TE0421 – WIRELESS COMMUNICATION LAB

Laboratory Manual

DEPARTMENT OF TELECOMMUNICATION ENGINEERING

SRM UNIVERSITY

S.R.M. NAGAR, KATTANKULATHUR – 603 203.

SRM UNIVERSITY

DEPARTMENT OF TELECOMMUNICATION ENGINEERING

TE0421- WIRELESS COMMUNICATION LAB

(2010-2011)

Revision No : 0 **PREPARED BY**

Date :

Authorized by

HOD/TCE

Department of Telecommunication Engineering
SRM UNIVERSITY

TE0421 - Wireless Communication Lab

List of Experiments

Study of wireless Communications using Communication Trainer Kits

1.a Baseband Communication
1.b. Adaptive Linear Equalizer
1. c. Code Division Multiple Access (CDMA) - Multipath
1. d Code Division Multiple Access (CDMA) – Multiuser
1. e Global System for Mobile Communication (GSM)
(Using WiCOMM-T - Wireless Digital Communication Training system – SDR Platform)
1.f. Spread Spectrum – DSSS Modulation & Demodulation
(Using Emona 101 Tranier Kit)

Wireless Path loss Computations - Study of Propagation Path loss Models : Indoor & Outdoor(Using Matlab Programming)

2.a Free Space Propagation – Path Loss Model
2.b Link Budget Equation for Satellite Communication
2.c Carrier to Noise Ratio in Satellite Communication
2.d Outdoor Propagation – Okumura Model
2.e Outdoor Propagation – Hata Model

Antenna Design Concept (using 4NEC2)

3.a Dipole Antennas
3.b Yagi – Uda Antenna – 3 element
3.c Yagi – Uda Antenna – 5 element
3.d Yagi – Uda Antenna – 7 element

STUDY OF WIRELESS COMMUNICATIONS USING COMMUNICATION TRAINER KITS

WIRELESS PATH LOSS COMPUTATIONS

STUDY OF PROPAGATION PATH LOSS MODELS: INDOOR & OUTDOOR

ANTENNA DESIGN CONCEPT (USING 4NEC2)

EXPERIMENT 1.a

BASEBAND COMMUNICATION

Aim

To setup base-band digital communication link using Raised Cosine spectrum pulses with the chosen roll-off factor, to study the characteristics of the RC pulse, to explore the behavior of the timing acquisition algorithm and to understand clock slip control in the tracking algorithm.

Concepts

- Pulse shaping
- RC pulse generation
- Pairing
- Interpolation
- Tracking algorithm
- Clock slip control

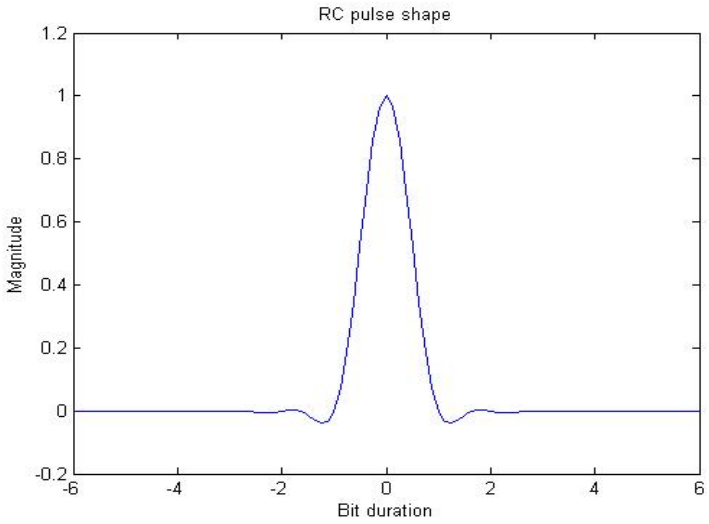
Pulse shaping

Inter Symbol Interference (ISI) is the smearing of symbols (when they reach the receiver) so that energy from one symbol affects the next one that results in improper interpretation of received signals. ISI is an unavoidable consequence of wired as well as wireless communication systems. To avoid ISI one way is to slow down the bit rate which is not an acceptable solution when there is a need to accommodate more and more data in the available band. Another way of avoiding ISI is by pulse shaping. Pulse shaping also improves the spectral efficiency without reducing the accuracy and without increasing the bandwidth. In general, when the pulse is shaped (refer figure 1.1) the spectrum consists of a main lobe representing the middle of the spectrum and various side lobes located on either side of the main lobe. By shaping the spectrum we can achieve two things. One is that the main lobe is made as narrow as possible, and the other is that the maximum side lobe level is made as small as possible relative to the main lobe. By shaping the spectrum this way the energy of one symbol will be confined to it and will not get leaked to the adjacent symbols.

RC Pulse Generation

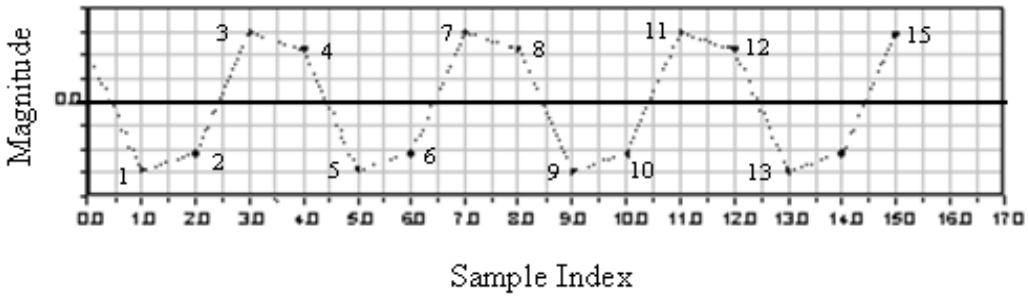
Pulse shaping using raised cosine pulse is one way to counter the effect of ISI. The pulse shaping is done at the transmitter side such that ISI effects can be reduced at the receiver. The impulse response of an ideal RC filter is given by $h(t) = \text{sinc}(2Bt)*\cos(2\pi Bt)/(1-16(rBt)^2)$ where $B=1/2T_b$ and r is the roll-off factor chosen by the user. The roll-off factor of a raised cosine pulse indicates how much bandwidth is being used over the ideal bandwidth. If the factor is smaller the scheme is more efficient. The percentage over the minimum required Nyquist bandwidth is called the excess bandwidth. It is 100% for roll off factor of 1 and 50% for a roll

off factor of 0.5 Roll off factor also governs the rate at which the tails of the pulse decay. A value of 0 offers the narrower bandwidth, but the slowest rate of decay in the time domain. When the value becomes 1 the bandwidth is more but the time domain tails decay rapidly. So roll off factor gives a trade off between increased data rate and the time domain tail suppression. Time domain decay is of prime importance for systems with relatively high timing jitter at the receiver.



Pairing

Pairing is done at the receiver to determine the pairs in the received samples which are having the ISI free instants The figure 1.2 shows samples at the receiver corresponding to a transmitted bit sequence of alternating 1 and -1. There are two samples per bit. The sample pairs (0, 1), (2, 3) and (4, 5) are having the zeros crossings between them, while the other pairs (1, 2) and (3, 4) are having the ISI –free instant between them. The task of pairing is to find out the set of sample pairs having the ISI free instants between them in a given block of received samples. To achieve this, the pairing function counts the number of zero crossings between the (even, odd) pairs as well as the (odd, even) pairs. The one which has lesser number of zero crossings is chosen as having the ISI – free instant between them.



Interpolation

Interpolation is used to acquire the ISI-free instant from the received samples. For interpolation a SINC interpolator is used to interpolate 8 instants between the samples. The impulse response of the SINC interpolator is given by $h_{1p}=\sin(wn)/(\pi n)$ where $w = \pi/L$, $L=9$ and 128 tap FIR approximation is used. So 'n' varies from -64 to 63. Then a magnitude averaging is done for all 8 instants across all the bits. These results in eye pattern get rectified and the ISI getting averaged, revealing the main lobe of the RC pulse. The sampling instant which gives the maximum determines the best instant.

