

## Wireless Communication Experiments

### Ex. No. 1

#### Aim

Write a code to analyze the performance of Quadrature Amplitude Modulation (QAM) and M-ary Phase Shift Keying (PSK) scheme in AWGN channel, and compare the results with theoretical results.

#### Algorithm

Step1: Generate random binary data

Step2: Construct the constellation symbols for the generated binary sequence

Step3: Generate complex normal random variable for additive white Gaussian noise (AWGN)

Step4: Add AWGN noise with the transmitted symbol

Step5: Construct maximum likelihood (ML) receiver to decode the symbol from the received signal

Step6: Compare the decoded sequence with the original sequence to estimate the number of errors

Step7: Average bit error rate (ABER) can be calculated by computing the ratio between total number of errors and total number of bits

#### Matlab Code:

```
clc;
close all;
clear all;
% Number of information bits
m= 10^5;
%Range of SNR values
snr_dB = [0:1:10];
for j=1:length(snr_dB)
n_err = 0;
n_bits = 0;
while n_err < 100
% Generate sequence of binary bits
inf_bits=round(rand(1,m));
% BPSK Constellation symbols
x=-2*(inf_bits-0.5);
% Noise variance
N0=1/10^(snr_dB(j)/10);
% Send over Gaussian Link to the receiver
y=x + sqrt(N0/2)*(randn(1,length(x))+i*randn(1,length(x)));
% Decision making at the Receiver
est_bits=y < 0;
% Calculate Bit Errors
diff=inf_bits-est_bits;
```

```

n_err=n_err+sum(abs(diff));
n_bits=n_bits+length(inf_bits);
end
% Calculate Bit Error Rate
BER(j)=n_err/n_bits;
end
% AWGN Theoretical BER
theoryBerAWGN=0.5*erfc(sqrt(10.^(snr_dB/10)));
semilogy(snr_dB,theoryBer,'-', 'LineWidth',2);
hold on;
semilogy(snr_dB,BER,'or','LineWidth',2);
hold on;
semilogy(snr_dB,theoryBerAWGN,'blad-', 'LineWidth',2);
legend('AWGN Simulated', 'AWGN Theoretical');
axis([0 20 10^-5 0.5]);
xlabel('SNR (dB)');
ylabel('BER');
grid on;

```

Compare the simulation results with the theoretical results

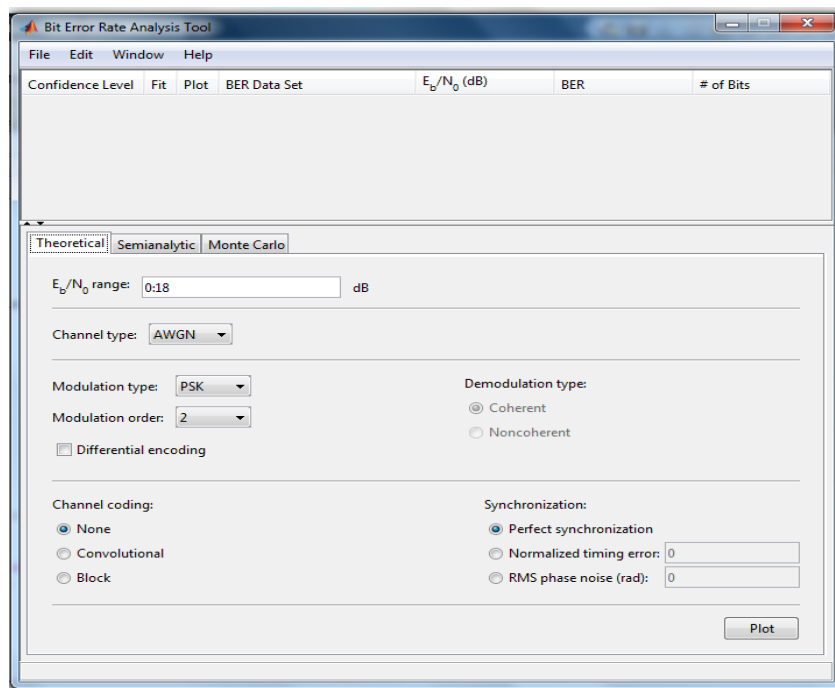
Repeat the procedure for different modulation schemes and plot the performance curve (ABER Vs SNR)

