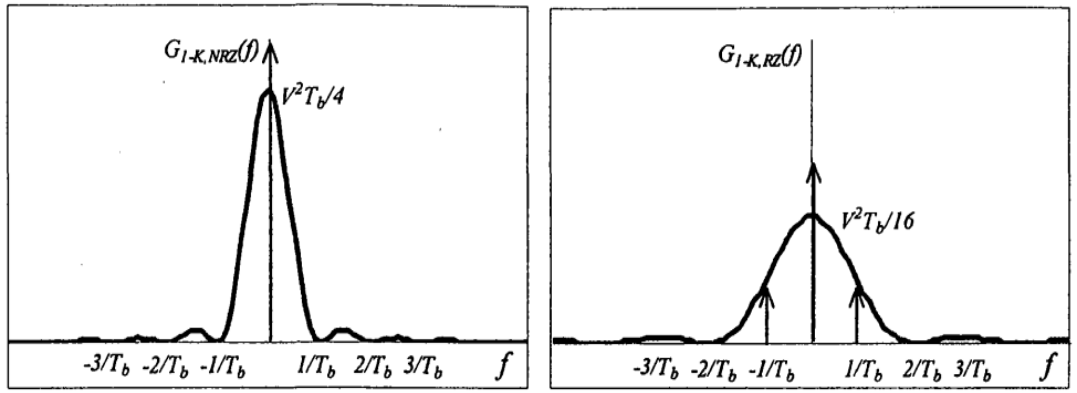
- Tabanbant *başla-dur anahtarlaması (OOK)* olarak da adlandırılan tek-kutuplu işaretleşmenin, Şekil 4.2'de gösterilen, *sıfıra dönmeyen (NRZ)* ve sıfıra dönen (RZ) çeşitleri bulunmaktadır.
- Şekil 4.2'de gösterilen RZ darbeleri sadece zaman diliminin ilk yarısında pozitif gerilimde olup ikinci yarısında sıfıra dönmektedir, ve dolayısıyla bu darbelerin *doluluk-boşluk oranı %50 olmaktadır (Duty cycle)*.
- RZ darbesinin *pozitif gerilimde olduğu süresi* τ ve bit-zaman dilimi T_b, ile gösterildiğinde *doluluk-boşluk oranı* τ/T_b olarak bulunmaktadır.
- Doluluk-boşluk oranı doğrudan iletim için *gerekli bant-genişliğini*, *kullanılan ortalama gücü* ve *hata olasılığını* etkilediğinden uygulamada farklı doluluk-boşluk oranları kullanılabilmektedir.



Şekil 4.2. Tek-kutuplu işaretleşme.

1. Tek-kutuplu İşaretleşme

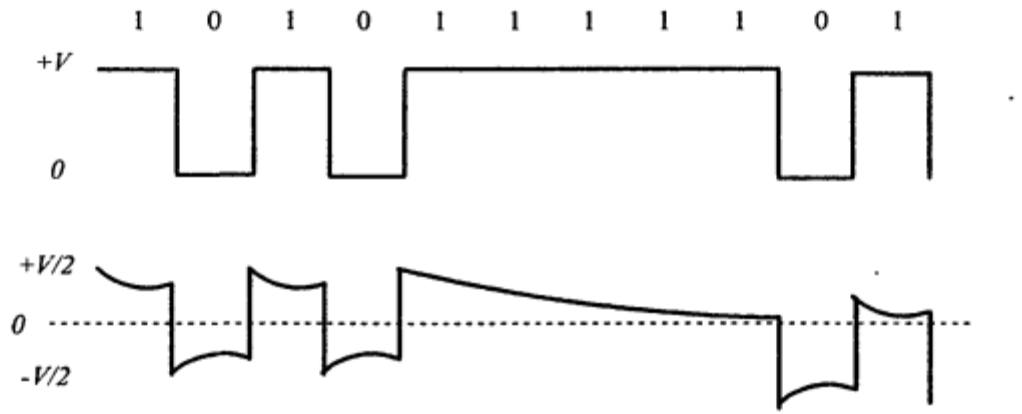
- Şekil 4.3'de tek-kutuplu NRZ ve RZ işaretleşmesi için elde edilen güç spektral yoğunlukları gösterilmektedir.
- Bu güç spektral yoğunlukları incelendiğinde, doluluk-boşluk oranı %50 olan tek-kutuplu sıfıra dönen RZ işaretleşmenin, sıfıra dönmeyen NRZ işaretleşmeye *oranla iki kat fazla bant genişliği kapsadığı görülmektedir*.
- RZ işaretleşmede zaman uzayında darbenin genişliği azaldığından, frekans uzayında spektrumun genişlemesi beklenmektedir.
- Ayrıca, RZ spektrumunda sembol oranı f_b'nin tek katlarında dürtü şeklinde spektral bileşenler oluşmaktadır.
- Dürtü şeklinde oluşan spektral bileşenlerinin frekansları doğrudan sembol oranını verdiğinden, alıcıda bu bileşenlerin ayrıştırılmasıyla işaretten *sembol oranı* yani *zamanlama bilgisi* çıkartılabilmektedir.
- Bu sayede harici bir sisteme ihtiyaç olmadan, doğrudan bilgi darbeleri kullanılarak *eşzamanlılığın sağlanması* mümkün olmaktadır.



Şekil 4.3. Tek-kutuplu NRZ ve RZ işaretleşmesi için güç spektral yoğunlukları.

1. Tek-kutuplu İşaretleşme

- Tek-kutuplu işaretlerde, güç spektral yoğunluklarının f=0 Hz frekansına karşılık gelen, de seviyesi bulunmaktadır.
- Bu nedenle tek-kutuplu işaretlerin, trafo veya kapasitör bağlaşımlı (ac bağlaşımlı) yineleyicilerin bulunduğu hatlardan iletimi durumunda darbe şekillerinde bozulmalar ve genlik seviyelerinde düşmeler meydana gelmektedir.
- Şekil 4.4'de tek-kutuplu bir işaretin ortalama değerinin ac-bağlaşım nedeniyle tutulması durumunda oluşan duruma bir örnek gösterilmektedir.
- Arka arkaya gelen 0 veya 1 sembollerinin sayısına bağlı olarak işaretin kısa-süreli ortalama değeri değişeceğinden darbelerin genliği farklı seviyelere ötelenebilmektedir.
- Bu nedenle tek-kutuplu işaretler ac-bağlaşımlı hatlar üzerinden iletim için uygun olmamaktadır.



Şekil 4.4. Tek-kutuplu NRZ işaretleşmede AC-bağlaşım sonucu bozulma etkisi.

1. Tek-kutuplu İşaretleşme

Örnek 4.2. Bir ikili bilgi işareti [10100011] değerlerini almaktadır. Bu bilgi işaretine göre tek-kutuplu NRZ ve doluluk-boşluk oranı %50 olan tek-kutuplu RZ işaretlerini oluşturarak gösterecek bir Matlab programı yazınız.

main.m | tekkutuplu.m

```
% ana Program
Fs = 10000; % Tek-kutuplu kodlanmış (tabanbant modüle edilmiş)
            %işaretin örnekleme frekansı
Fd = 1000;  % İkili işaretin örnekleme frekansı (modülasyon öncesi).
           % Bu değer ikili işaretteki bir bitin programda kaç örnek ile
           % gösterildiğini tanımlamaktadır. Bu nedenle Tb = 1 / Fd olacaktır.
           % Bu değer ikili iletim oranına eşit olduğundan rb=1 kbit/s dir.
           % Fs/Fd tamsayı olmalıdır
b=[1 0 1 0 0 0 1 1]; % İkili bit dizini

bnrz=tekkutuplu(b,Fd,Fs, 'nrz'); % Tek-kutuplu NRZ işareti elde et
                                 % Not: tekkutuplu(.) fonksiyonu aşağıdadır
                                 % NRZ işareti çizdir
figure;
plot (bnrz,'Color', [0 0 1], 'LineWidth', 2);
title('tek-kutuplu NRZ');
figure;
brz=tekkutuplu(b,Fd,Fs,'rz'); % Tek-kutuplu RZ işareti elde et
plot (brz, 'Color', [1 0 0], 'LineWidth', 2); % RZ işareti çizdir
title('tek-kutuplu RZ');
```

1. Tek-kutuplu İşaretleşme

Örnek 4.2

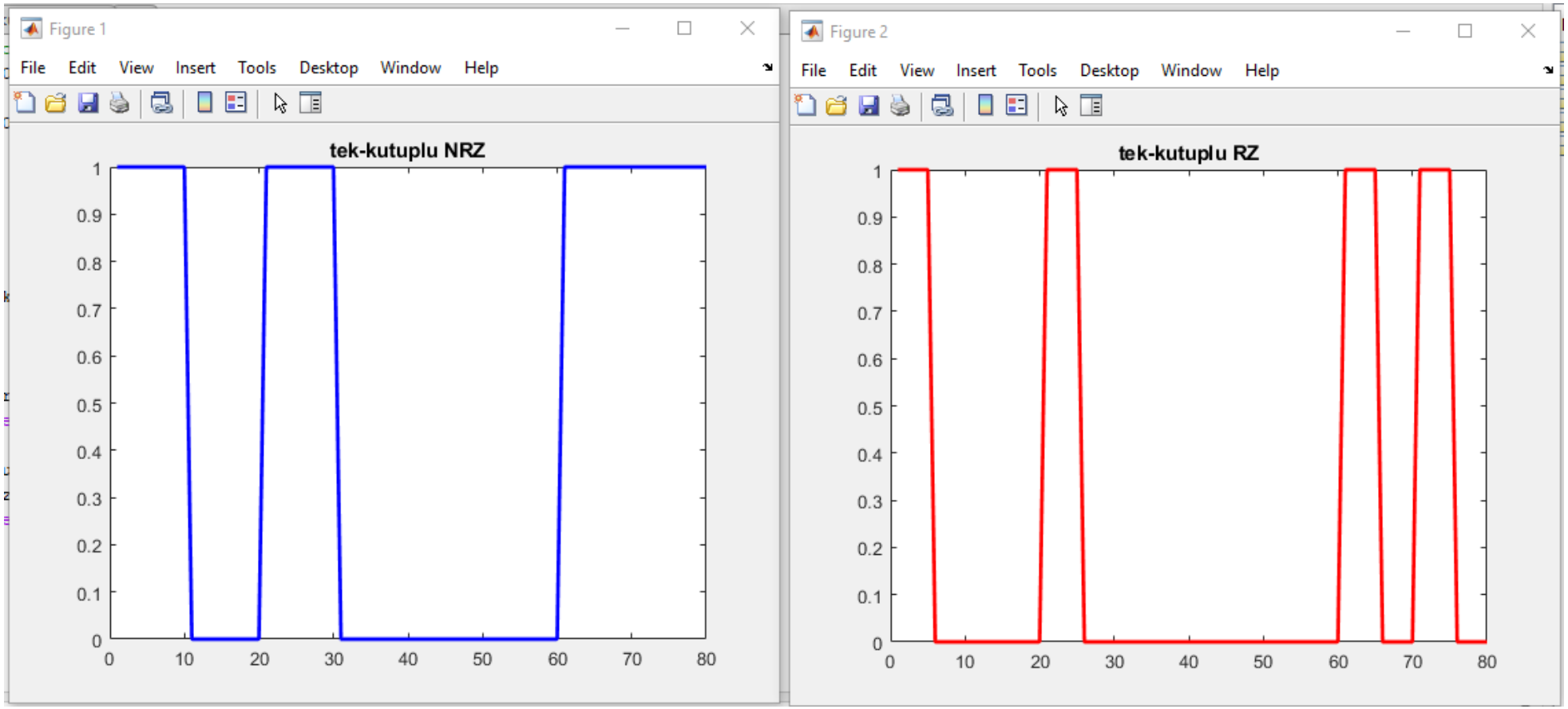
main.m | tekkutuplu.m

```
% tekkutuplu.m fonksiyon
function y= tekkutuplu(isaret, fd, fs, kodlama)
    oran=fs/fd; % ikili bite ait her örneğin kaç örneğe modüle edileceği oranı
    if strcmp(kodlama, 'nrz') % NRZ kodlama isteniyor
        for i=0:max(size(isaret))-1
            if isaret(i+1)==0 % ikili değer sıfır ise
                y(i*oran+1:(i+1)*oran)=zeros(1,oran); % ikili değer bir ise
            else
                y(i*oran+1:(i+1)*oran)=ones(1,oran);
            end
        end
    elseif strcmp(kodlama, 'rz') % RZ kodlama isteniyor
        for i=0:max(size(isaret))-1
           if isaret(i+1)==0 % ikili değer sıfır ise
               y(i*oran+1:(i+1)*oran)=zeros(1,oran);
           else % ikili değer bir ise
               y(i*oran+1:ceil((i+0.5)*oran))=ones(1,ceil(oran*0.5));
               y(ceil((i+0.5)*oran+1):(i+1)*oran)=zeros(1,oran-round(oran*0.5));
           end
        end
    else
        error('Hatalı hat kodu');
    end
```

4.2. Hat Kodlaması

1. Tek-kutuplu İşaretleşme

Örnek 4.2



1. Tek-kutuplu İşaretleşme

Örnek 4.3. Matlab kullanarak tek-kutuplu NRZ ve doluluk-boşluk oranı %50 olan tek-kutuplu RZ işaretlerin güç spektral yoğunluğunu bulunuz.