Tracking algorithm

Once the best instant is found as explained above, it is enough to find only the interpolator values on adjacent positions of the best instant. So to be computationally efficient in the tracking portion the interpolation is implemented as polyphase filtering. There are 8 polyphase filters. The impulse response of the polyphase filter is

$$h_1(n) = h_{1p}(-64 + 8n)$$

$$h_2(n) = h_{2p}(-63 + 8n)$$

$$h_3(n) = h_{3p}(-62 + 8n)$$

$$h_4(n) = h_{4p}(-61 + 8n)$$

$$h_5(n) = h_{5p}(-60 + 8n)$$

$$h_6(n) = h_{6p}(-59 + 8n)$$

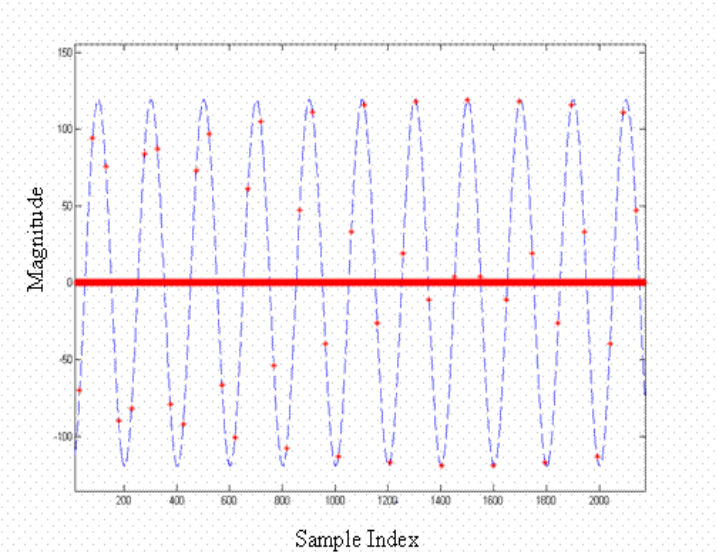
$$h_7(n) = h_{7p}(-58 + 8n)$$

$$h_8(n) = h_{8p}(-57 + 8n)$$

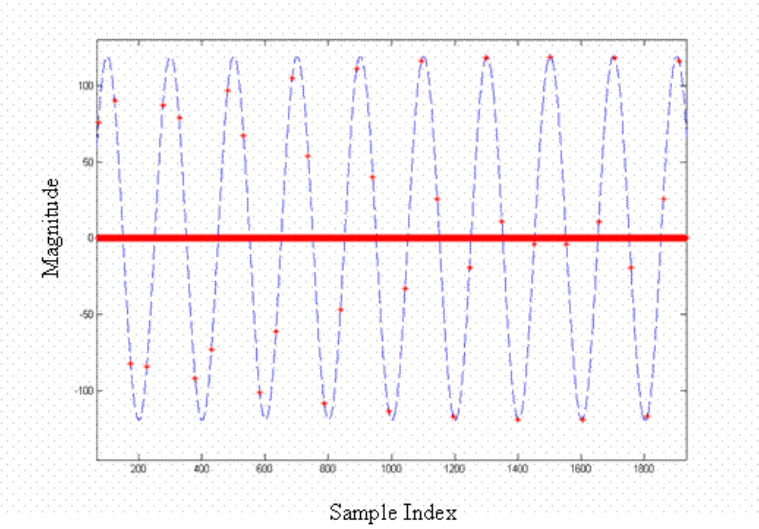
where $n = 0$ to 15 The convolution with $h_1(n)$ gives the interpolated value at the first instant, convolution with $h_2(n)$ gives the interpolated value at the second instant and so on. The bits are decoded and the best instant in each iteration is stored for plotting.

Clock slip control

If the transmitter and receiver clocks are not having any error then the best interpolator value will remain constant for all the blocks. If the receiver clock is faster than the transmitter clock, 3 samples per bit will be received instead of the usual 2 samples per bit as time progresses. In that case, for correcting the pairing, the extra one sample in the bit is dropped. So the best interpolator value will move from 0 – 1 – 2 and after remaining at 7 for some time it cannot go to 8 and hence wrapped back to 0. This is called *clock slip* and the clock correction is done by dropping the extra sample in the last block. This gets repeated periodically



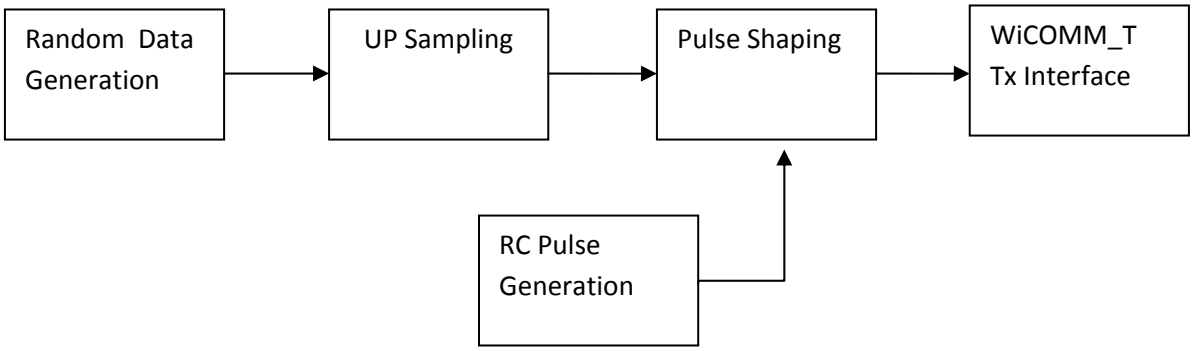
If the receiver clock is slower than the transmitter clock then one sample per bit will be received instead of the usual 2 samples per bit as time progresses. In that case, for correcting the pairing, simply reuse the sample. So the best interpolator value will move from 7 – 6 – 5 and after remaining at 0 for some time it gets wrapped back to 7. This is called *clock slip* and the clock correction is done by folding the right most sample of the previous block in the next block. This gets repeated periodically.



Receiver clock faster than the transmitter clock – extra samples per bit

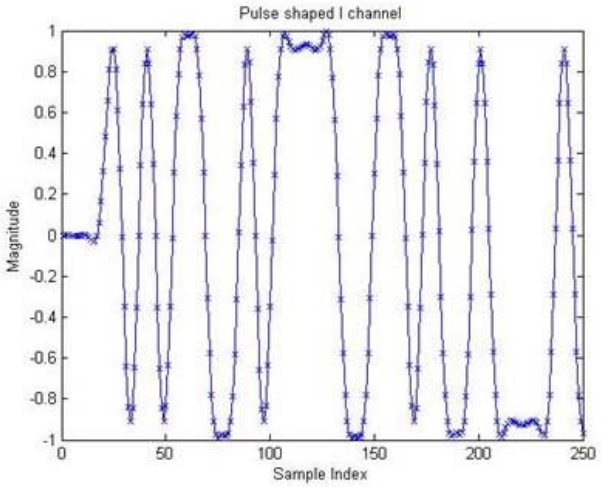
MATLAB Code implementation

Transmitter

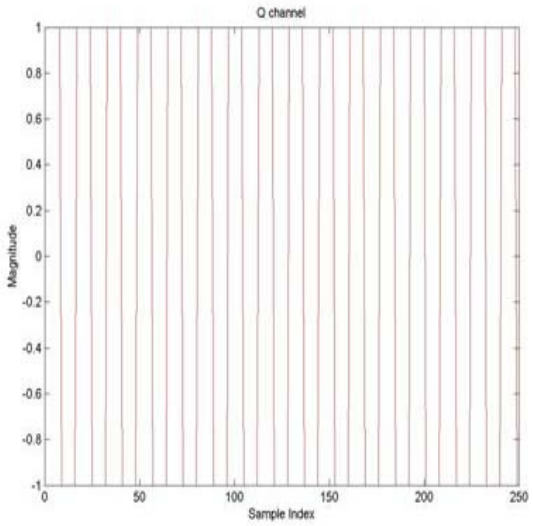


Transmitter block diagram in MATLAB

1. A RC pulse of duration $-6T_b$ to $+6T_b$ (where T_b is the bit duration) is generated.
2. Random data to be transmitted is generated.
3. The random data generated is upsampled by number of samples in one bit duration. This is the I channel data.
4. A clock at half the symbol rate is also generated in the range -127 to +127. This is the Q channel data. This is for triggering the oscilloscope for viewing the I channel data.
5. I data is convolved with the RC pulse to obtain the pulse shaped bits.
6. I and Q channel data are plotted in fig 6a and.6b
7. Pulse shaped bits are given to the WiCOMM-T Tx interface block to send it through WiCOMM-T.