Compare and discuss the performance of different modulation schemes

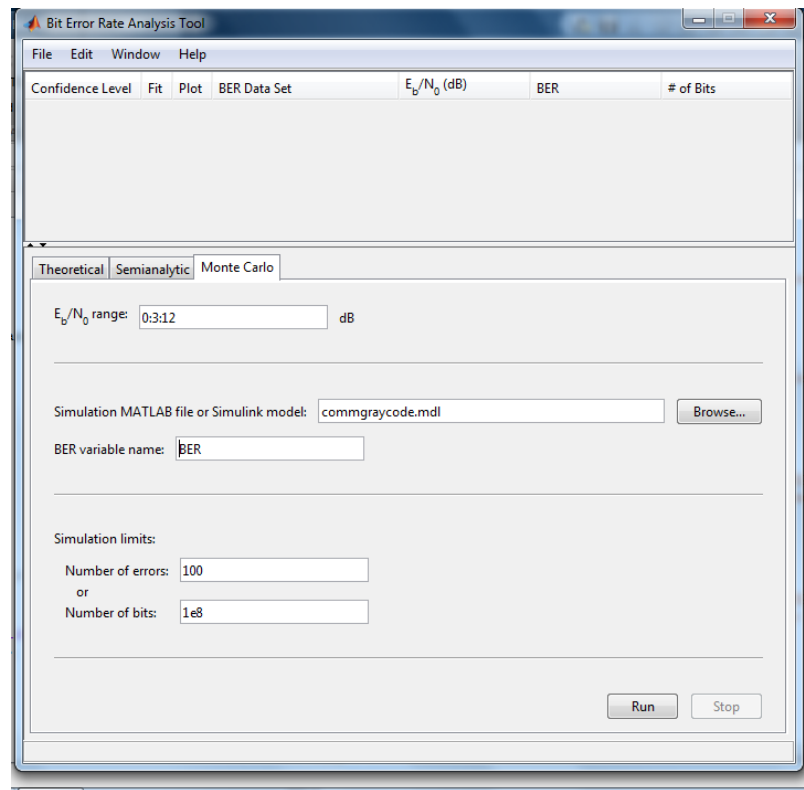
### Simulink Model:

#### Procedure:

- Type Simulink in the command window to open Simulink
- Construct the communication model in a new Simulink file
- Type bertool in a command window

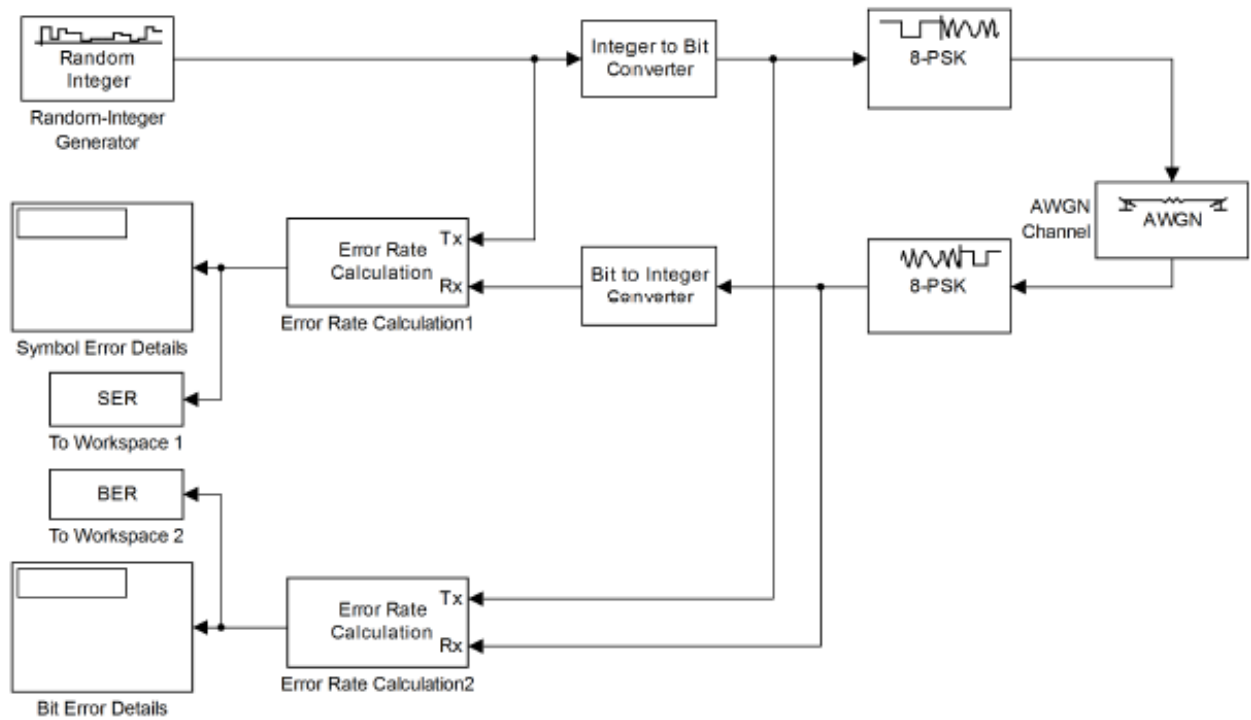


- Select theoretical option and choose the corresponding information for your modulation scheme
- Plot the theoretical curve