1. **Sinyallerin Tanımlanması:**
   - Tek-kutuplu NRZ: 0 ve A seviyelerinde değişen bir sinyaldir.
   - Tek-kutuplu RZ (%50 doluluk oranı): 0 ve A seviyelerinde değişen ancak bit süresinin yarısında sıfıra dönen bir sinyaldir.
2. **Zaman Çözünürlüğünün Belirlenmesi:**
   - Bit süresi T_b belirlenir.
   - Örnekleme frekansı f_s belirlenerek zaman vektörü oluşturulur.
3. **Sinyal Üretilmesi:**
   - Rastgele bit dizisi oluşturulur.
   - NRZ ve RZ sinyalleri uygun şekilde üretilir.
4. **Güç Spektral Yoğunluğu (PSD) Hesaplanması:**
   - Hızlı Fourier Dönüşümü (FFT) kullanarak frekans bileşenleri hesaplanır.
   - Welch yöntemi ile güç spektral yoğunluğu elde edilir.
5. **Grafik Çizimi:**
   - Tek-kutuplu NRZ ve RZ sinyallerinin zaman domeninde gösterimi ve PSD grafikleri.

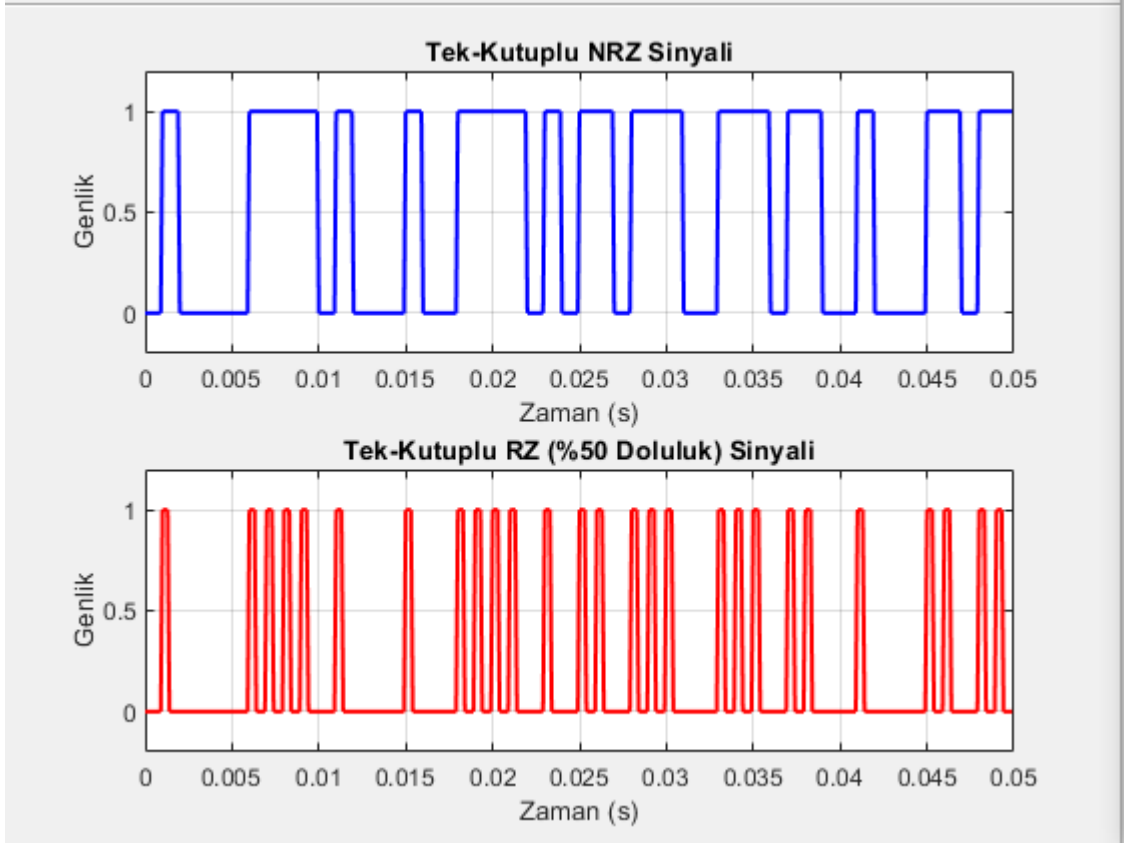
1. Tek-kutuplu İşaretleşme

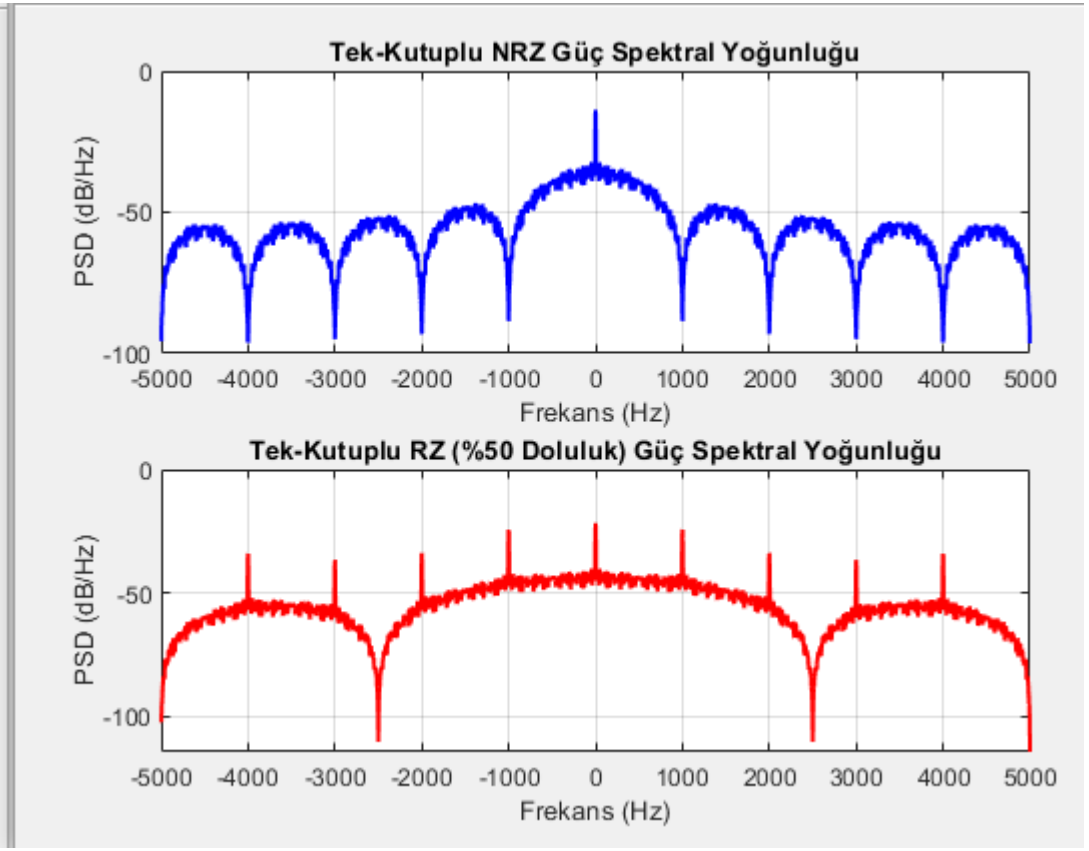
Örnek 4.3.

```
clc; clear; close all;

% Parametreler
Rb = 1000;  % Bit hızı (bps)
Tb = 1/Rb;  % Bit süresi
fs = 10*Rb; % Örnekleme frekansı
N = 1000;   % Bit sayısı
t = 0:1/fs:(N*Tb)-1/fs; % Zaman vektörü

% Rastgele bit dizisi (0 ve 1 seviyelerinde)
bit_stream = randi([0 1], 1, N);

% Tek-kutuplu NRZ işareti
NRZ_signal = repelem(bit_stream, fs*Tb);

% Tek-kutuplu RZ işareti (%50 doluluk oranı)
RZ_signal = repelem(bit_stream, fs*Tb);
for i = 1:N
    RZ_signal((i-1)*fs*Tb + round(fs*Tb/2) : i*fs*Tb) = 0; % Bit süresinin yarısında sıfıra dönüyor
end

% Güç Spektral Yoğunluğu (PSD) hesaplama
[PSD_NRZ, f_NRZ] = pwelch(NRZ_signal, [], [], [], fs, 'centered');
[PSD_RZ, f_RZ] = pwelch(RZ_signal, [], [], [], fs, 'centered');

% Grafikler
figure;
subplot(2,1,1);
plot(t(1:500), NRZ_signal(1:500), 'b', 'LineWidth', 1.5);
xlabel('Zaman (s)'); ylabel('Genlik'); title('Tek-Kutuplu NRZ Sinyali');
ylim([-0.2 1.2]); grid on;

subplot(2,1,2);
plot(t(1:500), RZ_signal(1:500), 'r', 'LineWidth', 1.5);
xlabel('Zaman (s)'); ylabel('Genlik'); title('Tek-Kutuplu RZ (%50 Doluluk) Sinyali');
ylim([-0.2 1.2]); grid on;

figure;
subplot(2,1,1);
plot(f_NRZ, 10*log10(PSD_NRZ), 'b', 'LineWidth', 1.5);
xlabel('Frekans (Hz)'); ylabel('PSD (dB/Hz)'); title('Tek-Kutuplu NRZ Güç Spektral Yoğunluğu');
grid on;
```

1. Tek-kutuplu İşaretleşme

Örnek 4.3





1. Tek-kutuplu İşaretleşme

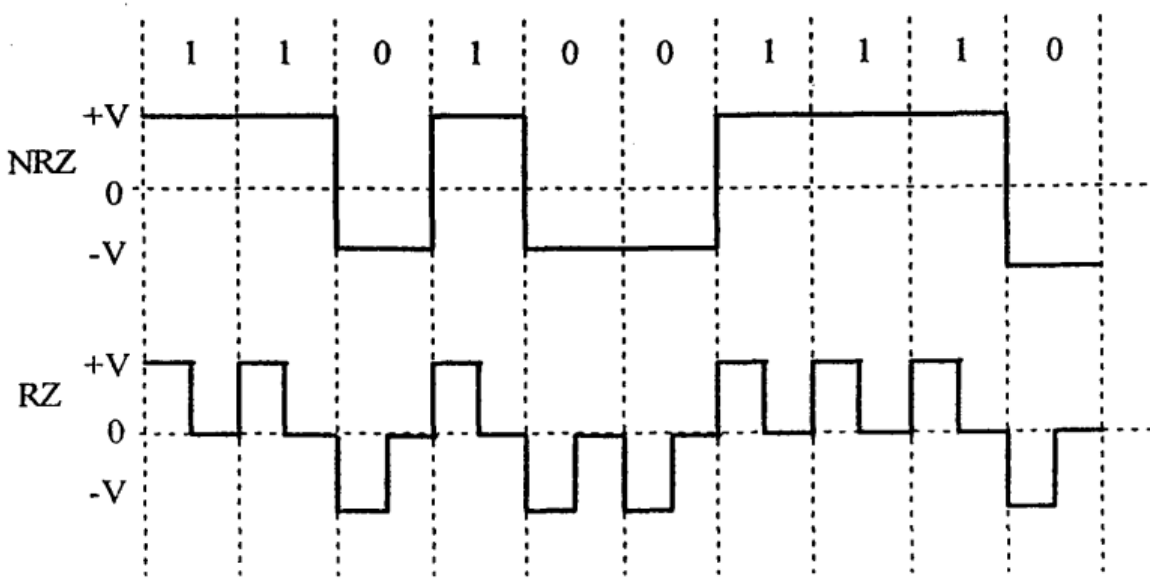
Örnek 4.3

- **Tek-kutuplu NRZ:**
  - 0 ve 1 seviyelerinde değişen, sürekli bir sinyaldir.
  - Güç spektral yoğunluğu, ana frekans bileşeni etrafında yoğunlaşır ve geniş bir bant genişliği içerir.
- **Tek-kutuplu RZ (%50 doluluk oranı):**
  - Bit süresinin ilk yarısında "1", ikinci yarısında "0" olan bir sinyaldir.
  - Spektrumda daha geniş bir bant genişliğine sahiptir, çünkü bit başına enerji daha kısa sürede yayılır.
- **pwelch Fonksiyonu:**
  - Welch metodu ile güç spektral yoğunluğunu hesaplar.
  - Daha az dalgalanma ile düzgün bir spektrum elde edilir.