Transmitter Waveform – I data



Transmitter Waveform – Q Data

Receiver

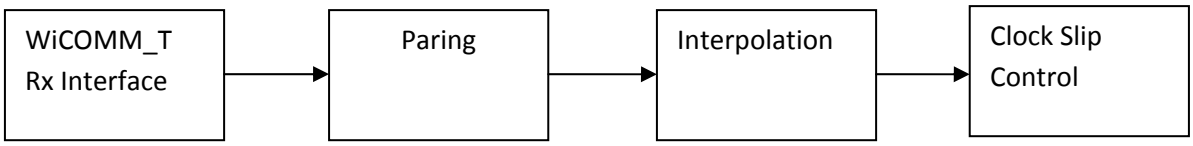


Figure 7: Receiver Block diagram in MATLAB

1. The samples are received from the WiCOMM-T Rx interface as blocks
2. To find out the ISI free sample pairs, Pairing is done.
3. To find out the ISI free instants, Interpolation is done.
4. To find out the best sampling instant, Tracking algorithm is used.
5. Clock slip correction is carried out.
6. The movement of interpolator values, best timing instant and the decision variable histogram are plotted in fig 8, 9 & 10

Procedure

1. Setup WiCOMM-T in baseband loopback. Refer Appendix A on how to setup for baseband loopback.
2. Generate the transmitter modem samples. Refer Appendix B on how to generate the modem samples.
3. Transmit the modem samples through WiCOMM-T. Refer Appendix B on how to transmit and receive the samples through WiCOMM-T.
4. Observe the eye pattern in oscilloscope. Refer Appendix A on how to connect the scope to WiCOMM-T.

5. Vary the roll-off factor from 0.11 to 0.91 and repeat steps 3 & 4. Refer Appendix B on how to vary the parameters in MATLAB.
6. For each roll-off factor, transmit and receive the modem samples through WiCOMM-T.
7. Analyze the received modem samples. Refer Appendix B on how to analyze the modem samples.
8. Observe the various plots generated by MATLAB.
9. Connect 2 WiCOMM-Ts such that one as transmitter and other as receiver at baseband level. Refer Appendix A on how to connect WiCOMM-Ts with baseband connectivity. Repeat steps 2 to 8 and observe the results.

Observation

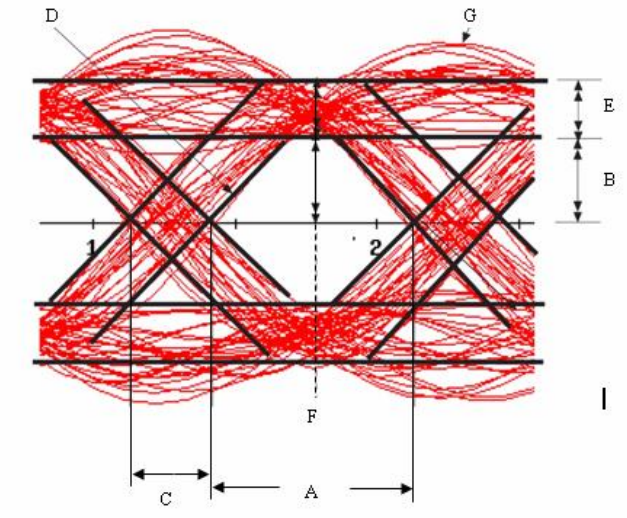
Raised Cosine Pulse:

Observe the generated raised cosine pulse shape plot for various values of roll off factor. Note the larger the value of the roll off factor the narrower the main lobe is and flatter the tails of the pulse. Larger value of roll-off factor means less susceptible to ISI and hence less susceptible to timing jitter. But larger value of roll-off factor means greater the bandwidth. Choosing the value of roll-off factor therefore involves trading off between increased bandwidth and timing jitter. Find out the optimum roll-off factor best suited for wireless communication.

Transmitted signal:

Observe the pulse shaped signal plot in I channel and the clock signal plot in Q channel. How does the pulse shaped signal vary for different values of roll off factor

EYE PATTERN:



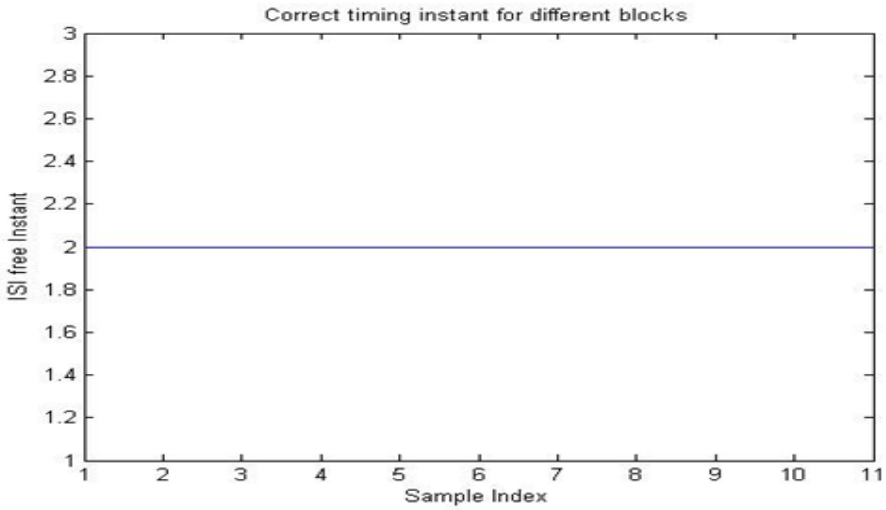
The open part eye A signifies the time that we can sample the signal with fidelity. So larger the opening of the eye the better will be the signal. For raised cosine signals larger the value of the roll-off factor larger will be the opening. The opening will be smaller for smaller values of roll-off factors and hence the error will be more if not sampled at the best timing instant which happens to be the centre of the eye. B represents the margin over noise, C represents the amount of variations or distortions that occur during zero crossings, D represents the slope of the eye which indicates the sensitivity to timing error. Smaller slope means eye will be wide open and hence the error will be less. E represents the amount of distortion at sampling instant. A smaller band means larger SNR. F represents best time to sample. G represents the wasted power. Observe the values of A to G in the eye pattern for various values of roll-off factor. What do you infer from these values when you change the roll-off factor? From this can you find out the best roll-off factor?

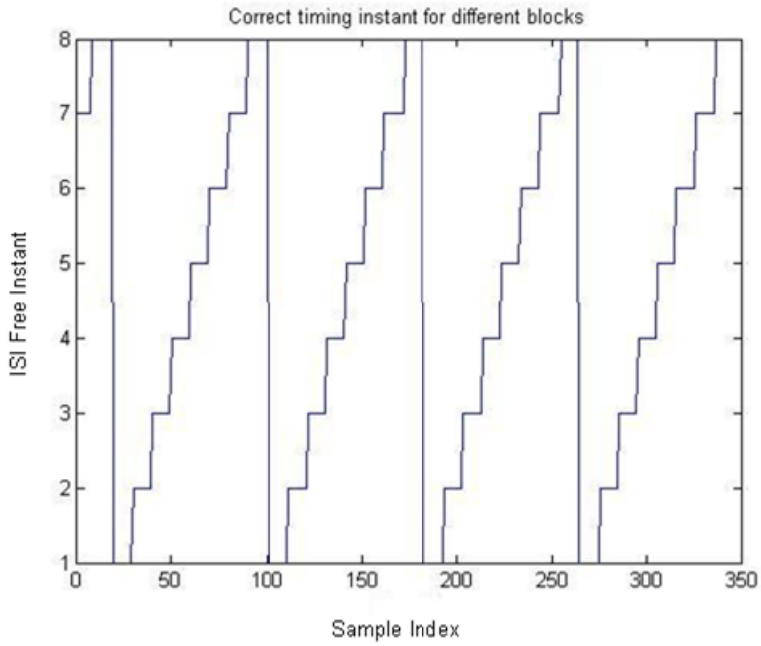
Movement of interpolator values:

The receiver MATLAB code splits the received samples into number of blocks. Using a SINC interpolator we interpolate 8 instants between the samples. Then a magnitude averaging is done for all 8 instants across all the bits. The sampling instant which gives the maximum determines the best instant.

Best timing instant:

Once the interpolation is over the next step is the clock tracking and slip control. Here the code reads block after block and finds the best timing instant. When there is no timing error between the transmitter and receiver clocks then the best timing instant will remain same for all the blocks. Observe the best timing instant plot for the baseband loop back. Connect 2 WiCOMM-Ts through baseband and observe the best timing instant. What is the difference between those two plots and why is the difference?