- Then, select Monte Carlo option and link the path of your saved Simulink model
- Plot the simulation curve and compare the simulation results with the theoretical results

## Gray Coded 8-PSK



**Results:**

## Ex. No. 2

### Aim

Write a code to compute Bit Error Rate (BER) for different digital modulation scheme in frequency-flat and slowly varying fading channel.

### Algorithm

Step1: Generate random binary data

Step2: Construct the constellation symbols for the generated binary sequence

Step3: Generate complex channel fading coefficient

Step3: Generate complex normal random variable for additive white Gaussian noise (AWGN)

Step4: Add AWGN noise with the transmitted symbol

Step5: Construct maximum likelihood (ML) receiver to decode the symbol from the received signal

Step6: Compare the decoded sequence with the original sequence to estimate the number of errors

Step7: Average bit error rate (ABER) can be calculated by computing the ratio between total number of errors and total number of bits

Repeat the procedure for different modulation schemes and plot the performance curve ABER Vs SNR

### Matlab Code:

```
clc;
close all;
clear all;
% Number of information bits
m= 10^5;
%Range of SNR values
snr_dB = [0:1:20];
for j=1:1:length(snr_dB)
n_err = 0;
n_bits = 0;
while n_err < 100
% Generate sequence of binary bits
inf_bits=round(rand(1,m));
% BPSK modulator
x=-2*(inf_bits-0.5);
% Noise variance
N0=1/10^(snr_dB(j)/10);
% Rayleigh channel fading
h=1/sqrt(2)*[randn(1,length(x)) + i*randn(1,length(x))];
% Send over Gaussian Link to the receiver
```

```

y=h.*x + sqrt(N0/2)*(randn(1,length(x))+i*randn(1,length(x)));
% decision metric
y=y./h;
% Decision making at the Receiver
est_bits=y < 0;
% Calculate Bit Errors
diff=inf_bits-est_bits;
n_err=n_err+sum(abs(diff));
n_bits=n_bits+length(inf_bits);
end
% Calculate Bit Error Rate
BER(j)=n_err/n_bits;
end
% Rayleigh Theoretical BER
snr = 10.^(snr_dB/10);
theoryBer=0.5.*(1-sqrt(snr./(snr+1)));
% AWGN Theoretical BER
theoryBerAWGN=0.5*erfc(sqrt(10.^(snr_dB/10)));
semilogy(snr_dB,theoryBer,'-', 'LineWidth',2);
hold on;
semilogy(snr_dB,BER,'or', 'LineWidth',2);
hold on;
semilogy(snr_dB,theoryBerAWGN,'blad-', 'LineWidth',2);
legend('Rayleigh Theoretical','Rayleigh Simulated', 'AWGN Theoretical');
axis([0 20 10^-5 0.5]);
xlabel('SNR (dB)');
ylabel('BER');
grid on;

```

**Results:**

### Ex. No. 3

#### Aim

Bit error rate analysis of digital communication receivers with Maximal Ratio Combining (MRC) receive diversity in frequency-flat and slowly varying fading channel.

#### Matlab Code:

```
% BER performance of BPSK receiver in Rayleigh fading channel, with 2 branch MRC receive diversity
clc;
clear all;
close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Initialization %%%%%%%%%%%%%
N=5; % Number of trials
m = 10^6; % Number of bits in each trial
ip = rand(1,m)>0.5; % Generated bits
BPSK = 2*ip-1; % Generated BPSK symbols
snr_dB = 0:1:15; % range of snr values
snr = 10.^(snr_dB/10); % snr value in the normal scale
L=2; % Number of diversity branches
% theoretical BER value for MRC combiner with 2 diversity branches
p_R_MRC = 1/2 - 1/2*(1+1./snr).^(-1/2);
ber_MRC_ana = p_R_MRC.^2.*(1+2*(1-p_R_MRC));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Receive MRC one by Two System %%%%%%%%%%%%%
n_err=zeros(1,length(snr_dB)); % Initialize the bit error counter
for p = 1:N
    for q = 1:length(snr_dB)
        % Generate white noise samples
        No = 1/sqrt(2)*[randn(L,m) + 1j*randn(L,m)];
        % Generate channel coefficient
        h = 1/sqrt(2)*[randn(L,m) + 1j*randn(L,m)];
        symbol = kron(ones(L,1),BPSK); % array of symbols
        rec_vector = h.*symbol + 10^(-snr_dB(q)/20)*No;% received symbol
        % Decision metric
        dec_metric = sum(conj(h).*rec_vector,1)./sum(h.*conj(h),1);
        ip_hat = real(dec_metric)>0; % Estimated symbol
        n_err(q) = n_err(q)+size(find([ip- ip_hat]),2); % compare input and estimated symbols
    end
end
ber_MRC_sim = n_err/(N*m);
semilogy(snr_dB,ber_MRC_ana,'-r','LineWidth',2)
hold on;
semilogy(snr_dB,ber_MRC_sim,'ob','LineWidth',2)
```