Bu kodu çalıştırarak hem zaman domeninde sinyalleri görebilir hem de güç spektral yoğunluklarını analiz edebilirsiniz.

2. Kutuplu İşaretleşme

- Kutuplu işaretleşme kullanıldığında, ikili sembollerden birisi ***p(t)*** darbesi ile gösterildiğinde, diğer sembol ***-p(t)*** ile gösterilmektedir.
- Bu nedenle kutuplu işaretleşmede, 1 biti p(t) ile gösterilen bir darbe kullanılarak iletildiğinde, 0 biti -p(t) darbesi kullanılarak iletilmektedir.
- Şekil 4.6'da dikdörtgensel darbeler için kutuplu işaretleşmenin ***sıfıra dönmeyen (NRZ)*** ve ***sıfıra dönen (RZ)*** dalga biçimleri gösterilmektedir.



Şekil 4.6. Kutuplu işaretleşme NRZ ve RZ dalga biçimleri.

2. Kutuplu İşaretleşme

- Kutuplu işaretleşme NRZ ve RZ dalga biçimlerinin spektrumlarında tek-kutuplu işaretleşme NRZ ve RZ dalga biçimlerinin spektrumlarının aksine *dürtü şeklinde frekans bileşenleri oluşmamaktadır*.
- Bunun nedeni, kutuplu işaretleşmede *bir sembol için kullanılan darbenin **tersinin** diğer sembol için kullanılması* nedeniyle *güç spektral yoğunluğunda dürtü bileşenlerinin birbirlerini götürmesidir*.
- *Güç spektral yoğunluğunda* dürtü şeklinde *frekans bileşenlerinin* gözükmemesi nedeniyle kutuplu RZ işaretlerinden zamanlama bilgisinin doğrudan elde edilmesi ***mümkün değilmiş gibi gözükmektedir***.
- Fakat, kutuplu RZ işaretleri, bir *doğrultucudan geçirilerek* kolayca ***tek-kutuplu RZ işaretlerine dönüştürülebilmektedir*** ve *bu nedenle doğrultma işleminden sonra zamanlama bilgisi çıkartılabilmektedir*.
- Kutuplu işaretler, tek-kutuplu işaretlerle aynı spektral özelliklere sahip olduğundan iletim için gerekli bant genişliği aynı olmaktadır.
- Kutuplu işaretleşme sayesinde iletilen dalga biçiminin ortalama değeri düşürülmekle beraber, kısa süreli ortalama değer her zaman sıfır olmayacağından ac-bağlaşımlı hatlar üzerinden iletim yapıldığında tek-kutuplu işaretleşmeye benzer bozulmalar gözlenebilmektedir.

2. Kutuplu İşaretleşme

Örnek 4.5: Bir ikili bilgi işareti [1 0 1 0 0 0 1 1] değerlerini almaktadır. Bu bilgi işaretine göre kutuplu NRZ ve doluluk-boşluk oranı %50 kutuplu RZ işaretlerini oluşturup gösteren bir Matlab programı yazınız.

main4_5.m | kutuplu.m

```
Fs=10000;   % Tek-kutuplu kodlanmış (tabanbant modüle edilmiş) işaretin
% örnekleme frekansı
Fd=1000; % İkili işaretin örnekleme frekansı (modülasyon öncesi).
% Bu değer ikili işaretteki bir bitin programda kaç örnek ile
% gösterildiğini tanımlamaktadır. Bu nedenle Tb=1/Fd olacaktır.
% Bu değer ikili iletim oranına eşittir, bu nedenle bu örnekte
% rb=1 kbit/s dir. Not: Fs/Fd mutlaka tamsayı olmalıdır
b=[1 0 1 0 0 0 1 1]; % İkili bit dizini
bnrz=kutuplu(b,Fd,Fs, 'nrz'); % Tek-kutuplu NRZ işareti elde et
% Not: kutuplu(.) fonksiyonu aşağıdadır
figure;
plot (bnrz,'Color', [1 0 0], 'LineWidth', 2); % Kutuplu NRZ işareti çizdir
title('kutuplu NRZ');
brz=kutuplu(b,Fd,Fs,'rz');  % Tek-kutuplu RZ işareti elde et
figure;
plot (brz, 'Color', [0 0 1], 'LineWidth', 2); % Kutuplu RZ işareti çizdir
title('kutuplu RZ');
```

2. Kutuplu İşaretleşme

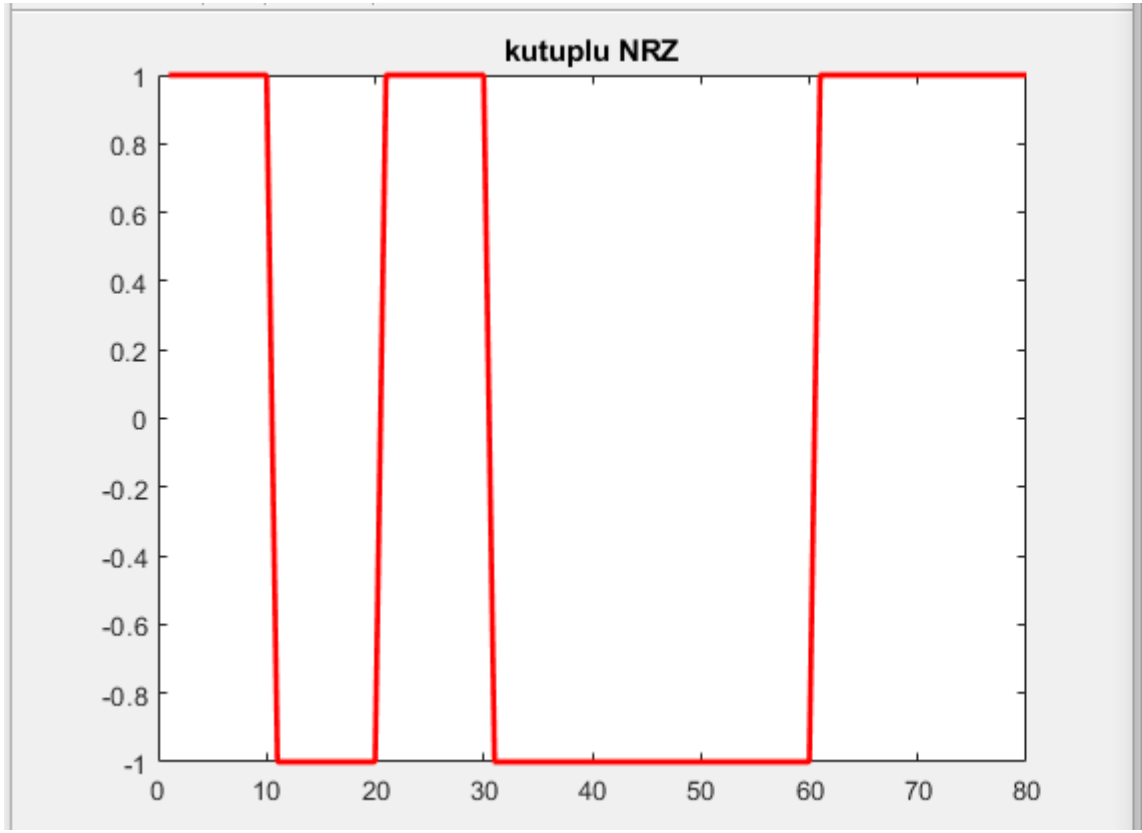
Örnek 4.5:

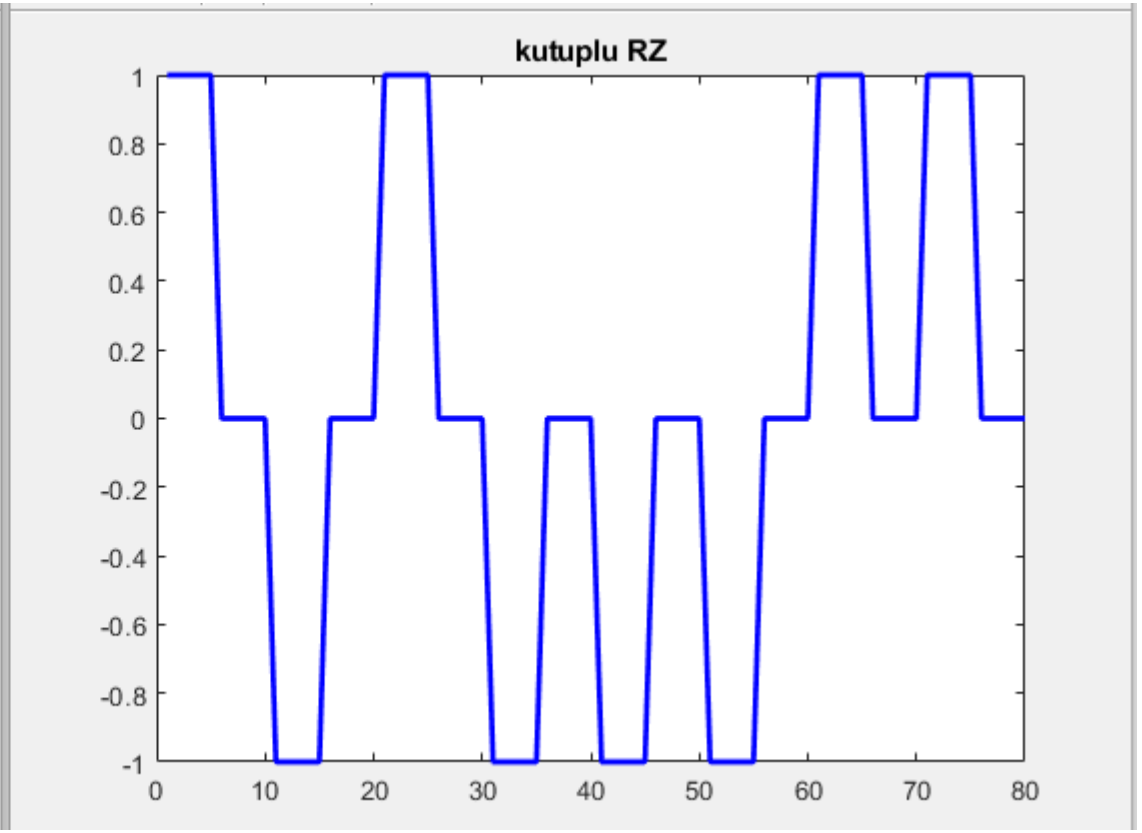
main4_5.m | kutuplu.m

```
function y= kutuplu (isaret, fd, fs, kodlama) % fonksiyon tanımı
oran=fs/fd; % ikili bite ait her örneğin kaç örneğe modüle edileceği oranı
if strcmp(kodlama, 'nrz') % NRZ kodlama isteniyor
    for i=0:max(size(isaret))-1
        if isaret(i+1)==0 % ikili değer sıfır ise
            y(i*oran+1:(i+1)*oran)=(-1)*ones(1,oran);
        else  % ikili değer bir ise
            y(i*oran+1:(i+1)*oran)=ones(1,oran);
        end
    end
elseif strcmp(kodlama, 'rz') % RZ kodlama isteniyor
    for i=0:max(size(isaret))-1
        if isaret(i+1)==0 % ikili değer sıfır ise
            y(i*oran+1:ceil((i+0.5)*oran))=(-1)*ones(1,ceil(oran*0.5));
            y(ceil((i+0.5)*oran+1):(i+1)*oran)=zeros(1,oran-round(oran*0.5));
        else % ikili değer bir ise
            y(i*oran+1:ceil((i+0.5)*oran))=ones(1,ceil(oran*0.5));
            y(ceil((i+0.5)*oran+1):(i+1)*oran)=zeros(1,oran-round(oran*0.5));
        end
     end
else
    error('Hatalı hat kodu');
end
```

2. Kutuplu İşaretleşme

Örnek 4.5:





2. Kutuplu İşaretleşme

Örnek 4.6: Kutuplu NRZ ve doluluk-boşluk oranı %50 olan kutuplu RZ işaretlerin güç spektral yoğunluğunu bulunuz.

1. **Sinyallerin Tanımlanması:**
   - **Kutuplu NRZ:** -A ve A seviyelerinde değişen bir sinyaldir.
   - **Kutuplu RZ (%50 doluluk oranı):** -A ve A seviyelerinde değişen ancak bit süresinin yarısında sıfıra dönen bir sinyaldir.
2. **Zaman Çözünürlüğünün Belirlenmesi:**
   - Bit süresi T_b belirlenir.
   - Örnekleme frekansı f_s belirlenerek zaman vektörü oluşturulur.
3. **Sinyal Üretilmesi:**
   - Rastgele bit dizisi oluşturulur.
   - Kutuplu NRZ ve RZ sinyalleri uygun şekilde üretilir.
4. **Güç Spektral Yoğunluğu (PSD) Hesaplanması:**
   - Hızlı Fourier Dönüşümü (FFT) kullanarak frekans bileşenleri hesaplanır.
   - Welch yöntemi ile güç spektral yoğunluğu elde edilir.
5. **Grafik Çizimi:**
   - Kutuplu NRZ ve RZ sinyallerinin zaman domeninde gösterimi ve PSD grafikleri.