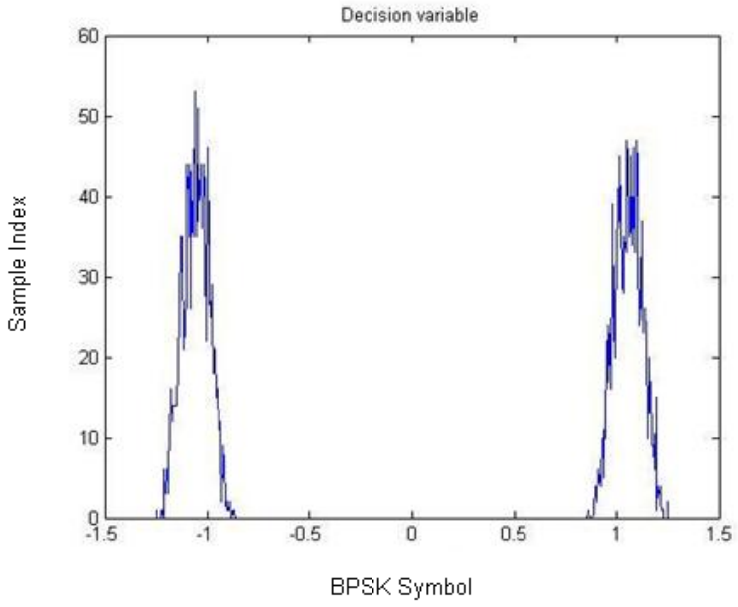




Sampling Index using two WiCOMM-Ts

Decision variables:

The histogram in figure 1.10 shows the distribution of the received decoded samples. Observe that they are grouped under +1 and -1.



Decision Variable histogram

EXPERIMENT 1.b

ADAPTIVE LINEAR EQUALIZER

Aim

To mitigate the distortion introduced by the channel on the transmitted signal using Adaptive Linear Equalizer (LE) on the received samples from ADC output.

Concepts

- Why Equalization?
- Types of Equalizers
- Fractionally spaced Equalizers
- Adaptive Linear Fractionally spaced Equalizer
- RRC Pulse generation
- Channel Modeling
- LMS rule for Adaptive linear equalizer

Description

Why Equalization?

In the previous two experiments, we discussed the transmission of digital information through a White Gaussian Noise Channel (AWGN) where the channel was assumed to have ideal response (ie. have a constant gain and phase) over the band-width of the signal. In this experiment, we consider the problem of signal transmission when the channel is band limited to some specified bandwidth of **B**Hz. A channel is said to be non-distorting or ideal if the amplitude response $|c(f)|$ is constant for all $|f| \leq B$ and the envelope delay characteristics $|\iota(f)|$ is constant for all $|f| \leq B$. Thus when the channel is ideal and the bandwidth is B, a signal pulse can be designed to allow us to transmit at 2B symbols/s without ISI and the bits that can be transmitted depend on the type of modulation technique employed. On the other hand, when the channel is not ideal, signal transmission at symbol rate equal to or exceeding 2B results in inter symbol interferences (ISI) among the adjacent symbols. In order to have a design with zero ISI, it is necessary to reduce the symbol rate 1 / T below the Nyquist rate of 2B symbols/s and hence we can realize practical transmitting and receiving filters. But to achieve a symbol transmission rate of 2B symbols/s we should relax the condition of zero ISI to have a controlled amount of ISI. In a design where the channel frequency response is known for $|f| \leq B$ then we can design a modulator and demodulator using filters whose responses may be selected to minimize the error probability at the detector. However in practical digital communication system transmitting through band-limiting channels, the frequency response of the channel c(f) is not known a priori to design an optimum filter for modulator and demodulator. We need to design a receiver in the presence of channel distortion (which is not known), AWGN and ISI to compensate for the high error rates. An equalizer is one such compensator that reduces high error rates.

Types of equalizers

There are different equalization methods available. An optimum equalization technique is available based on the maximum likelihood sequence detection (MLSE) criterion. But MLSE is computationally complex and the complexity grows exponentially with the length of the channel time dispersion. So a sub optimum equalization technique is discussed in this experiment and in the next experiment. In experiment 5, MLSE using Viterbi Algorithm is given as exercise for the students to try out. One method of doing the sub optimal detection is based on the use of a linear filter with adjustable coefficients known as Linear Equalizers. Second one is the method that uses the previously detected symbols to suppress the ISI in the present symbol being detected and is called Decision Feedback Equalizers.

Fractionally spaced Equalizers

In Linear Equalizers the equalizer taps are spaced at the reciprocal of the symbol rate ie. at the reciprocal of the signaling rate 1/T. This sampling time is optimum if the equalizer is proceeded by the filter matched to the channel distorted transmitted pulse. When the channel characteristics are unknown, the receiver filter is matched to the transmitted signal pulse and the sampling time is optimized for the filter. But the limitation of the symbol rate equalizer is that it can only compensate for the frequency response characteristics for the aliased received signals and not compensate for the channel distortion inherent in the signal. To overcome this problem we are using fractionally spaced equalizer in which the incoming signal is sampled at least as fast as the Nyquist rate. For example, if the transmitted signal consists of pulses having a raised cosine spectrum with a roll off factor r then it is passed through an equalizer with tap spacing of T / (1+r). When r is 1 we would have T/2 spaced equalizer and when r = 0.5 then would have 2T/3 and so on. In general the fractionally spaced equalizer compensates for the channel distortion in the received signal before the aliasing effects due to symbol rate sampling. In effect, the fractionally spaced equalizer is equivalent to the optimum linear equalizer consisting of the matched filter followed by a symbol rate equalizer.

Adaptive Linear Equalizers

The objective of the Adaptive Linear Equalizer is to adapt the coefficients to minimize the noise and ISI at the output. The adaptation of the equalizer is driven by the error signal which is computed using an adaptive algorithm like Least Mean Square (LMS). There are 2 modes that the Adaptive equalizers work. One is the training mode and the other is the decision directed mode. In training Mode, to make equalizer suitable in the initial acqusition duration, a training signal is needed. This is done to gather information about the channel. In this mode of operation, the transmitter generates a data symbol sequence known to the receiver. The error signal e[k] is computed from the training signal d[n]. The error signal $e[k] = d[n] - y[k]$ where $d[n] = I[k-\Delta]$. Here Δ is called as decision delay. The training mode adaptive equalizer is shown in Figure 1. The error signal generated based on the known training sequence is used initially to adjust the coefficients of the equalizer. Once the coefficients are converged to their optimum values using the training sequence, the decisions at the output of the slicer are generally sufficiently reliable so that they may be used to continue the coefficient adaptation process. This is called a decision directed mode.

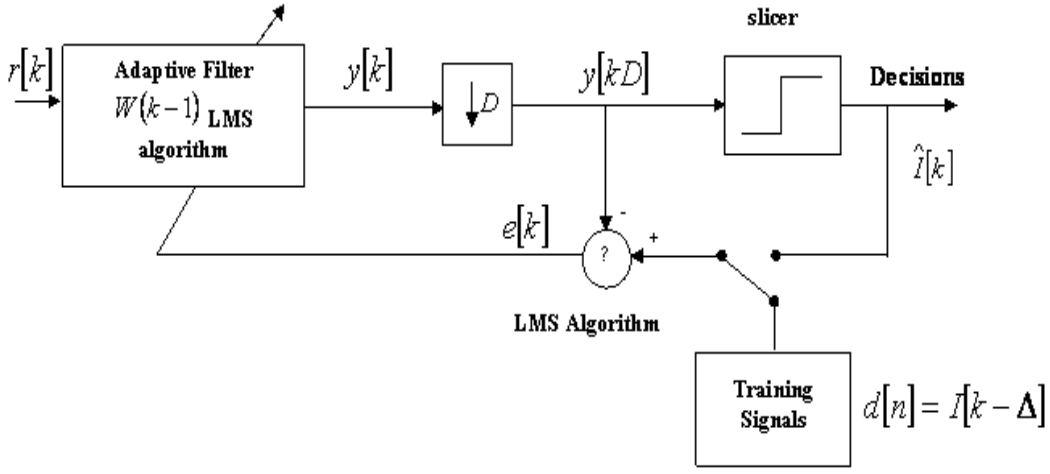


Fig 1 : Block Diagram of Adaptive Linear Equalizer – Training Sequence Mode

In Decision Directed Mode the receiver decisions are used to generate the error signal. Decision directed equalizer adjustment is effective in tracking slow variations in the channel response. However, this approach is not effective during initial acqusition. Here in WiCOMM-T the pre-distortion given in the transmitter part is equalized using an FIR filter. A fractionally spaced adaptive linear filter of order L is used for this purpose. The error signal between slicer input and output will be used to adapt the adaptive filter in the decision directed mode. Since it is a fractionally spaced equalizer, the filter operates at twice the symbol rate. The decision directed mode adaptive equalizer is shown in Figure 2