#### Results:

## Ex. No. 4

### Aim

Bit error rate analysis of digital communication receivers with Equal Gain Combining (EGC) receive diversity in frequency-flat and slowly varying fading channel.

### Matlab Code:

```
% BER performance of BPSK receiver in Rayleigh fading channel, with 2 branch EGC receive diversity
```

```
clc;
```

```
close all;
```

```
clear all;
```

```
% Number of information bits
```

```
m= 10^3;
```

```
%Range of SNR values
```

```
snr_dB = [0:1:20];
```

```
for j=1:1:length(snr_dB)
```

```
    n_err = 0;
```

```
    n_bits = 0;
```

```
while n_err < 100
```

```
inf_bits=round(rand(1,m));
```

```
    % BPSK modulator
```

```
    x=-2*(inf_bits-0.5);
```

```
    % Noise variance
```

```
    N0=1/10^(snr_dB(j)/10);
```

```
    n1 = sqrt(N0/2)*abs((randn(1,length(x)) + i*randn(1,length(x)))); %noise for the first
```

```
    n2 = sqrt(N0/2)*abs((randn(1,length(x)) + i*randn(1,length(x)))); %noise for the first
```

```
    h1 = sqrt(0.5)*abs((randn(1,length(x)) + i*randn(1,length(x)))); %rayleigh amplitude 1
```

```
    h2 = sqrt(0.5)*abs((randn(1,length(x)) + i*randn(1,length(x)))); %rayleigh amplitude 1
```

```
%Equal Gain combining
```

```
    y1 = h1.*x+n1; % Signal 1
```

```
    y2 = h2.*x+n2; % Signal 2
```

```
    y_equal = 0.5*(y1+y2);
```

```
%    dec_metric=(norm(y_equal- h1*x-h2*x))^2;
```

```
% Decision making at the Receiver
```

```
    est_bits=y_equal < 0;
```

```
% Calculate Bit Errors
```

```
    diff=inf_bits-est_bits;
```

```
    n_err=n_err+sum(abs(diff));
```

```
    n_bits=n_bits+length(inf_bits);
```

```
end
```

```
% Calculate Bit Error Rate
```

```
    BER(j)=n_err/n_bits;
```

```
end
```

```
semilogy(snr_dB,BER,'or-','LineWidth',2);
```

```
legend('Rayleigh EGC Simulated', 'Rayleigh Theoretical');
```

```
axis([0 20 10^-5 1]);
```

```
xlabel('SNR (dB)');
```

```
ylabel('BER');
```

```
grid on;
```

### Results



## Ex. No. 5

### Aim

To study the performance of the BPSK direct sequence spread spectrum system in AWGN channel.