2. Kutuplu İşaretleşme

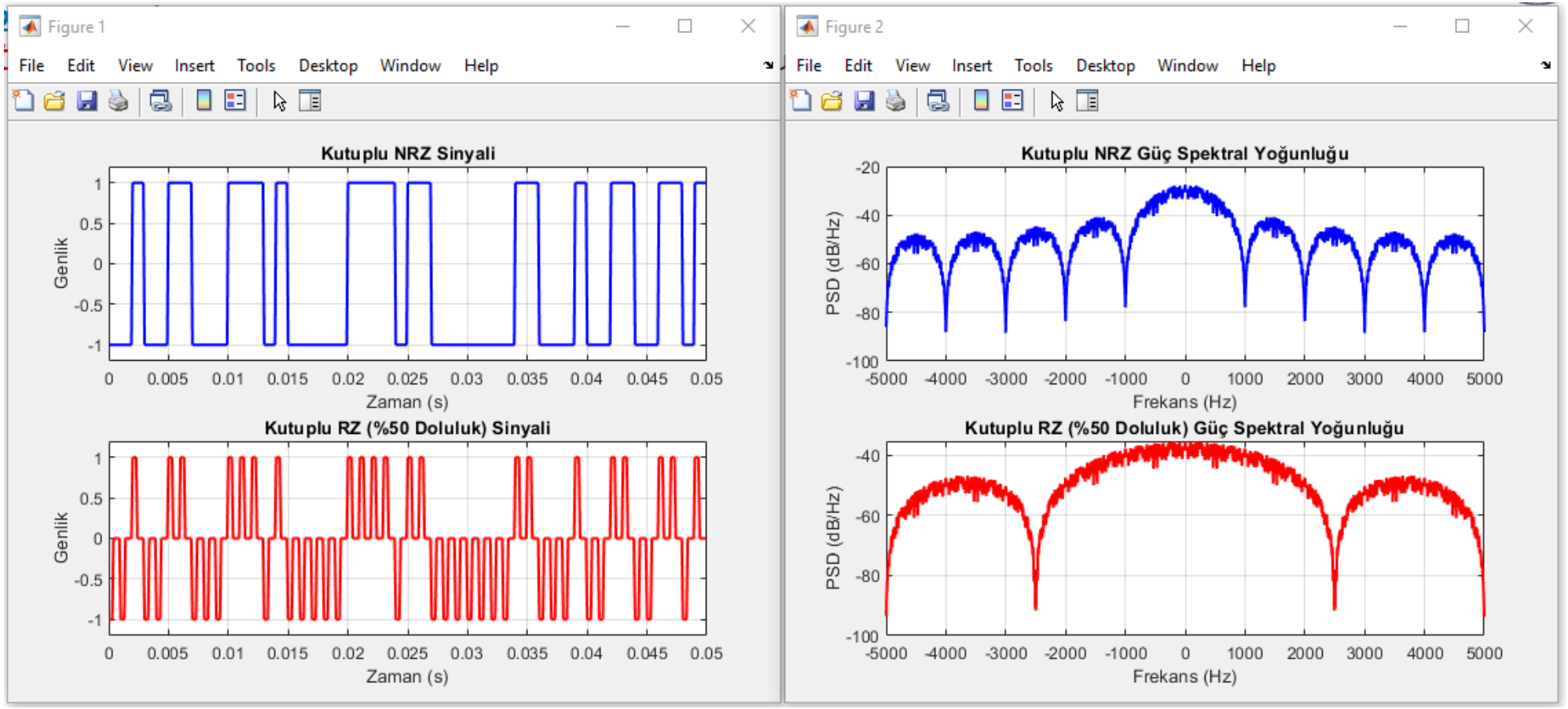
Örnek 4.6:

main4_6.m

```
clc; clear; close all;
% Parametreler
Rb = 1000;  % Bit hızı (bps)
Tb = 1/Rb;  % Bit süresi
fs = 10*Rb; % Örnekleme frekansı
N = 1000;   % Bit sayısı
t = 0:1/fs:(N*Tb)-1/fs; % Zaman vektörü

% Rastgele bit dizisi (-1 ve 1 seviyelerinde)
bit_stream = 2*randi([0 1], 1, N) - 1;

% Kutuplu NRZ işareti
NRZ_signal = repelem(bit_stream, fs*Tb);
% Kutuplu RZ işareti (%50 doluluk oranı)
RZ_signal = repelem(bit_stream, fs*Tb);
for i = 1:N
    RZ_signal((i-1)*fs*Tb + round(fs*Tb/2) : i*fs*Tb) = 0; % Bit süresinin yarısında sıfıra dönüyor
end

% Güç Spektral Yoğunluğu (PSD) hesaplama
[PSD_NRZ, f_NRZ] = pwelch(NRZ_signal, [], [], [], fs, 'centered');
[PSD_RZ, f_RZ] = pwelch(RZ_signal, [], [], [], fs, 'centered');

% Grafikler
figure;
subplot(2,1,1);
plot(t(1:500), NRZ_signal(1:500), 'b', 'LineWidth', 1.5);
xlabel('Zaman (s)'); ylabel('Genlik'); title('Kutuplu NRZ Sinyali');
ylim([-1.2 1.2]); grid on;
subplot(2,1,2);
plot(t(1:500), RZ_signal(1:500), 'r', 'LineWidth', 1.5);
xlabel('Zaman (s)'); ylabel('Genlik'); title('Kutuplu RZ (%50 Doluluk) Sinyali');
ylim([-1.2 1.2]); grid on;

figure;
subplot(2,1,1);
plot(f_NRZ, 10*log10(PSD_NRZ), 'b', 'LineWidth', 1.5);
xlabel('Frekans (Hz)'); ylabel('PSD (dB/Hz)'); title('Kutuplu NRZ Güç Spektral Yoğunluğu');
grid on;
subplot(2,1,2);
plot(f_RZ, 10*log10(PSD_RZ), 'r', 'LineWidth', 1.5);
xlabel('Frekans (Hz)'); ylabel('PSD (dB/Hz)'); title('Kutuplu RZ (%50 Doluluk) Güç Spektral Yoğunluğu');
grid on;
```

2. Kutuplu İşaretleşme

Örnek 4.6:



2. Kutuplu İşaretleşme

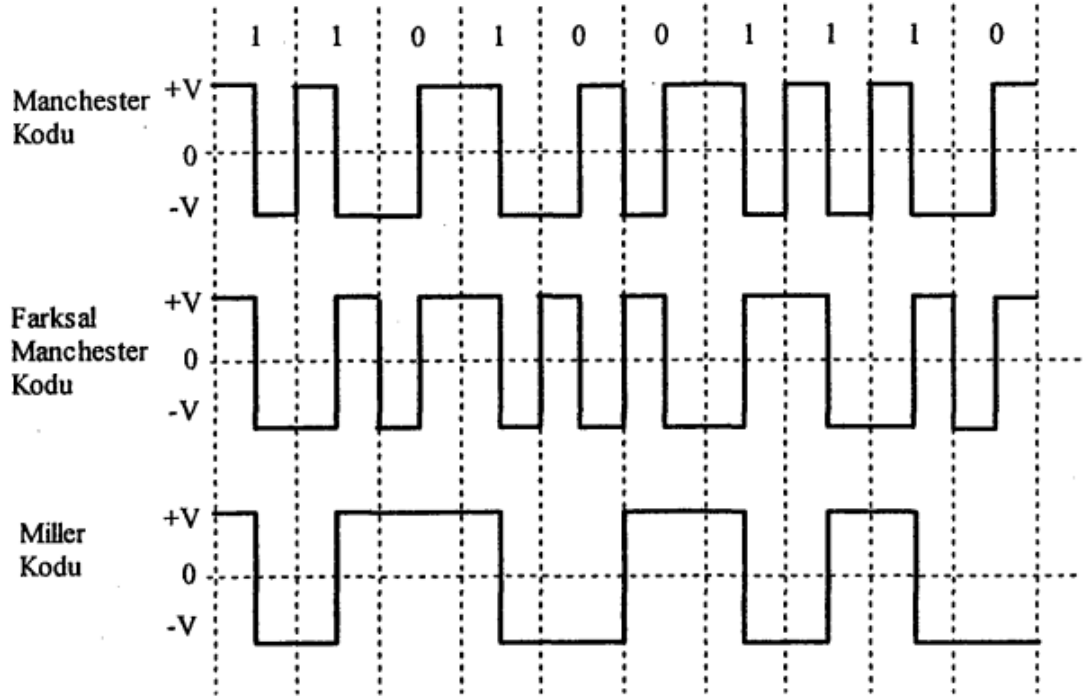
Örnek 4.6:

- **Kutuplu NRZ:**
  - -1 ve 1 seviyelerinde değişen, sürekli bir sinyaldir.
  - Güç spektral yoğunluğu, ana frekans bileşeni etrafında yoğunlaşır ve geniş bir bant genişliği içerir.
- **Kutuplu RZ (%50 doluluk oranı):**
  - Bit süresinin ilk yarısında -1 veya 1, ikinci yarısında sıfır olan bir sinyaldir.
  - Spektrumda daha geniş bir bant genişliğine sahiptir, çünkü bit başına enerji daha kısa sürede yayılır.
- **pwelch Fonksiyonu:**
  - Welch metodu ile güç spektral yoğunluğunu hesaplar.
  - Daha az dalgalanma ile düzgün bir spektrum elde edilir.

Bu kodu çalıştırarak hem zaman domeninde sinyalleri görebilir hem de güç spektral yoğunluklarını analiz edebilirsiniz.

3. Faz-Kodlanmış İşaretleşme

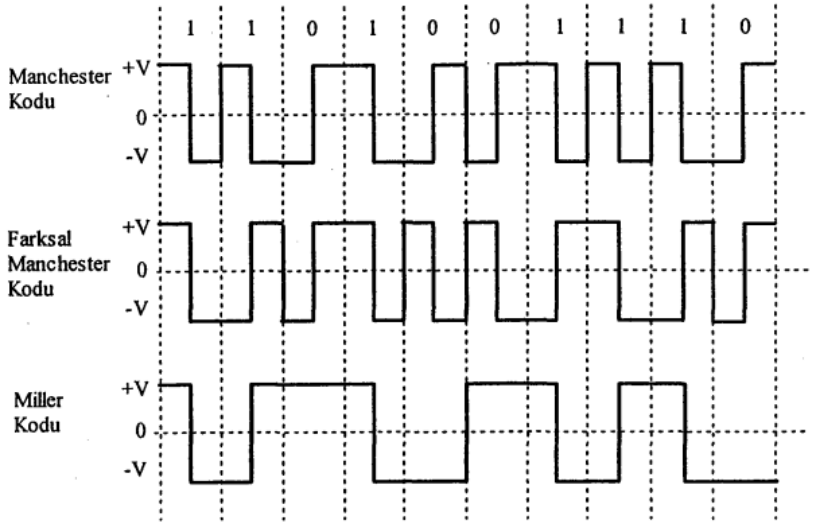
- Faz kodlanmış işaretleşme, sıfır veya *çok düşük dc değerine* sahip bir hat kodlaması sağlamak üzere geliştirilmiştir.
- Faz kodlanmış işaretleşmede en yaygın kullanılan dalga biçimleri Şekil 4.8'de gösterilen Manchester ve Miller kodlarıdır.



Şekil 4.8.Faz-kodlanmış işaretleşme dalga biçimleri.

3. Faz-Kodlanmış İşaretleşme

- ***Manchester kodu***, 1 sembolü için bit zaman diliminin yarısına kadar pozitif gerilimde, yarısından sonra negatif gerilimde; 0 sembolü için de tam tersi olacak şekilde bit zaman diliminin yarısına kadar negatif gerilimde yarısından sonra pozitif gerilimde olan bir dalga biçimi kullanmaktadır.
- ***Bölünmüş-faz kodlaması*** olarak da bilinen Manchester kodunun DC bileşeni sıfır olduğu için, Manchester kodlaması ac-bağlaşımlı hatlar üzerinden iletim için kullanılabilmektedir.
- Manchester kodlamasında her zaman sembol oranında seviyeler arasında geçiş olduğu için işaretten zamanlama bilgisi çıkartılabilmektedir.
- Manchester kodlaması Ethernet yerel alan ağlarında (LAN) kullanılmaktadır.
- ***Jetonlu halka ağ*** standardı olarak da ***farksal Manchester*** kodlaması kullanılmaktadır.
- Farksal Manchester kodunda, bit zaman diliminin yarısında bir geçiş olmakla beraber, 1 sembolü için iletilen dalga biçimi bir önceki zaman dilimindeki seviyeden devam etmekte, 0 sembolü için ise seviye değiştirilmekte ve bit-zaman diliminin yarısında diğer seviyeye geçiş gerçekleştirilmektedir.



Şekil 4.8.Faz-kodlanmış işaretleşme dalga biçimleri.