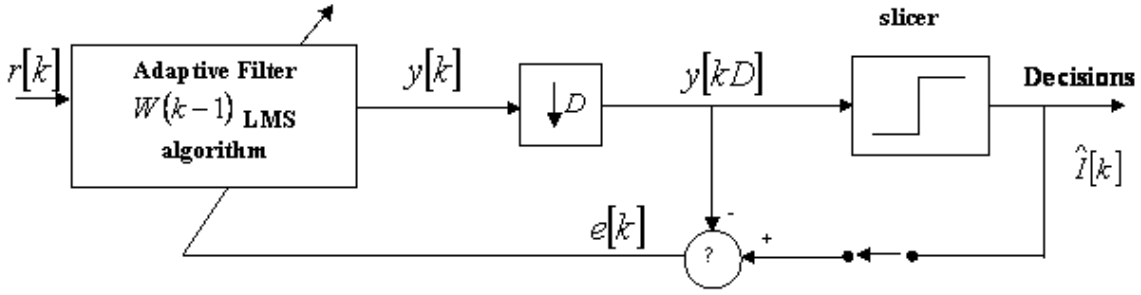


Fig 2 : Block Diagram of Adaptive Linear Equalizer – Decision Directed Mode – L tapped filter

MATLAB IMPLEMENTATION

Transmitter

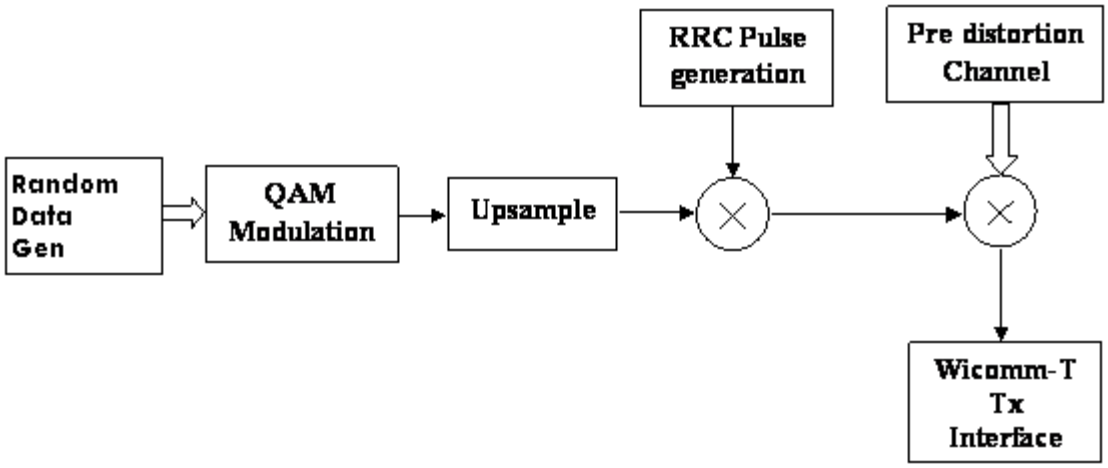
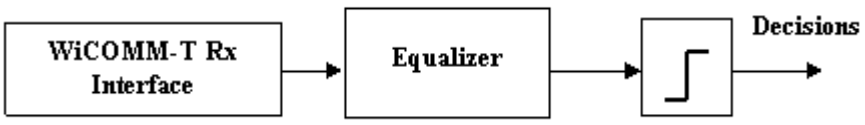


Fig: Transmitter block diagram in MATLAB

1. A RRC pulse of duration $-3T_b$ to $+3T_b$ (where T_b is the bit duration) is generated. The roll factor r can be changed between 0.11 and 0.99. The default value of r is 0.65.
2. Random data to be transmitted is generated.
3. Random data is QAM modulated.
4. The QAM symbols are upsampled by a factor of 8 samples per symbol. For a sampling rate of 2Msamples /sec, the symbol rate is $2 \times 10^6 / 8 = 250$Ksymbols/sec
5. The upsampled QAM symbols are convolved with the RRC pulse to obtain the pulse shaped bits.
6. The pre-distorted Channel with the FIR channel coefficients as given in Table 3.0 (Ref. WiCOMM-T user Mannual) is imposed on these RRC pulse shaped QAM symbols.
7. Pre-distorted symbols are given to the WiCOMM-T Tx interface block to send through WiCOMM-T.

Receiver



Receive block diagram in Matlab

The samples are received from the WiCOMM-T Rx interface blocks

1. The linear adaptive equalizer is applied to mitigate the channel effect.
2. The equalizer output is passed to the slicer for decisions.
3. The mean square error curve, constellation plot before and after equalizer and the frequency response plot of the equalizer are plotted.

Procedure

The following are the default values used for this experiment.

Transmitter

WiCOMM-T sampling rate = 2MBps

Roll-off factor (alpha) = 0.65

block_size = 20000 (Number of symbols)

upsample_facto = 8

Channel = Channel# 1 - Benign Channel

Receiver

WiCOMM-T sampling rate = 2MBps

block_size = 20000 (Number of symbols)

upsample_factor = 8

decimation_factor = 1

number_of_taps (L) = 20

Decision Delay (Δ) = 0

learning constant (μ) = 0.01

1. Connect WiCOMM-T in base-band loop back with the sampling rate set to 2MBps.
2. Generate the transmitter modem sample.
3. Transmit and receive the modem sample through WiCOMM-T and analyze the received modem samples.
4. Vary learning constant between 0.001 and 0.02, decision delay between 0 and 9 and observe the performance
5. Vary number_of_taps of equalizer 5 25 and observe the effects for two different channel models using the FIR channel coefficients mentioned in Table 3.0
6. Observe the various plots generated by MATLAB.

7. Connect WiCOMM-T in IF loop-back and repeat steps 2 to 6.
8. Connect 2 WiCOMM-Ts such that one is transmitter and the other is receiver and repeat steps 2 to 6

Note: For running this experiment between two WiCOMM-Ts such that one will be transmitter and other will be receiver, 'data1.bin', 'channel1.bin' generated by transmitter Matlab file under 'C:\WiCOMM-T\EXPERIMENTS\LE\REF_Data' directory should be copied to receiver 'C:\WiCOMM-T\EXPERIMENTS\LE\REF_Data' directory since receiver Matlab code refers 'data1.bin' file for equalization & 'channel1.bin' file for channel estimation.

EXPERIMENT 1.c

CODE DIVISION MULTIPLE ACCESS - MULTIPATH

Aim

1. To observe the BER performance of DS-CDMA using mixed codes in multipath channel using RAKE receiver for single user case.
2. Observe the BER performance of MRC combining and equal combining varying with SNR.

Concepts

1. What is Multipath?
2. Effect of Multipath on the performance of CDMA
3. What is RAKE Receiver?
4. Maximum Ratio Combining Technique
5. Equal Gain Combining Technique

Description

What is Multipath?

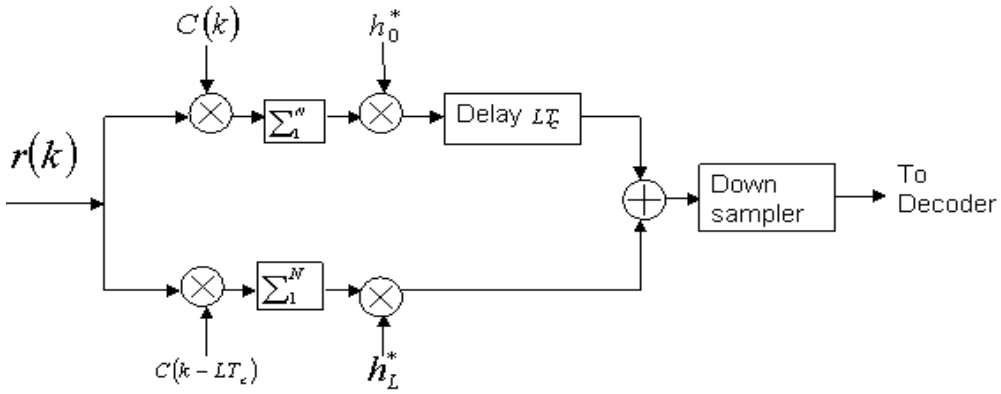
Multipath occurs when a radio signal is split into two or more signals causing the receiving antenna to receive multiple copies of the same signal. The radio signal can be split by obstacles such as walls, chairs, tables and other objects. As the signal bounces off an object it causes a longer path to the receiver. Some signals may bounce off several objects before reaching the receiver. The longer the path, the greater the amount of delay. As radio signals are delayed, they reach the receiving antenna at different times sometimes overlapping. The receiver becomes confused by the signals and is unable to interpret them correctly which causes data errors requiring retransmission of the signal. Performance can be significantly reduced by the delayed signals and retransmissions.