### Matlab Code:

```
clc;
close all;
clear all;
% Number of information bits
m= 10;
f_data=1;      % DATA FREQUENCY
f_chip=7;      % LENGTH OF CHIP SEQUENCE
fc=210;        % RELATIVE CARRIER FREQUENCY
fs=fc*3;       % SAMPLING FREQUENCY
N=fs/f_chip;   % CODING RATE
%Range of SNR values
snr_dB = [0:2:26];
for jj=1:1:length(snr_dB)
n_err = 0;
n_bits = 0;
while n_err < 100
% Generate sequence of binary bits
inf_bits=round(rand(1,m));
PN_sequence=round(rand(11,1)); % GENERATION OF PN SEQUENCE
%Spread the information bits with PN sequence
j=1;
for i = 1:m
    for k = j:j+f_chip-1
        msg_spread(k)=inf_bits(i);
    end;
    msg_spread(j:(j+f_chip-1)) = xor(msg_spread(j:(j+f_chip-1)),PN_sequence(1:f_chip));
    j = f_chip*i+1;
end;
len_msg_spr=length(msg_spread);
% MODULATING THE SPREAD MESSAGE
% BPSK Constellation symbols
x=-2*(msg_spread-0.5);
% Noise variance
N0=1/10^(snr_dB(jj)/10);
% Send over Gaussian Link to the receiver
y=x + sqrt(N0/2)*(randn(1,length(x))+i*randn(1,length(x)));
% Decision making at the Receiver
msg_demod=y < 0;
% CORRELATING WITH THE PN SEQUENCE
j=1;
for i = 1:m
    msg_demod(j:(j+f_chip-1)) = xor(msg_demod(j:(j+f_chip-1)),PN_sequence(1:f_chip));
    j = f_chip*i+1;
end;
% DESPREADING THE RECEIVED SIGNAL
j=1;
```

```

for i = 1:m
    s1=0;
    for k = j:j+f_chip-1
        s1=s1+msg_demod(k);
    end;
    if (s1>=6)
        msg_demod_rec(i)=1;
    else
        msg_demod_rec(i)=0;
    end;
    j = f_chip*i+1;
end;
% Calculate Bit Errors
diff=inf_bits-msg_demod_rec;
n_err=n_err+sum(abs(diff));
n_bits=n_bits+length(inf_bits);
end
% Calculate Bit Error Rate
BER(jj)=n_err/n_bits
end
semilogy(snr_dB,BER,'or-','LineWidth',2);
legend('AWGN Simulated');
axis([0 26 10^-4 1]);
xlabel('SNR (dB)');
ylabel('BER');
grid on;

```

**Results:**

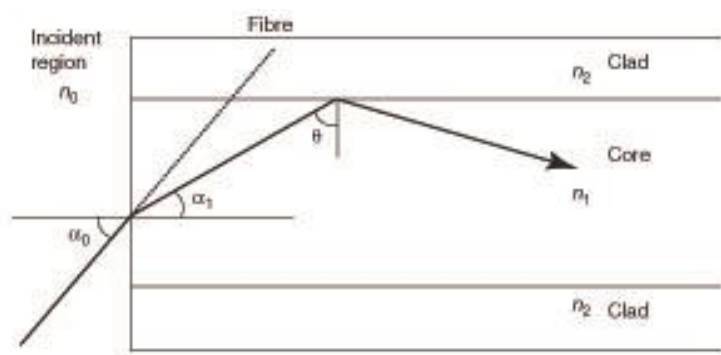
## Ex. No. 6

**Aim** To determine the Numerical aperture, Attenuation (propagation) loss, Bending loss and coupling loss and of a given fiber using OFT. Measurement of bit rate using digital link and study of TDM.

**Fiber-optic communication** is a method of transmitting information from one place to another by sending pulses of light through an optical fibre. The light forms an electromagnetic carrier wave that is modulated to carry information. First developed in the 1970s, fibre-optic communication systems have revolutionized the telecommunications industry and have played a major role in the advent of the Information Age.

The process of communicating using fibre-optics involves the following basic steps: Creating the optical signal involving the use a transmitter, relaying the signal along the fibre, ensuring that the signal does not become too distorted or weak, receiving the optical signal, and converting it into an electrical signal. Light pulses move easily down the fibre-optic line because of a principle known as total internal reflection, which states that *when the angle of incidence exceeds a critical value, light cannot get out of the fibre; instead, the light bounces back in.*

The **index of refraction** is a way of measuring the speed of light in a material. Light travels fastest in vacuum, such as outer space. The actual speed of light in vacuum is 299,792 kilometers per second. Index of refraction is calculated by dividing the speed of light in vacuum by the speed of light in some other medium. The index of refraction of a vacuum is therefore 1, by definition. The typical value for the cladding of an optical fibre is 1.46. The core value is typically 1.48. The larger the index of refraction the more slowly light travels in that medium. When light traveling in a dense medium hits a boundary at a steep angle (larger than the *critical angle* for the boundary), the light will be completely reflected. This effect is used in optical fibres to confine light in the core. Light travels along the fibre bouncing back and forth off of the boundary. Because the light must strike the boundary with an angle less than the critical angle, only light that enters the fibre within a certain range of angles can travel down the fibre without leaking out. This range of angles is called the acceptance cone of the fibre. The size of this acceptance cone is a function of the refractive index difference between the fibre's core and cladding.