4.2. Hat Kodlaması

3. Faz-Kodlanmış İşaretleşme

- Miller kodlamasında, 1 sembolü bir önceki zaman dilimindeki seviyeden devam ederek bit zaman diliminin ortasında diğer seviyeye geçmektedir.
- Miller kodlamasında, 0 sembolü eğer 1 sembolünden sonra geliyorsa hiçbir değişim göstermeden aynı seviyede devam eden bir dalga biçimi kullanılmakta, 0 sembolü başka bir 0 sembolünden sonra geliyorsa bit zaman diliminin başında diğer seviyeye geçerek o seviyeden devam eden bir dalga biçimi kullanılmaktadır.
- Bu sayede işaretin ortalama değeri çok düşük seviyelerde tutulabilmekte ve arka arkaya gelen sıfırlarda seviye değiştirildiğinden zamanlama bilgisi kaybolmamaktadır.
- Miller kodunun en büyük avantajı $0.4/T_b$, merkezli çok dar bir spektrum bandına sahip olmasıdır.
- Miller kodunun tabanbant bant genişliği B= $0.6/T_b$, olarak alınabilmektedir.
- *Bu dar bant genişliği*, esasında *zaman uzayında darbe genişliğinin arttırılması sonucunda oluşmaktadır.*
- Gecikme modülasyonu olarak da adlandırılan Miller kodu düşük ortalaması nedeniyle ***yüksek kalitede sayısal teyp*** kaydı için geliştirilmiştir.
- Miller kodlamasındaki seviye geçişleri sayesinde işaretten zamanlama bilgisi çıkartılabilmektedir.
- Miller kodunun çözülmesi için Miller kodundan *sıfıra dönmeyen kutuplu (NRZ) biçime dönüşüm* yapılabilmektedir.

3. Faz-Kodlanmış İşaretleşme

Örnek 4.7 : Bir ikili bilgi işareti [10100011]değerlerini almaktadır. Bu bilgi işaretine göre Manchester ve Miller işaretlerini gösterebilen bir Matlab programı yazınız.

main4_7.m | fazkodla.m

```
Fs=10000;    % Tek-kutuplu kodlanmış (tabanbant modüle edilmiş) işaretin
             % örnekleme frekansı
Fd=1000;     % İkili işaretin örnekleme frekansı (modülasyon öncesi).
             % Bu değer ikili işaretteki bir bitin programda kaç örnek ile
             % gösterildiğini tanımlamaktadır. Bu nedenle Tb=1/Fd olacaktır.
b=[1 0 1 0 0 0 1 1]; % İkili bit dizini
manch=fazkodla(b, Fd, Fs, 'manchester'); % Manchester işaretini elde et
figure;
plot (manch,'Color', [1 0 0], 'LineWidth', 2);
title('Manchester');
mil=fazkodla(b, Fd, Fs, 'miller'); % Miller işaretini elde et
figure;
plot (mil,'Color', [0 0 1], 'LineWidth', 2);
title('Miller');
```

3. Faz-Kodlanmış İşaretleşme

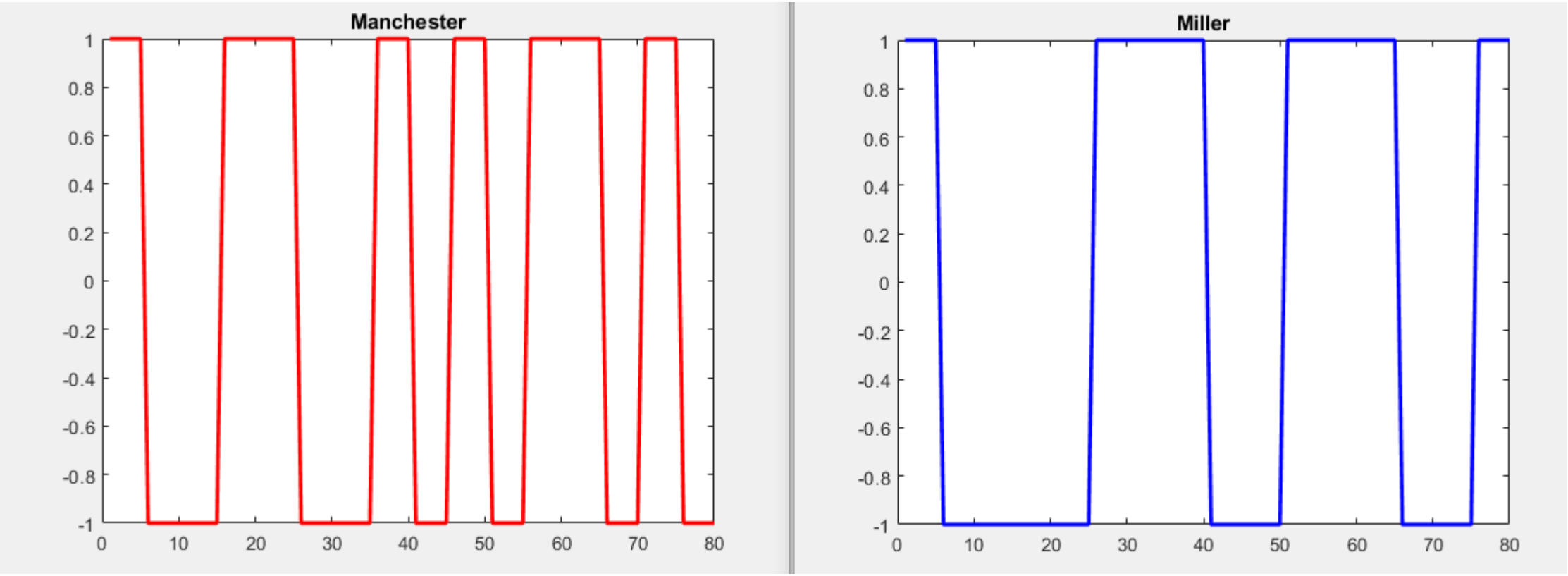
Örnek 4.7 : Bir ikili bilgi işareti [10100011]değerlerini almaktadır. Bu bilgi işaretine göre Manchester ve Miller işaretlerini gösterebilen bir Matlab programı yazınız.

main4_7.m | fazkodla.m

```
function y=fazkodla(isaret, fd, fs, kodlama) % fonksiyon tanımı
oran=fs/fd; % ikili bite ait her örneğin kaç örneğe modüle edileceği oranı
if strcmp(char(kodlama), 'manchester') % Manchester isteniyor
    for i=0:max(size(isaret))-1
        if isaret(i+1)==0 % ikili değer sıfır ise
            y(i*oran+1:ceil((i+0.5)*oran))=(-1)*ones(1,ceil(oran*0.5));
            y(ceil((i+0.5)*oran+1):(i+1)*oran)=ones(1, oran-round(oran*0.5));
        else  % ikili değer bir ise
            y(i*oran+1:ceil((i+0.5)*oran))=ones(1,ceil(oran*0.5));
            y(ceil((i+0.5)*oran+1):(i+1)*oran)=(-1)*ones(1, oran-round(oran*0.5));
        end
    end
elseif strcmp(kodlama, 'miller') % Miller isteniyor
    t=1; % bir önceki seviye nedir?
    p=0; % bir önceki bit nedir?
    for i=0:max(size(isaret))-1
        if isaret(i+1)==0 % ikili değer sıfır ise
            if p==0 % bir önceki bit sıfır ise
                t=t*(-1); % seviye değiştir
            end
            y(i*oran+1:(i+1)*oran)=t*ones(1,oran);
            p=0; % artık bir önceki bit sıfır olacak
        else % ikili değer bir ise
                y(i*oran+1:ceil((i+0.5)*oran))=t*ones(1, ceil(oran*0.5));
                y(ceil((i+0.5)*oran+1):(i+1)*oran)=t*(-1)*ones(1,oran-round(oran*0.5));
                t=t*(-1); % seviye değiştir
                p=1; % artık bir önceki bit bir olacak
        end
    end
else
error('Hatalı hat kodu');
end
```

3. Faz-Kodlanmış İşaretleşme

Örnek 4.7 : Bir ikili bilgi işareti [10100011]değerlerini almaktadır. Bu bilgi işaretine göre Manchester ve Miller işaretlerini gösterebilen bir Matlab programı yazınız.



3. Faz-Kodlanmış İşaretleşme

Örnek 4.8 : Bir ikili bilgi işareti [10100011]değerlerini almaktadır. Bu bilgi işaretine göre Manchester ve Miller işaretlerini gösterebilen bir Matlab programı yazınız.

1. **Manchester ve Miller Kodlamalarının Tanımlanması:**
   - Manchester:
     - Her bit süresinde (T_b) "0" → (1, -1), "1" → (-1, 1) olarak değişir.
   - Miller:
     - "1" bitinde her zaman ortada bir geçiş vardır.
     - "0" bitinde, eğer bir önceki bit "0" ise değişmez, "1" ise başında bir geçiş olur.
2. **Zaman Çözünürlüğünün Belirlenmesi:**
   - Bit süresi T_b belirlenir.
   - Örnekleme frekansı f_s belirlenerek zaman vektörü oluşturulur.
3. Sinyal Üretilmesi:
   - Rastgele bit dizisi oluşturulur.
   - Manchester ve Miller sinyalleri uygun şekilde üretilir.
4. **Güç Spektral Yoğunluğu (PSD) Hesaplanması:** Welch yöntemi kullanılarak güç spektral yoğunluğu elde edilir.
5. **Grafik Çizimi:** Manchester ve Miller sinyallerinin zaman domeninde gösterimi ve PSD grafikleri.

3. Faz-Kodlanmış İşaretleşme

Örnek 4.8 :

```
clc; clear; close all;

% Parametreler
Rb = 1000;  % Bit hızı (bps)
Tb = 1/Rb;  % Bit süresi
fs = 10*Rb; % Örnekleme frekansı
N = 1000;   % Bit sayısı
t = 0:1/fs:(N*Tb)-1/fs; % Zaman vektörü

% Rastgele bit dizisi (0 ve 1 seviyelerinde)
bit_stream = randi([0 1], 1, N);

% Manchester Kodlaması
Manchester_signal = zeros(1, N*fs*Tb);
for i = 1:N
    start_idx = (i-1)*fs*Tb + 1;
    mid_idx = start_idx + round(fs*Tb/2);
    end_idx = i*fs*Tb;
    if bit_stream(i) == 0
        Manchester_signal(start_idx:mid_idx) = 1;
        Manchester_signal(mid_idx+1:end_idx) = -1;
    else
        Manchester_signal(start_idx:mid_idx) = -1;
        Manchester_signal(mid_idx+1:end_idx) = 1;
    end
end

% Miller Kodlaması
Miller_signal = zeros(1, N*fs*Tb);
previous_bit = 1; % İlk bit öncesi referans
for i = 1:N
    start_idx = (i-1)*fs*Tb + 1;
    mid_idx = start_idx + round(fs*Tb/2);
    end_idx = i*fs*Tb;
    if bit_stream(i) == 1
        Miller_signal(start_idx:mid_idx) = previous_bit;
        previous_bit = -previous_bit;
        Miller_signal(mid_idx+1:end_idx) = previous_bit;
    else
        Miller_signal(start_idx:end_idx) = previous_bit;
        if i > 1 && bit_stream(i-1) == 1
            previous_bit = -previous_bit;
        end
    end
end

% Güç Spektral Yoğunluğu (PSD) hesaplama
[PSD_Manchester, f_Manchester] = pwelch(Manchester_signal, [], [], [], fs, 'centered');
[PSD_Miller, f_Miller] = pwelch(Miller_signal, [], [], [], fs, 'centered');
```