Effect of Multipath on the performance of DS-CDMA

CDMA is inherently tolerant to multipath delay spreading signals as any signal that is delayed by more than one chip time becomes uncorrelated to the PN code used to decode the signal. This results in the multipath simply appearing as noise. This noise leads to an increase in the amount of interference seen by each user subjected to the multipath and thus increases the received BER. The BER is essentially flat for delay spreadings of greater than one chip time (0.8 ms), which is to be expected as the reflected signal becomes uncorrelated. Also the multipath delay spreading leads to an increase in the equivalent number of users in the cell, as it increases the amount of interference seen by the receiver.

RAKE Receiver

A RAKE receiver is a radio receiver designed to nullify the effect of multipath fading. It uses number of sub-receivers called fingers. Each finger is a correlator and is designed to a different multipath component. Each finger independently decodes a single multipath component. The output of all the correlators is combined to increase the SNR in a multipath environment. The multipath channel through which a radio wave transmits can be viewed as transmitting the line of sight wave plus a number of multipath components. Multipath components are delayed copies of the original transmitted wave traveling through a different echo path, each with a different magnitude and time of arrival at the receiver. Since each component contains the original information, if the magnitude and phase of each component is computed at the receiver through a method called channel estimation then all the components can be added coherently to improve the information reliability. The RAKE receiver is so named because it looks like a garden rake, each finger collecting the symbol energy similar to how the fingers in a garden rake collects leaves. To minimize the distortions introduced in the DS-CDMA systems, RAKE receiver uses a technique called diversity.



RAKE Receiver

In our case, RAKE receiver has 2 fingers. Each finger of the receiver process one path of the composite multipath signal. All the processing in the RAKE fingers should be done at chip level. Here c(k)indicates the spreading code used for that particular user. h_0 and h_L are the multipath channel coefficients. LT_c is the delay that is used in the multipath channel model.

Maximum Ratio Combining (MRC)

Here the signal from different paths is weighted according to their individual signal voltage to noise power ratios and then summed. Hence maximum ratio combining produces an output SNR equal to the sum of the individual SNRs which is the acceptable SNR even when none of the individual signals are themselves acceptable. In MRC, the receiver corrects the phase rotation caused by a fading channel and then combines the received signals of different paths proportionally to the strength of each path. Since each path undergoes different attenuations, combining them with different weights yield an optimum solution under an AWGN channel.

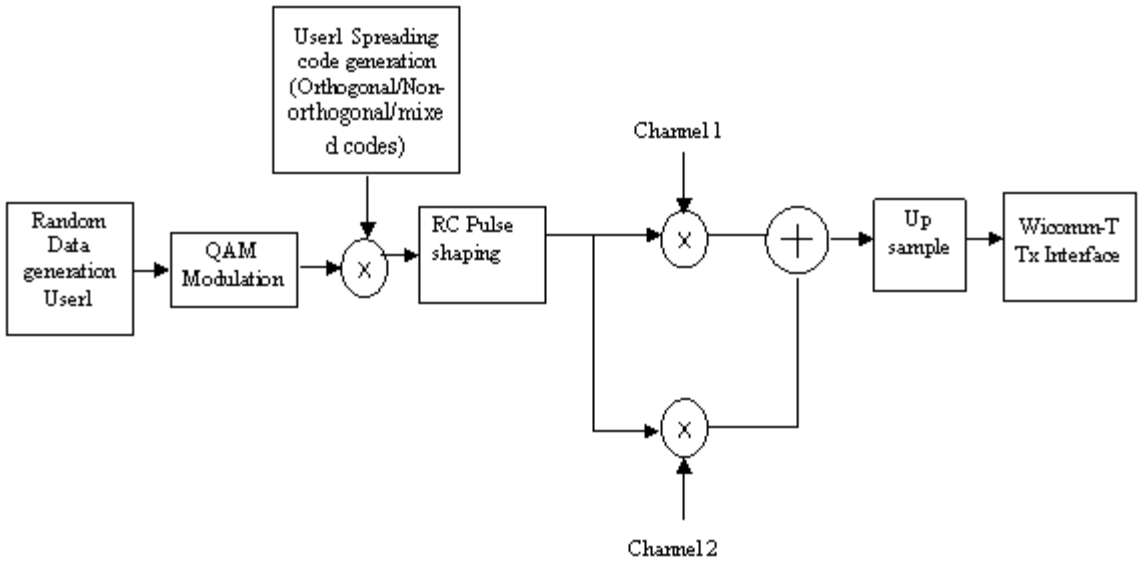
MRC is the optimum way (in the sense of the least BER) to use information from different paths to achieve decoding in an additive white Gaussian channel (AWGN)

Equal Gain Combining (EGC)

In certain cases, it is not convenient to provide for the variable weighting capability required for true maximal ratio combining. In such cases, the branch weights are all set to unity, but the signals from each branch are co-phased to provide equal gain combining diversity. This allows the receiver to exploit signals that are simultaneously received on each branch. The possibility of producing an acceptable signal from a number of unacceptable inputs is still retained, and performance is only marginally inferior to MRC The receiver corrects the phase rotation of the received signals caused by the fading channel and combines the received signals of different paths with equal weight. In a Rayleigh fading channel, the MRC performance is the best, followed by EGC. The performance of MRC is the same as that of EGC if signals from each path are of equal strength.

MATLAB Code Implementation

Transmitter



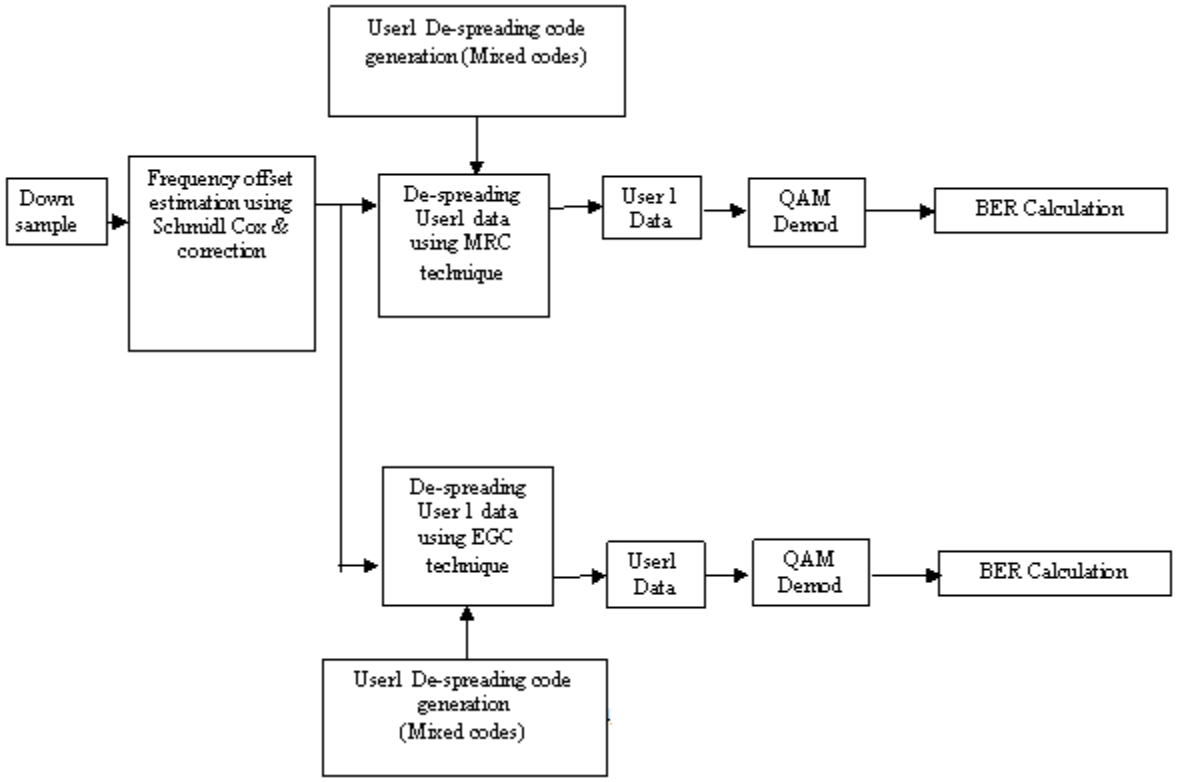
Block diagram of CDMA- Multipath Transmitter

1. Random data to be transmitted for User1 is generated.
2. Random data of User1 is QAM modulated.
3. The QAM modulated User1 data is convolved with its spreading code.
4. The convolved data of User1 is RC Pulse shaped.
5. The RC pulse shaped data is multiplied with different channels to show the multipath effect.
6. The data convolved with channel 1 and channel 2 are summed together.