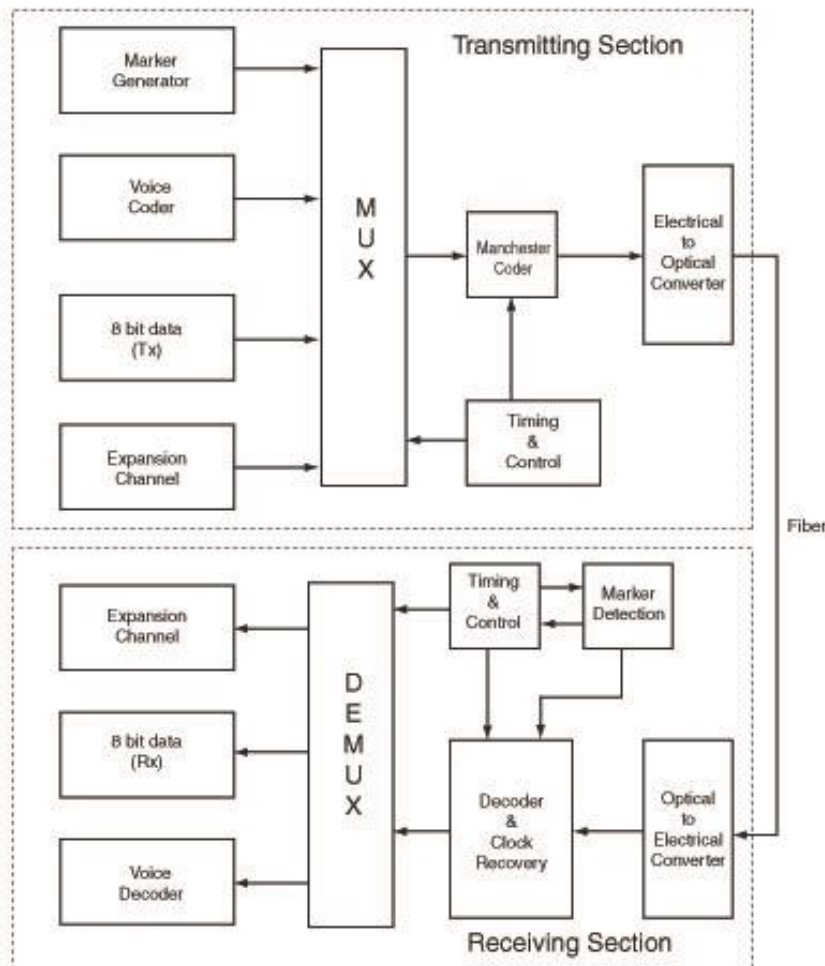


**Figure 6.1: Internal reflection in fiber**

An **Optical Fiber kit** (OFT) helps to study the principles of Fiber Optic Communication in a laboratory. It generally consists of the following elements:

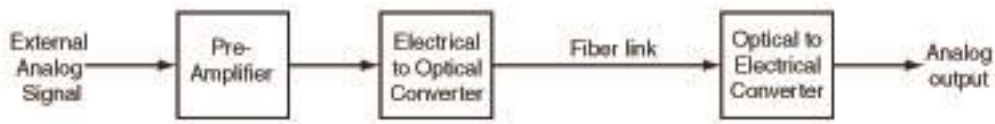
1. Optical Transmitter, including Electrical to Optical Converter

2. Optical Receiver, including Optical to Electrical Converter
3. Optical Fiber
4. Timing Recovery Unit
5. Line Coder
6. Line Decoder
7. Voice Coders and Multiplexer
8. Demultiplexer and Voice Decoders



**Figure 6.2: Block diagram of an OFT**

A typical block diagram of an OFT illustrating different elements is shown in Figure 6.2. The fibre cable used in trainer kits generally is of plastic due to its easiness of termination and the distance of application is very less. The wave lengths of the light used in OFT are typically 850 nm and 650 nm. An LED at the transmitter end converts electrical signal in to light and a photo transistor is used to convert back the light in to electrical signal at the receiver end. The fibre carries the signal in terms of light from transmitter to receiver.



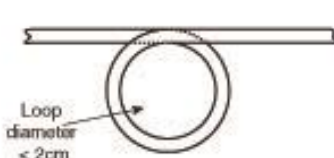
**Figure 6.3: Block diagram – Attenuation and bending loss**

Block diagram to conduct the experiment is shown in Figure 6.3. An analog signal is applied to the specified input point that is converted as light by means of an LED. A fibre cable is used to carry the light from transmitter point to the receiver. At the receiver end, the light is converted back to an electrical signal using a photo detector. The received signal can be observed on an oscilloscope and compared with the transmitted signal. Gain control knob provided in the kit can be adjusted for the required gain.

### Procedure

#### Bending loss:

- The OFT connections are made as per the instruction.
- Switch on the DC power supply.
- Apply a sinusoidal signal of 1V, 1 kHz as the message signal at the transmitting end.
- Use a One meter fiber between transmitting and receiving end.
- Check the output signal at the receiving end. Make sure that there is no bend in the fiber cable. Adjust the gain control knob to get the required output.
- Make a loop (diameter more than 1 cm) with the cable as shown in Figure 6.4 and observe the change in the output.
- Observe the output for different loop diameters and tabulate the magnitude of the output.
- Now you can have a feeling of the bending loss by observing the output versus bending diameter.
- A graph may be plotted with the output against the diameter of loop.