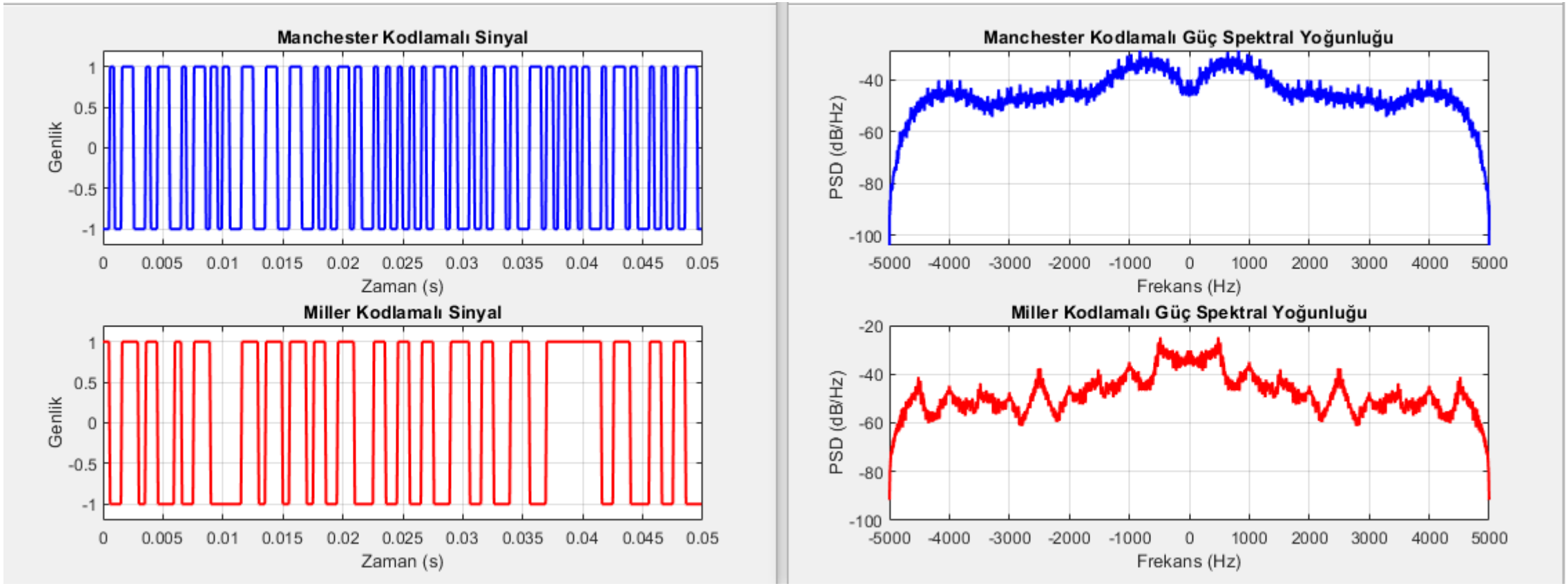
3. Faz-Kodlanmış İşaretleşme

Örnek 4.8 :

```
% Grafikler
figure;
subplot(2,1,1);
plot(t(1:500), Manchester_signal(1:500), 'b', 'LineWidth', 1.5);
xlabel('Zaman (s)'); ylabel('Genlik'); title('Manchester Kodlamalı Sinyal');
ylim([-1.2 1.2]); grid on;

subplot(2,1,2);
plot(t(1:500), Miller_signal(1:500), 'r', 'LineWidth', 1.5);
xlabel('Zaman (s)'); ylabel('Genlik'); title('Miller Kodlamalı Sinyal');
ylim([-1.2 1.2]); grid on;

figure;
subplot(2,1,1);
plot(f_Manchester, 10*log10(PSD_Manchester), 'b', 'LineWidth', 1.5);
xlabel('Frekans (Hz)'); ylabel('PSD (dB/Hz)'); title('Manchester Kodlamalı Güç Spektral Yoğunluğu');
grid on;

subplot(2,1,2);
plot(f_Miller, 10*log10(PSD_Miller), 'r', 'LineWidth', 1.5);
xlabel('Frekans (Hz)'); ylabel('PSD (dB/Hz)'); title('Miller Kodlamalı Güç Spektral Yoğunluğu');
grid on;
```

3. Faz-Kodlanmış İşaretleşme

Örnek 4.8 :



3. Faz-Kodlanmış İşaretleşme

Örnek 4.8 :

- **Manchester Kodlaması:**
  - Her bit süresinde ortada bir sıfır geçişi olur.
  - Spektrumu geniş bantlıdır ve DC bileşeni içermez.
- **Miller Kodlaması:**
  - "1" bitlerinde her zaman ortada geçiş olur.
  - "0" bitlerinde geçiş olup olmaması önceki bitlere bağlıdır.
  - Daha dar bant genişliğine sahiptir.
- **pwelch Fonksiyonu:**
  - Welch metodu ile güç spektral yoğunluğunu hesaplar.
  - Daha düzgün bir spektrum elde edilir.

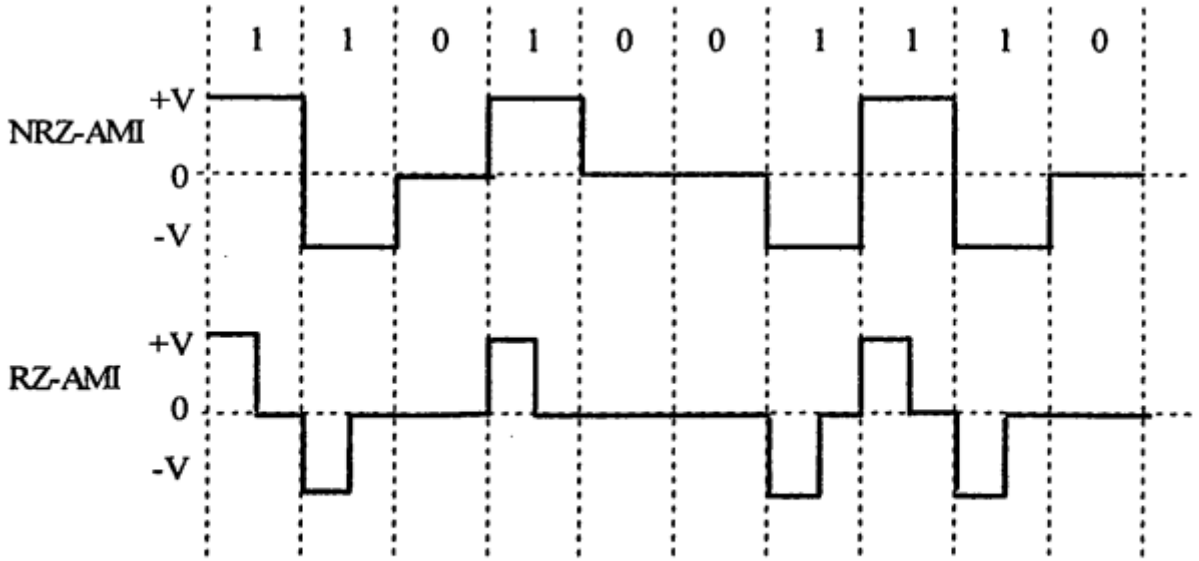
Bu kodu çalıştırarak hem zaman domeninde sinyalleri görebilir hem de güç spektral yoğunluklarını analiz edebilirsiniz.

4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi)

- ***Ters im değişimi (AMI)*** olarak da adlandırılan ***çift-kutuplu işaretleşme***, *iki adet sembol (0 ve 1) için üç adet gerilim seviyesi (-V, 0, +V) kullanmakta* ve bu nedenle ***sözde-üçlü kodlama*** olarak da nitelendirilmektedir.
- AMI kodlamasında 0 sembolü bir darbenin yokluğu yani sıfır seviyesinde bir gerilim ile gösterilmekte,
- 1 sembolü ise *sürekli değişerek* sırayla *pozitif ve negatif gerilim seviyelerindeki* darbeler ile gösterilmektedir (ters im kuralı).
- Bir önceki 1 biti pozitif seviyeden iletilmişse gelen 1 biti negatif seviyeden, bir önceki 1 biti negatif seviyeden iletilmişse gelen 1 biti pozitif seviyeden iletilmektedir.
- *Bu sayede iletilen dalga biçiminin ortalama değeri sıfırlanmaktadır*.

4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi)

- Darbelerin sıfıra dönüp dönmemesine göre Şekil 4.10'da gösterildiği gibi sıfıra dönmeyen AMI (NRZ-AMI) veya sıfıra dönen AMI (RZ-AMI) çeşitleri kullanılabilmektedir.



Şekil 4.10. Çift-kutuplu (AMI) işaretleşme dalga biçimleri

4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi)

- AMI işaretleşmesi, tabanbant PCM sistemlerinde yaygın olarak kullanılmaktadır.
- ***Çift-kutuplu*** işaretleşmenin bant genişliği, eşdeğer özellikteki ***tek-kutuplu*** ve ***kutuplu*** işaretleşmelere göre daha düşük olduğu için daha verimli bir iletim sağlanmaktadır.
- Ayrıca, AMI tek bitlik bir hata algılama imkanı sağlamaktadır.
- AMI işaretleşmesinde, sezme esnasında bir hata yapıldığı taktirde ters im kuralı bozulacağından bir bit hata yapıldığı algılanabilmektedir.
- Birden fazla hata durumunda ise hatalar işareti ters-im kuralına uygun şekilde bırakabileceği için bir bitten fazla hata yapılması durumunda hatalar algılanmayabilmektedir.

4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi)

- *Sıfıra dönen çift-kutuplu RZ-AMI* işareti bir doğrultucudan geçirilerek ***sıfıra dönen tek-kutuplu RZ-OOK*** işaretine dönüştürülebilmektedir.
- Bunun neticesinde sembol oranı frekansında oluşan spektrum bileşenleri kullanılarak zamanlama bilgisi çıkartılabildiği için uygulamada RZ-AMI dalga biçimi daha yaygın olarak kullanılmaktadır.
- Çift-kutuplu işaretleşmede sezici yapısı biraz daha karmaşıktır, çünkü alıcıda artık üç farklı seviyenin (-V, 0, +V) sezilmesi gerekmektedir.
- ***Çift-kutuplu işaretleşmedeki en büyük problem saydamlıktır***.
- Sayısal işarette arka arkaya çok sayıda 0 sembolü bulunduğunda, bu süre boyunca bir darbe gönderilmediği için zamanlama işaretinde sönümlenme meydana gelmektedir.
- Bunu engellemek için uzun süreli 0 seviyelerinin iletimine müsaade etmeyen, fakat bu nedenle daha karmaşık bir sistem yapısına sahip olan, ***yüksek yoğunluklu çift-kutuplu (HDB)*** işaretleşme kullanılabilmektedir.

4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi)

Örnek 4.9: Bir ikili bilgi işareti, [1 0 1 0 0 0 1 1] değerlerini almaktadır. Bu bilgi işaretine göre AMI-NRZ ve AMI-RZ işaretlerini oluşturan bir Matlab programı yazınız.

main4_9.m | amikodla.m

```
Fs=10000;   % Tek-kutuplu kodlanmış (tabanbant modüle edilmiş) işaretin
% örnekleme frekansı
Fd=1000;    % İkili işaretin örnekleme frekansı (modülasyon öncesi).
% Bu değer ikili işaretteki bir bitin programda kaç örnek ile
% gösterildiğini tanımlamaktadır. Bu nedenle Tb=1/Fd olacaktır.
b=[1 0 1 0 0 0 1 1]; % İkili bit dizini
aminrz=amikodla(b,Fd,Fs, 'ami-nrz'); % AMI-NRZ işaretini elde et
figure;
plot (aminrz,'Color', [1 0 0], 'LineWidth', 1.5);
title('Manchester');
amirz=amikodla(b, Fd,Fs, 'ami-rz'); % AMI-RZ işaretini elde et
figure;
plot (amirz,'Color', [0 0 1], 'LineWidth', 1.5);
title('Miller');
```

4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi)

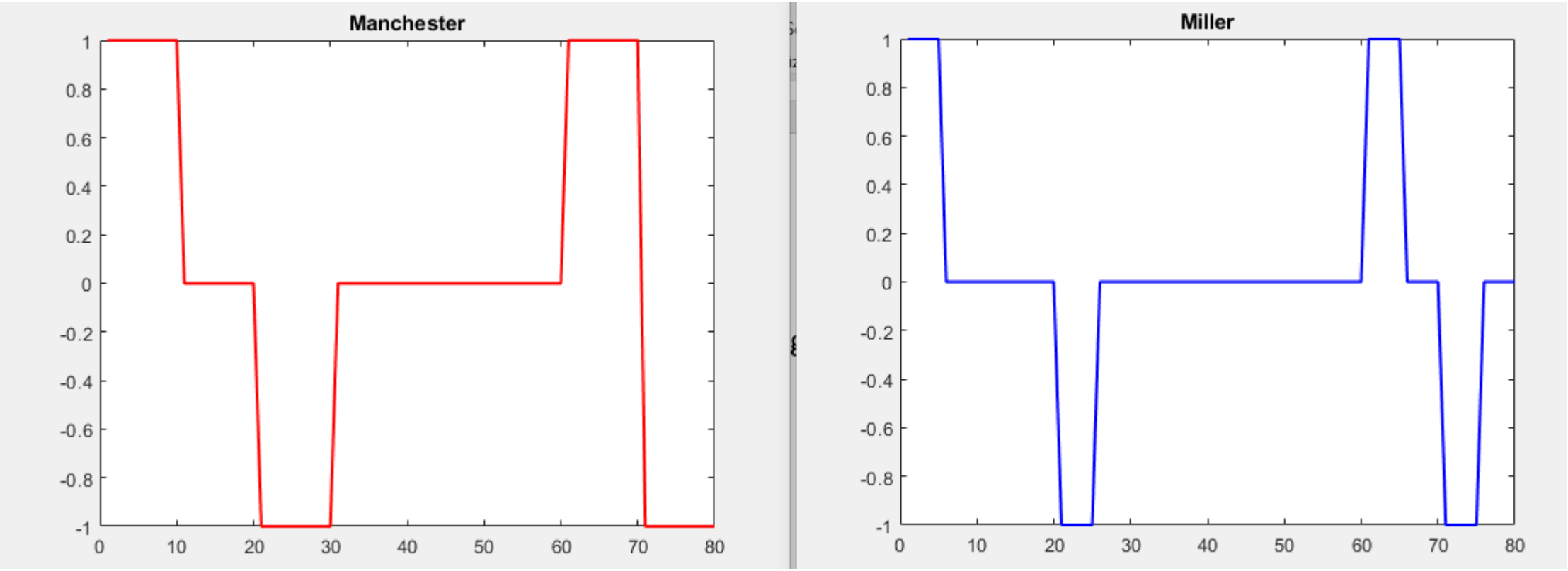
Örnek 4.9: Bir ikili bilgi işareti, [1 0 1 0 0 0 1 1] değerlerini almaktadır. Bu bilgi işaretine göre AMI-NRZ ve AMI-RZ işaretlerini oluşturan bir Matlab programı yazınız.