7. The summed up data is upsampled.
8. The upsampled data is then given to the WiCOMM-T Tx interface block to send through the WiCOMM-T.

Receiver

1. The samples are received from the WiCOMM-T Rx interface block
2. The received samples are down sampled
3. The down sampled signals are de-spreaded using User1 de-spreading codes using MRC and EGC technique
4. The de-spreaded data are QAM demodulated for both MRC and EGC.
5. BER is calculated for the QAM demodulated data for both MRC and EGC.



Procedure

Note: Refer Appendix A on how to setup WiCOMM-T and Appendix B on how to generate the modem samples, vary the parameters, transmit, receive and analyzing the received modem samples etc. The following are the default values used for this experiment.

1. Connect WiCOMM-T for baseband loop back.
2. Select CDMAPART2 from the experiments list in EXPERIMNT window.

3. Select the SNR maximum and minimum value from pop up menu for generating the transmitter modem sample.
4. Transmit and receive the modem sample through WiCOMM-T.
5. Analyse the received modem samples.
6. Observe the BER plot generated by MATLAB for MRC and EGC techniques.
7. Connect WiCOMM-T in IF loop back and repeat steps 2 to 6
8. Connect 2 WiCOMM-Ts in baseband level and repeat steps 2 to 6
9. Connect the 2 WiCOMM-Ts in IF level and repeat steps 2 to 6

Note: For running this experiment between two WiCOMM-Ts such that one will be transmitter and other will be receiver, 'data1.bin', 'data2.bin'generated by transmitter Matlab file under 'C:\WiCOMM-T\EXPERIMENTS\CDMAPART2\REF_Data' directory should be copied to receiver 'C:\WiCOMM-T\EXPERIMENTS\CDMAPART2 \REF_Data' directory since receiver Matlab code refers 'data_1.bin'& 'data_2.bin' file for finding pilot symbols & BER calculations.

BER Calculation

The BER value for the two different types of rake receiver combining technique is obtained and tabulated. Find out the best combining technique from these results.

EXPERIMENT 1.d

CODE DIVISION MULTIPLE ACCESS - MULTIUSER

Aim

To understand the basic aspects of DS-CDMA in single user case and two user case.

Concepts

- Why Spread Spectrum Technique?
- Advantages of Spread Spectrum Technique
- Direct Sequence Spread Spectrum (DSSS)
- Spreading code
- Orthogonal Spreading Code
- Non-Orthogonal Spreading Code
- Mixed Spreading Code
- Near – Far effect
- Effect of delay in spreading the data
- Spreading and De-spreading of Data

Description

Why Spread Spectrum Technique?

Shannon's formula for channel capacity is a relationship between achievable bit rate, signal bandwidth and Signal to Noise Ratio (SNR).

$$\text{Channel Capacity} = \text{Bandwidth}*\log_2(1+\text{SNR})$$

When the signal is much smaller than the noise or under very low SNR condition the above relationship becomes much simpler as given below.

$$\frac{\textit{Channel Capacity}}{\textit{Bandwidth}} = 1.44*\text{SNR}$$

From the above relationship we can conclude that SNR can be traded for Bandwidth or vice versa. If there is a way to encode our data into a large signal bandwidth, then error free transmission is possible in a very low SNR condition. This is the reason why Spread Spectrum technique is used.

Advantages of Spread Spectrum Technique

Ability to selectively address

If the signal is spread and encoded properly, then the signal can only be decoded by a receiver which knows the transmitting code and hence a specific receiver in a group can be targeted. This is termed as Code Division Multiple Access

Bandwidth Sharing
If the proper modulation codes are selected, it is feasible to have multiple pairs of receivers and transmitters occupying the same bandwidth

Security

It is very difficult to intercept the signal if the modulation code of spread spectrum transmission is not known. If the proper spreading code is not known to demodulate, the signal will be seen as random electrical noise and not as useful signal. And also spread spectrum link puts out much less power per bandwidth than a conventional radio link, having spreading it over a wider bandwidth and hence a knowledge of the link's spreading code is required to demodulate. Hence it is very difficult to detect.

Immunity to Interference

If an external radio signal interferes with the spread spectrum signal, it will be rejected by the demodulator much as random noise and hence provide excellent error rate even with faint signals.

Direct Sequence Spread Spectrum

The Spread Spectrum technique can be divided into Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS). In DSSS the Pseudo Random sequence is applied directly to baseband data entering the carrier modulator. The modulator therefore sees a much larger bit rate, which corresponds to the chip rate of the PN sequence. This code sequence is typically Pseudo random binary code or PN specially chosen for desirable statistical properties. In effect, we are transmitting a wideband noise like signal which contains embedded message data. The time period of a single bit in the PN code is termed as chip and the bit rate of the PN code is termed Chip rate.

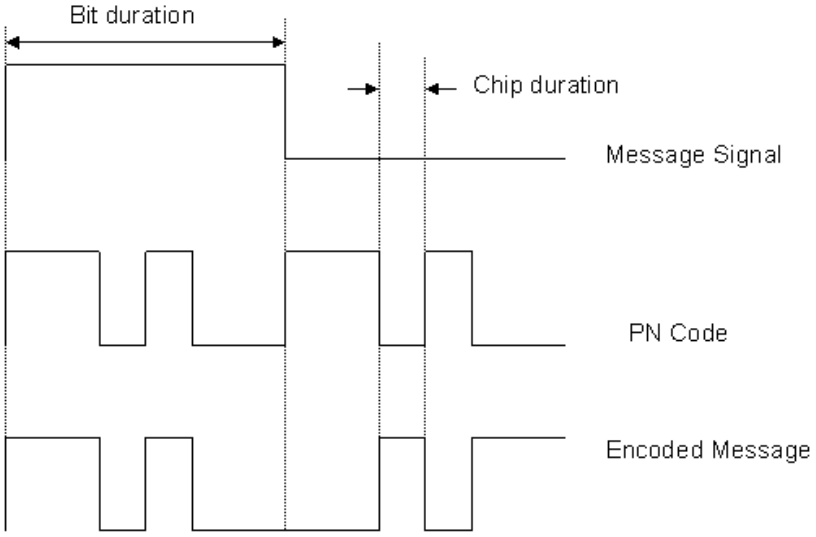


Figure 5.1.1: Message encoding using PN code

Spreading codes

The spreading code or the PN sequence should be ideally balanced with equal number of ones and zeros over the length of the sequence as well as cryptographically secure. Some of the most popular PN sequences are Barker, M – Sequence, Gold and Walsh. More complex sequences provide a more robust link but the implementation becomes very expensive. We have Orthogonal spreading codes, Non-Orthogonal spreading codes and Mixed spreading codes. Orthogonal codes are generated using Walsh-Hadamard series and the Non-orthogonal codes are generated using Linear Feedback Shift Register (LFSR). The mixed codes are generated by multiplying the orthogonal and non-orthogonal codes. The orthogonality property of the orthogonal codes is very important for any communication system. Because of the orthogonality property, two orthogonal signals can be transmitted at the same time and will not interfere with each other. But the auto correlation function of the Walsh – Hadamard matrix can have more than one peak and therefore it is not possible for the receiver to detect the beginning of code word without an external synchronization scheme. Also the cross correlation can also be non-zero for a number of time shifts and un-synchronous users can interfere with each other. The spreading is not over the entire bandwidth instead it is over a number of discrete frequency component. Orthogonality is affected by multi-path effect.
Gold sequences are popular for Non-orthogonal codes. Here the transmission can be asynchronous. The receiver can synchronize using the auto correlation property of the Gold Sequence.

Orthogonal Spreading Code

An important set of Orthogonal codes is the Walsh set. Walsh functions are generated using an iterative process of constructing a Hadamard matrix starting withH_1 =(1). The Hadamard matrix is built by where H_n is the inverse of H_n. For example Walsh – Hadamard matrix of length 2 and 4 are given as Type equation here.

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad H_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

From the above matrix the Walsh-Hadamard spreading codes are given by the rows of the matrix. For example one spreading code can be generated from the 2nd row of H_4which is given as$H_{4.2} = (1 \; -1 \; 1 \; -1)$ and the other spreading code can be generated from the 4th row of H_4 $H_{4.4} = (1 \; -1 \; -1 \; 1)$. Computing the orthogonality between $H_{4.2}$ and $H_{4.4}$ by multiplying element by element ((1x1)+(-1x-1)+(1x-1)+(-1x1)) = (1+ 1-1-1) = 0.This shows that both the spreading codes are orthogonal to each other. In our experiment we used H_6 and generated a code of length 32. Then the generated code is normalized by dividing it by$\sqrt{32}$.