	Tabular column for bending loss	
	<i>Diameter of loop</i>	<i>Output</i>
 <p>Loop diameter &lt; 2cm</p> <p><b>Figure 6.4: Bending loss</b></p>	1 cm	
	1.5 cm	
	2 cm	
	2.5 cm	
	3.0 cm	

### Propagation loss:

- Now tabulate the output ( $V_1$ ) with 1 meter ( $l_1$ ) fibre without any bend.
- Replace the 1 meter fiber with a 3 meter fiber ( $l_3$ ) and tabulate the output ( $V_3$ ), without changing the input signal.
- Make sure that gain control knob is not disturbed.
- Calculate the propagation loss using the relation,  $\frac{V_3}{V_1} = e^{[-\alpha(l_3-l_1)]}$

### Coupling loss:

- Connect one end of the 1m fiber to LED (transmitter) and other end to the detector PD1 (receiver).
- Measure the detector output and designate it as  $V_1$ .
- Disconnect the fiber from the detector. (keep the other end at the transmitter itself)
- Connect one end of the fiber to the detector.
- Bring the free ends of both the cables (1m and 3m) as close as possible and align them using the fiber alignment unit as shown in figure 6.5.
- Measure the detector output and designate it as  $V_4$ .
- Compute the coupling loss using the formula  $\eta = -10 \log \left( \frac{V_4}{V_1} \right) - \alpha'(l_3 + l_1)$

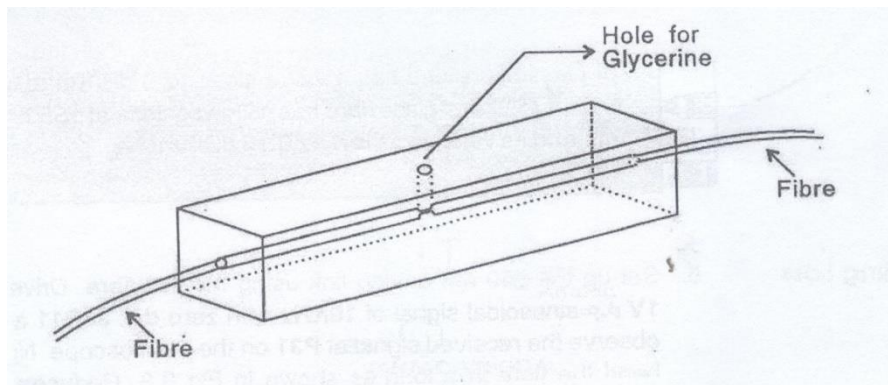


Fig. 6.5 Fiber alignment using the Alignment Unit

### Numerical Apperture

- Insert one end of the fiber into the Numerical Aperture Measurement Unit as shown in fig 6.6.
- Connect the other end of the fibre to LED2. Let  $d$  be the distance between the fiber tip and the screen.
- Measure the diameter of the circular patch of red light in two perpendicular directions (BC and DE in fig 6.7). The mean radius of the circular patch is given by  $r = \frac{DE+BC}{4}$
- Carefully measure the distance  $d$  between the tip of the fibre and the illuminated screen (OA in fig. 6.6).
- The numerical aperture is given by  $NA = \sin\theta = \frac{r}{\sqrt{d^2+r^2}}$
- Repeat the steps for different values of  $d$  and find the average value of numerical aperture.