main4_9.m | amikodla.m

```
function y= amikodla(isaret, fd, fs, kodlama) % fonksiyon tanımı
oran=fs/fd; % ikili bite ait her örneğin kaç örneğe modüle edileceği oranı
if strcmp(kodlama, 'ami-nrz') % NRZ isteniyor
    t=1; % bir önceki seviye nedir?
    for i=0:max(size(isaret))-1
        if isaret(i+1)==0 % ikili değer sıfır ise
            y(i*oran+1:(i+1)*oran)=zeros(1,oran);
        else % ikili değer bir ise
            y(i*oran+1:(i+1)*oran)=t*ones(1,oran);
            t=t*(-1);
        end
    end
elseif strcmp(kodlama, 'ami-rz') % RZ isteniyor
    t=1; % bir önceki seviye nedir?
    for i=0:max(size(isaret))-1
        if isaret(i+1)==0  % ikili değer sıfır ise
            y(i*oran+1:(i+1)*oran)=zeros(1,oran);
        else  % ikili değer bir ise
            y(i*oran+1:ceil((i+0.5)*oran))=t*ones(1,ceil(oran*0.5));
            y(ceil((i+0.5)*oran+1):(i+1)*oran)=zeros(1,oran-round(oran*0.5));
            t=t*(-1); % seviye değiştir
        end
    end
else
    error('Hatalı hat kodu');
end
```

4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi)

Örnek 4.9: Bir ikili bilgi işareti, [1 0 1 0 0 0 1 1] değerlerini almaktadır. Bu bilgi işaretine göre AMI-NRZ ve AMI-RZ işaretlerini oluşturan bir Matlab programı yazınız.



4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi)

Örnek : Bir ikili bilgi işareti, [1 0 1 0 0 0 1 1] değerlerini almaktadır. Bu bilgi işaretine göre AMI-NRZ ve AMI-RZ işaretlerinin güç spektrumunu oluşturan bir Matlab programı yazınız.

1. **Giriş Bit Dizisinin Tanımlanması:**
   - **Verilen bit dizisi:** [10100011]
   - **AMI (Alternate Mark Inversion) Kodlaması:**
     - "1" bitleri alternatif olarak +A+A+A ve −A-A−A seviyelerine atanır.
     - "0" bitleri 000 seviyesinde kalır.
   - **NRZ ve RZ farkı:**
     - **AMI-NRZ:** "1" bitleri tam süre boyunca (+/-) değerinde kalır.
     - **AMI-RZ:** "1" bitleri sürenin ilk yarısında (+/-) değerinde olur, ikinci yarısında sıfıra düşer.
2. **Örnekleme ve Zaman Çözünürlüğü Belirlenir:**
   - **Bit süresi:** $T_b = 1$ ms (örnek)
   - **Örnekleme frekansı:** $f_s = 10$ kHz
   - **Toplam bit süresi:** $N \times T_b$
3. **AMI-NRZ ve AMI-RZ İşaretleri Oluşturulur.**
4. **Güç Spektral Yoğunluğu (PSD) Welch Yöntemi ile Hesaplanır.**
5. **Grafikler Çizilir.**

4.2. Hat Kodlaması

4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi)

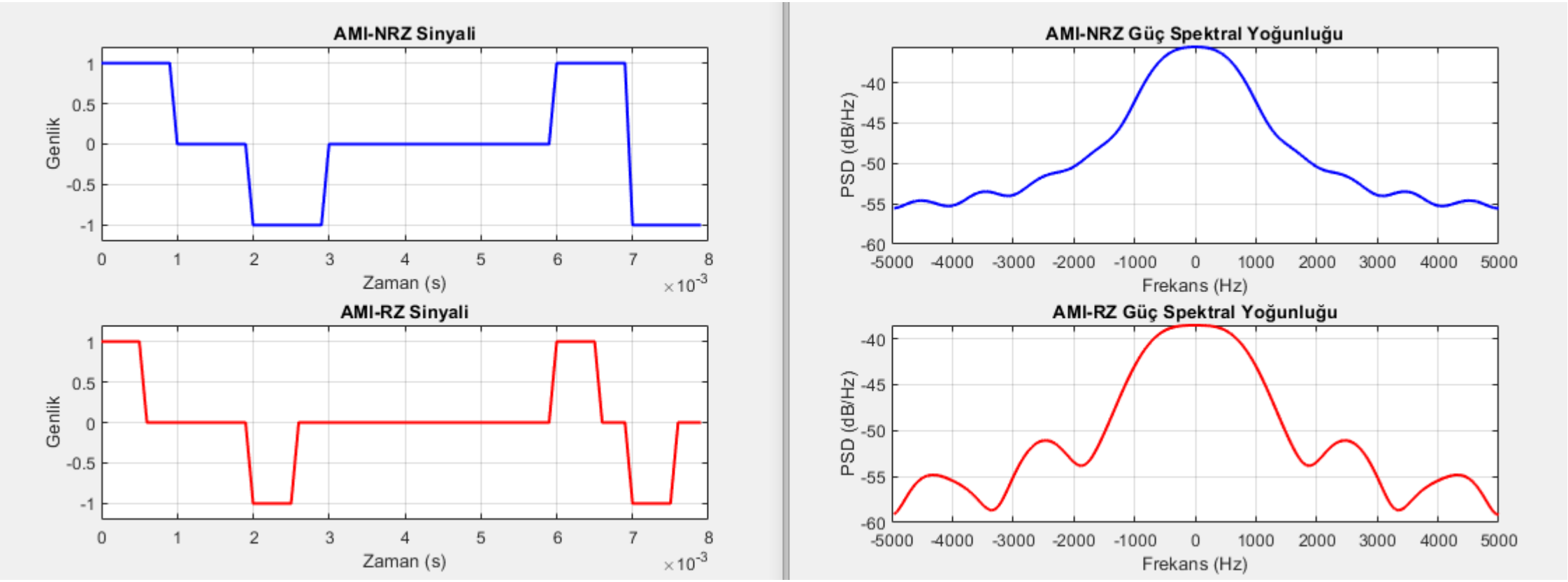
Örnek : Bir ikili bilgi işareti, [1 0 1 0 0 0 1 1] değerlerini almaktadır. Bu bilgi işaretine göre AMI-NRZ ve AMI-RZ işaretlerinin güç spektrumunu oluşturan bir Matlab programı yazınız.

mainornek.m

```
clc; clear; close all;

% Parametreler
Rb = 1000;      % Bit hızı (bps) -> 1 ms per bit
Tb = 1/Rb;      % Bit süresi
fs = 10*Rb;     % Örnekleme frekansı (10 kHz)
N = 8;          % Bit sayısı
t = 0:1/fs:(N*Tb)-1/fs; % Zaman vektörü

% İkili bilgi işareti
bit_stream = [1 0 1 0 0 0 1 1];

% AMI-NRZ ve AMI-RZ işaretlerini oluşturmak için değişkenler
AMI_NRZ_signal = zeros(1, N*fs*Tb);
AMI_RZ_signal = zeros(1, N*fs*Tb);
previous_level = 1; % AMI'de alternatif geçiş için başlangıç seviyesi

% AMI-NRZ ve AMI-RZ işaretlerinin oluşturulması
for i = 1:N
    start_idx = (i-1)*fs*Tb + 1;
    mid_idx = start_idx + round(fs*Tb/2);
    end_idx = i*fs*Tb;

    if bit_stream(i) == 1
        AMI_NRZ_signal(start_idx:end_idx) = previous_level;
        AMI_RZ_signal(start_idx:mid_idx) = previous_level;
        previous_level = -previous_level; % Alternatif olarak değiştir
    end
end

% Güç Spektral Yoğunluğu (PSD) hesaplama
[PSD_AMI_NRZ, f_AMI_NRZ] = pwelch(AMI_NRZ_signal, [], [], [], fs, 'centered');
[PSD_AMI_RZ, f_AMI_RZ] = pwelch(AMI_RZ_signal, [], [], [], fs, 'centered');

% Grafikler
figure;
subplot(2,1,1);
plot(t, AMI_NRZ_signal, 'b', 'LineWidth', 1.5);
xlabel('Zaman (s)'); ylabel('Genlik'); title('AMI-NRZ Sinyali');
ylim([-1.2 1.2]); grid on;

subplot(2,1,2);
plot(t, AMI_RZ_signal, 'r', 'LineWidth', 1.5);
xlabel('Zaman (s)'); ylabel('Genlik'); title('AMI-RZ Sinyali');
ylim([-1.2 1.2]); grid on;

figure;
subplot(2,1,1);
plot(f_AMI_NRZ, 10*log10(PSD_AMI_NRZ), 'b', 'LineWidth', 1.5);
xlabel('Frekans (Hz)'); ylabel('PSD (dB/Hz)'); title('AMI-NRZ Güç Spektral Yoğunluğu');
grid on;

subplot(2,1,2);
plot(f_AMI_RZ, 10*log10(PSD_AMI_RZ), 'r', 'LineWidth', 1.5);
xlabel('Frekans (Hz)'); ylabel('PSD (dB/Hz)'); title('AMI-RZ Güç Spektral Yoğunluğu');
grid on;
```

4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi)

Örnek : Bir ikili bilgi işareti, [1 0 1 0 0 0 1 1] değerlerini almaktadır. Bu bilgi işaretine göre AMI-NRZ ve AMI-RZ işaretlerinin güç spektrumunu oluşturan bir Matlab programı yazınız.



4. Çift-Kutuplu İşaretleşme (Ters İm Değişimi)

Örnek : Bir ikili bilgi işareti, [1 0 1 0 0 0 1 1] değerlerini almaktadır. Bu bilgi işaretine göre AMI-NRZ ve AMI-RZ işaretlerinin güç spektrumunu oluşturan bir Matlab programı yazınız.

- **AMI-NRZ:**
  - "1" bitleri alternatif olarak +1+1+1 ve −1-1−1 değerlerine sahiptir.
  - "0" bitleri sıfır seviyesinde kalır.
  - Bant genişliği NRZ'ye kıyasla daha dar olabilir.
- **AMI-RZ:**
  - "1" bitleri sürenin ilk yarısında (+/-) seviyesindedir, ikinci yarısında sıfıra iner.
  - "0" bitleri her zaman sıfırdır.
  - Daha geniş bant spektrumuna sahiptir.
- **pwelch Fonksiyonu:**
  - Welch metodu kullanılarak güç spektral yoğunluğu hesaplanır.
  - Daha pürüzsüz bir spektrum elde edilir.

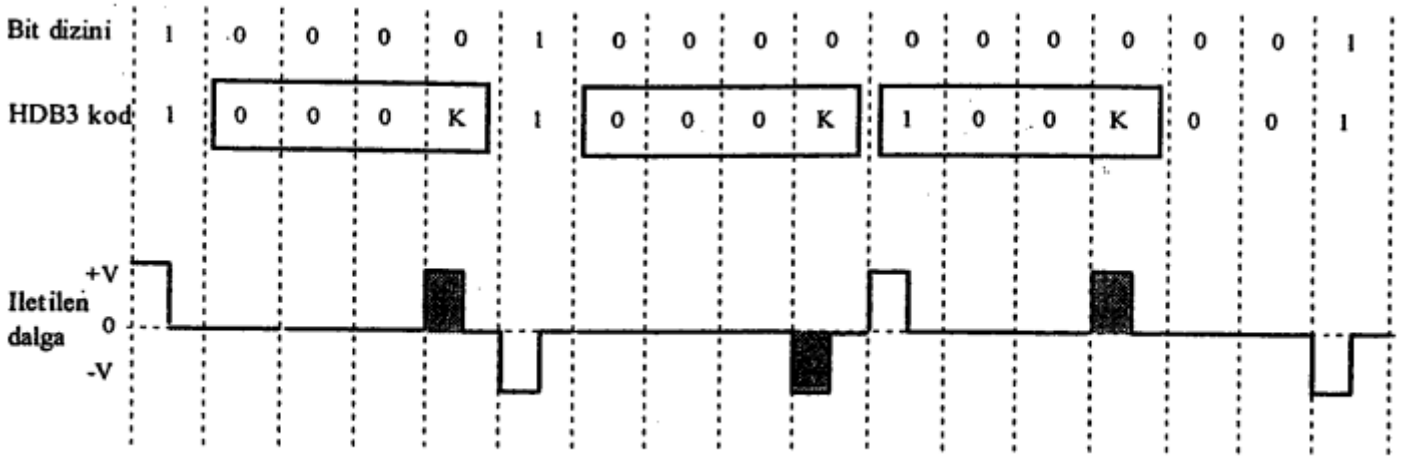
Bu MATLAB kodunu çalıştırarak **AMI-NRZ ve AMI-RZ sinyallerinin zaman domeninde nasıl göründüğünü** ve **frekans spektrumlarını** gözlemleyebilirsiniz.