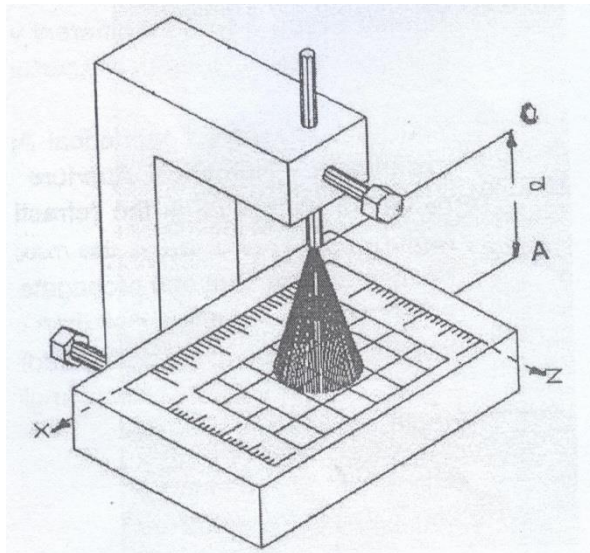


Fig: 6.6 Numerical aperture measurement setup

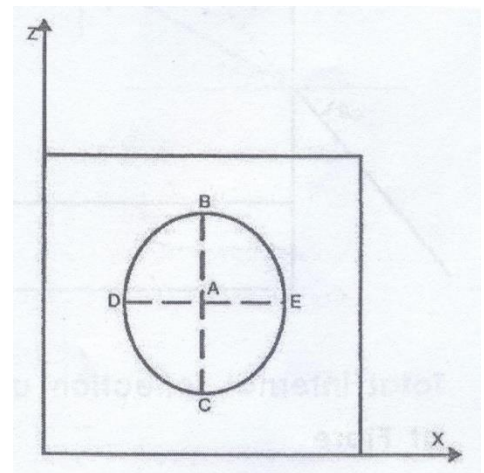


Fig 6.7 Measurement of radius

Tabular column for Numerical Aperture				
Distance $d$ (cm)	Vertical diameter (BC)	Horizontal diameter (DE)	Average Radius $r = \frac{DE + BC}{4}$	Numerical Aperture $NA = \frac{r}{\sqrt{d^2 + r^2}}$

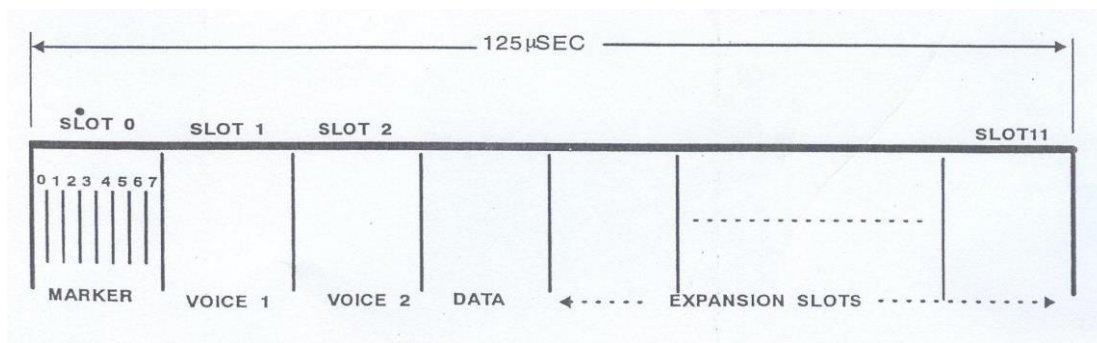
**Maximum bit rate:**

- The OFT connections are made as per the instruction.
- Switch on the DC power supply.
- Apply a TTL signal (0-5V), 1 kHz as the message signal at the transmitting end.
- Use a One meter fiber between transmitting and receiving end.
- check the output signal at the receiving end.
- Vary the frequency and observe the output.
- Find the maximum frequency the digital link can transmit
- Calculate the maximum bit rate by multiplying the maximum transmitted frequency by 2.



## TDM:

- The OFT connections are made as per the instruction.
- Switch on the DC power supply.
- Check all the jumper connections are intact.
- Connect a One meter fiber cable between transmitting and receiving end.
- Check the Marker data are in same format at the transmitter and receiver end.
- Use the data switches to set an 8-bit digital data pattern.
- Adjust the gain control knob (if any) to observe the received digital data in the LED block provided at the receiver end.
- Compare the received digital data pattern with the transmitted pattern.
- Use the voice channel to feed an analog signal of 1V at 1kHz.
- Observe the retrieved audio signal at the receiver end and compare with the original audio signal
- Observe the simultaneous transmission of digital and analog signal by changing the data pattern and audio signal simultaneously at the transmitter end.
- To observe the time switching of voice, use the voice channels with handsets instead of audio signals.
- Use mouth piece of one hand set to send the voice and use the ear piece of other to hear the voice and check the performance of TDM.
- Check whether the voice transmitted through one of the hand set can be heard on the other. If not, the system is in loop back mode.
- Make the cross connections at the appropriate block and check the performance again.
- Observe the frame time, slot time, marker data of TDM wave form and plot it on a graph sheet.



## Result:

Bending Loss (Analog Link) = \_\_\_\_\_

Propagation Loss (Analog Link) = \_\_\_\_\_

Coupling Loss (Analog Link) = \_\_\_\_\_

Numerical Aperture = \_\_\_\_\_

Maximum Bit Rate (Digital Link) = \_\_\_\_\_

Frame Time (TDM) = \_\_\_\_\_

Slot Time (TDM) = \_\_\_\_\_