5. Yüksek Yoğunluklu Çift-Kutuplu İşaretleşme (HDBn)

- Yüksek-yoğunluklu çift-kutuplu (HDBn) işaretleşmede arka arkaya gelen 0 sembolü miktarı n sayısını geçtiğinde bu semboller özel bir kod ile değiştirilerek uzun süreli 0 sembollerinin iletimine engel olunmaktadır.
- En yaygın kullanılan ve ITU-T tarafından 2 Mbit/s, 8 Mbit/s ve 34 Mbit/s çoğullanmış PCM sistemleri için standartlaştırılan HDB3 işaretleşmesinde arka arkaya gelen 0 sembolü miktarının üçü geçmesine izin verilmemektedir.
- HDB3'de bilgi işaretindeki her dört adet ardışık 0 sembolü, özel bir kod ile değiştirilmektedir.
- HDB3 işaretleşmesinde 0000 sembolleri verici (veya kodlayıcı) tarafından, duruma göre 000K veya 100K kodları ile değiştirilmektedir.
- Burada K, kural ihlalini, yani ters im kuralını ihlal eden bir adet 1 sembolünü göstermektedir.

5. Yüksek Yoğunluklu Çift-Kutuplu İşaretleşme (HDBn)

- 0000 sembollerinin değiştirildiği 000K veya 100K kodundan hangisinin kullanılacağına Şekil 4.11'de gösterildiği gibi, ters im kuralını ihlal eden K darbelerinin ortalaması sıfırlanacak şekilde karar verilmektedir.
- Dolayısıyla birbirini takip eden iki K darbesi farklı işarette olacak şekilde 000K veya 100K kodu tercih edilmektedir.
- Verici tarafından kasıtlı olarak yapılan kural ihlalleri (K'lar) kendi aralarında ters im kuralını sağlayacak şekilde yerleştirilmekte ve bu sayede sıfır işaret ortalaması korunmakta ve hata algılama özelliği devam etmektedir.



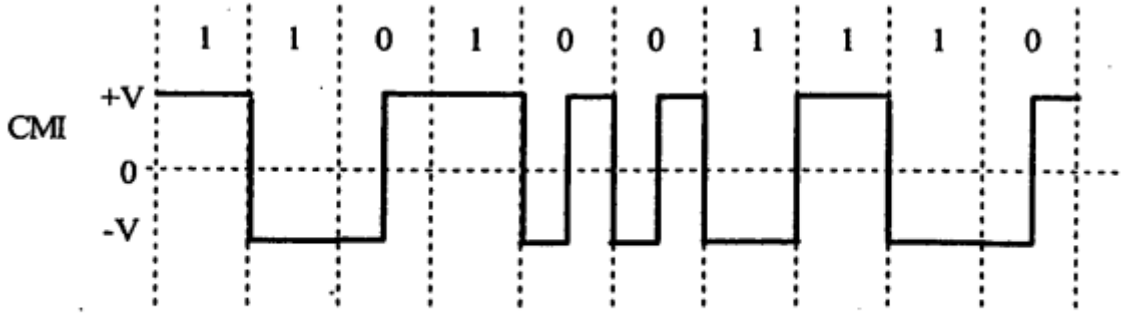
Şekil 4.11. Örnek bir HDB3 hat kodlaması.

5. Yüksek Yoğunluklu Çift-Kutuplu İşaretleşme (HDBn)

- 0000 sembollerinin değiştirildiği 000K veya 100K kodundan hangisinin kullanılacağına Şekil 4.11'de gösterildiği gibi, ters im kuralını ihlal eden K darbelerinin ortalaması sıfırlanacak şekilde karar verilmektedir.
- Dolayısıyla birbirini takip eden iki K darbesi farklı işarette olacak şekilde 000K veya 100K kodu tercih edilmektedir.
- Verici tarafından kasıtlı olarak yapılan kural ihlalleri (K'lar) kendi aralarında ters im kuralını sağlayacak şekilde yerleştirilmekte ve bu sayede sıfır işaret ortalaması korunmakta ve hata algılama özelliği devam etmektedir.

6. Kod İm Değişimi İşaretleşme (CMI)

- Kod im değişimi (CMI) işaretleşmede, Şekil 4.12'de gösterildiği gibi
- 0 sembolü *negatif gerilimden bit zaman diliminin yarısında pozitif* gerilime geçen bir dalga biçimi ile,
- 1 sembolü ise ters im kuralına uygun olarak sürekli değişerek sırayla pozitif ve negatif gerilim seviyelerindeki darbeler ile gösterilmektedir.
- Bu nedenle CMI, 0 sembolü için Manchester kodu, 1 sembolü için AMI kodunun kullanılması şeklinde düşünülebilmektedir. CMI işaretleşmesi, ITU-T tarafından 140 Mbit/s oranında çoğullanmış PCM için önerilmektedir.



Şekil 4.12. Örnek bir CMI işaretleşme dalga biçimi.

7. İkili Sembollerin Üçlü Kodlanması (nBmT)

- **4 ikili sembol → 3 üçlü sembole** eşlenir.
- **DC dengesini sağlamak için** bir önceki kod sözcüğüne göre polarite belirlenir.
- **4B3T Kodlaması:**
  - **4 ikili sembol → 3 üçlü sembole eşlenir.**
  - **DC dengesini sağlamak için bir önceki kod sözcüğüne göre polarite belirlenir.**
  - **000 kodu kullanılmaz, böylece alıcı zamanlama bilgisini koruyabilir.**
  - **PCM sistemine kıyasla %25 bant genişliği kazancı sağlar.**
  - **Kullanım Alanı: Alman Telekomu'nun ISDN şebekesi.**
- **6B4T Kodlaması:**
  - **6 ikili sembol → 4 üçlü sembole eşlenir.**
  - **İletim oranı PCM sistemine göre %33 daha verimlidir (2/3 oranında).**
  - **Daha karmaşık ve maliyetlidir.**
- **Genel Avantajlar:**
  - **DC bileşen sıfırlanır → Uzun mesafeli iletimde avantaj sağlar.**
  - **Çapraz-karışma gürültüsü azalır ve yineleyici aralığı artar.**
  - **Bant genişliği tasarrufu sağlar.**

İkili Kod Sözcüğü	Üçlü Kod Sözcüğü bir önceki üçlü kod sözcüğünün polaritesi	
	-3, -2,-1 veya 0 ise	1, 2, veya 3 ise
0000	+ 0 −	+ 0 −
0001	− +0	− +0
0010	0 − +	0 − +
0011	+ −0	+ −0
0100	0 + −	0 + −
0101	− 0 +	− 0 +
0110	00 +	00 −
0111	0 + 0	0 − 0
1000	+ 00	− 00
1001	+ + −	− − +
1010	+ − +	− + −
1011	− + +	+ − −
1100	0 + +	0 − −
1101	+ 0 +	− 0 −
1110	+ +0	− −0
1111	+ + +	− − −

Tablo 4.1. 4B3T kodlamasına örnek.

7. İkili Sembollerin Üçlü Kodlanması (nBmT)

- **4 ikili sembol → 3 üçlü sembole** eşlenir.
- **DC dengesini sağlamak için** bir önceki kod sözcüğüne göre polarite belirlenir.
- **4B3T Kodlaması:**
  - **4 ikili sembol → 3 üçlü sembole eşlenir.**
  - **DC dengesini sağlamak için bir önceki kod sözcüğüne göre polarite belirlenir.**
  - **000 kodu kullanılmaz, böylece alıcı zamanlama bilgisini koruyabilir.**
  - **PCM sistemine kıyasla %25 bant genişliği kazancı sağlar.**
  - **Kullanım Alanı: Alman Telekomu'nun ISDN şebekesi.**
- **6B4T Kodlaması:**
  - **6 ikili sembol → 4 üçlü sembole eşlenir.**
  - **İletim oranı PCM sistemine göre %33 daha verimlidir (2/3 oranında).**
  - **Daha karmaşık ve maliyetlidir.**
- **Genel Avantajlar:**
  - **DC bileşen sıfırlanır → Uzun mesafeli iletimde avantaj sağlar.**
  - **Çapraz-karışma gürültüsü azalır ve yineleyici aralığı artar.**
  - **Bant genişliği tasarrufu sağlar.**

İkili Kod Sözcüğü	Üçlü Kod Sözcüğü bir önceki üçlü kod sözcüğünün polaritesi	
	-3, -2,-1 veya 0 ise	1, 2, veya 3 ise
0000	+ 0 −	+ 0 −
0001	− +0	− +0
0010	0 − +	0 − +
0011	+ −0	+ −0
0100	0 + −	0 + −
0101	− 0 +	− 0 +
0110	00 +	00 −
0111	0 + 0	0 − 0
1000	+ 00	− 00
1001	+ + −	− − +
1010	+ − +	− + −
1011	− + +	+ − −
1100	0 + +	0 − −
1101	+ 0 +	− 0 −
1110	+ +0	− −0
1111	+ + +	− − −

Tablo 4.1. 4B3T kodlamasına örnek.

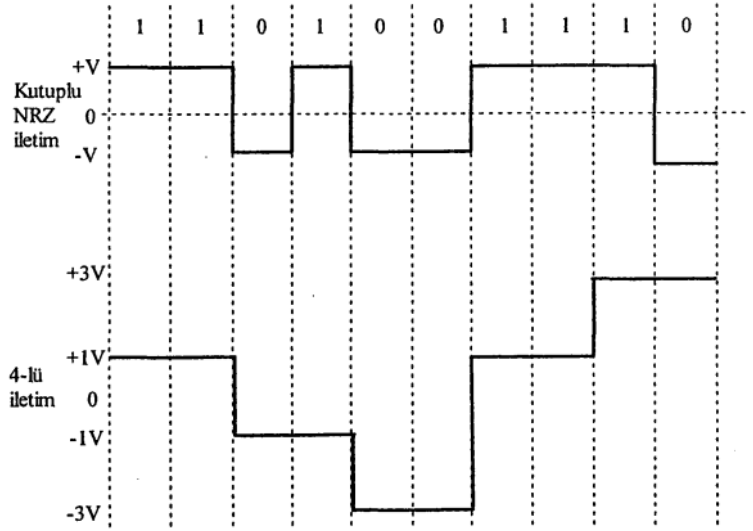
8. Çok-Seviyeli İşaretleme

- 4 İkili sistem (0,1) yerine, ikiden fazla sembol kullanan iletim sistemidir.
- M farklı seviyede darbeler kullanılır → M-li işaretleşme olarak adlandırılır.
- M seviyeli sinyaller kullanıldığı için M-li PAM (Pulse Amplitude Modulation) ile benzerdir.
- Bit gruplama yöntemiyle iletim oranı azaltılır → Bant genişliği düşer.
- Darbelerin zaman dilimi süresi arttığından bant genişliği tasarrufu sağlanır.
- **2B1Q Kodlaması (4PAM İletimi):**
  - 2 bit → 1 sembole kodlanır ($M = 2^2 = 4$ seviye).
  - İlk bit polariteyi, ikinci bit genlik seviyesini belirler.
  - Örnek genlikler:
    - 11 → +1V, 10 → +3V, 01 → -1V, 00 → -3V
  - Avantajlar:
    - İletim oranı yarıya düşer → Bant genişliği azalır.
    - Hat zayıflaması azalır, gürültüye karşı dayanıklılık artar.
    - ISDN şebekelerinde kullanılır (ABD ve Avrupa).
- **16PAM ve DSL:**

4 bit → 16 seviyeye kodlanır → Daha yüksek veri iletimi sağlanır.
Telefon hatları için uygundur, çapraz konuşmayı azaltır ve uzun mesafe iletime olanak tanır.

8. Çok-Seviyeli İşaretleme

- Uygulamada en yaygın olarak kullanılan çok-seviyeli iletim çeşidi, iki adet bitin gruplanması sonucunda oluşan $M = 2^k = 2^2 = 4$ farklı bilgi biti kombinasyonun iletimini dört farklı seviyeden gerçekleştirmektedir.
- Bu iletim biçimi, 2 adet ikili sembolün 1 adet dörtlü sembole kodlandığını belirten 2B1Q kodu olarak veya dört farklı genlik seviyesinden iletim yapıldığını belirten 4PAM iletimi olarak adlandırılmaktadır.
- Şekil 4.13'de 2B1Q kodu ile iletime bir örnek gösterilmektedir.



Şekil 4.13. Çok-seviyeli iletime örnek: 2B1Q iletimi.