Non-Orthogonal Spreading Code

To give up the orthogonality property among users and to reduce the interference between users by using spread spectrum technique, Non-orthogonal spreading codes are used. Non-orthogonal codes are generated using LFSR method. The LFSR is a shift register whose input bit is a linear function of the previous state. The initial value of the LFSR is called Seed. It has number of flip-flops termed as registers. For the feedback mechanism the bits contained in the selected positions in the shift registers are XORed and then fed as input to the first shift register. The bit positions selected for feedback is termed as taps. The Linear Feedback Shift Register is shown below. If the length of the LFSR is taken as n then the repetition rate of the PN sequence generated will be 2^n-1. The repetition rate is 2^n-1 because if the contents of all the registers are zero then the shift registers won't change their states. So the condition of all zeros in the PN sequence output is forbidden.

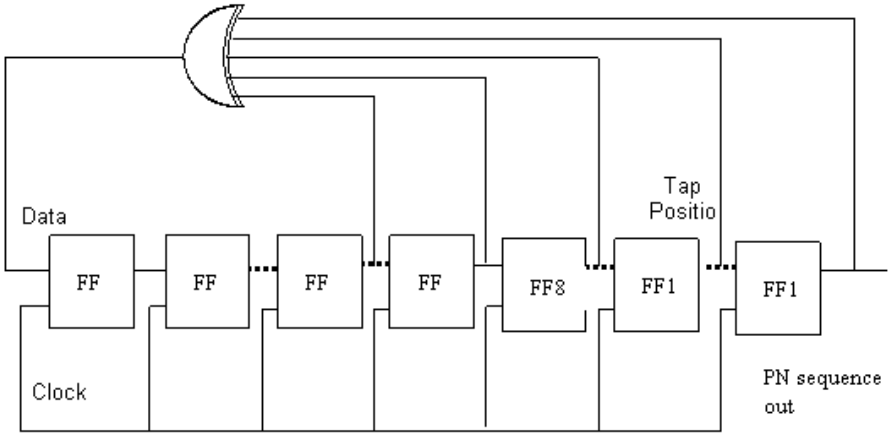


Figure 5.1.2: Linear Feedback Shift Register implementation for Non-Orthogonal codes

In our experiment the LFSR length is taken as 15 and hence the repetition rate of the PN sequence is 2^{15}-1 = 32767. The initial seed is taken as 110000000100001 and the tap positions are taken as 5,7,8,13,15. From the generated sequence any 32 length sequence can be picked up as spreading codes. The above LFSR is implemented in Matlab codes in our experiment.

Mixed Spreading Code

Mixed codes are generated to combine the advantages of both Orthogonal and non-orthogonal codes. This is generated by multiplying the orthogonal and non-orthogonal codes.

Near - Far Problem

The main problem with CDMA is the Near-Far effect. Consider a receiver and two transmitters; one close to the receiver; the other far away. If both transmitters transmit simultaneously and at equal powers, then the receiver will receive more power from the nearer transmitter than the farther transmitter. This makes the farther transmitter more difficult, if not impossible, to be understood. Since the signal from one transmitter is the noise for the other transmitter, the

Signal-to-noise ratio (SNR) for the nearer transmitter is much higher. If the nearer transmitter transmits a signal of higher power than the farther transmitter, then the SNR for the farther transmitter may be below the detectable level and the farther transmitter may look as if that it didn't transmit at all. This effectively jams the communication channel. In CDMA systems or other cellular phone-like networks, this is commonly solved by dynamic output power adjustment of the transmitters by the base stations. That is the closer transmitters use less power so that the SNR for all transmitters at the receiver is roughly the same

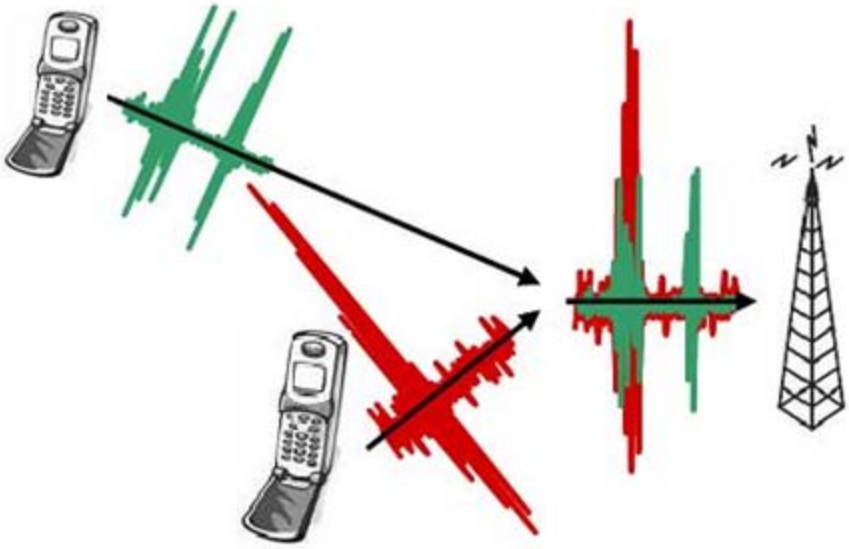


Fig 3: Near- Far problem in CDMA

This near-far problem is actually an uplink problem in reality. But in this experiment it is assumed as a downlink problem for the ease of implementation. Single Base station transmits the data at different powers to the two Users and thus the effect of one user data on other user is studied. The constellation plots for the two users are provided for ease of understanding of this phenomenon.

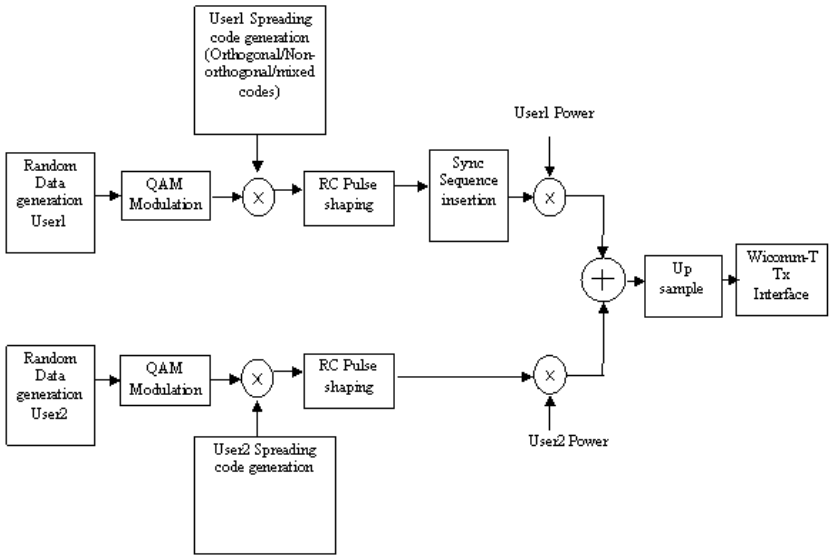
Effect of delay in spreading the data

If the signal from one user arrives little delayed than the other user, the orthogonality between user1 and user2 will be lost. When a CDMA receiver de-spreads a signal, it effectively computes the cross-correlation between the signal and a locally generated PN sequence. If this PN sequence is identical to the one used to spread the signal at the transmitter (ie. the message intended for the receiver) cross-correlation computations restore the original information. Otherwise, due to nonzero cross-correlation, some part of the other user data affects the desired user data depending on the power levels of the users.

This effect of delay is studied for three types of spreading codes discussed earlier. The delayed arrival of one user with respect to the other user is done by introducing delay between two users. It is observed that the mixed codes have a better constellation plot than the other two codes.

MATLAB Code Implementation

Transmitter



1. Random data to be transmitted for User1 and User2 are generated.
2. Random data of User1 and User2 are QAM modulated.
3. The QAM modulated User1 and User2 data are convolved with their corresponding spreading codes.
4. The convolved data of User1 and User2 are RC pulse shaped.
5. Sync sequence is generated and inserted in pulse shaped User1 data.
6. The Pulse shaped data of User1 and User2 are multiplied with their corresponding powers.
7. Two user data are added and then upsampled.
8. The upsampled data is then given to the WiCOMM-T Tx interface block to send through the WiCOMM-T.

Receiver

1. The samples are received from the WiCOMM-T Rx interface block
2. The received samples are down sampled
3. The start of the frame information is found out using Schmildl & Cox. Frequency offset in the down sampled signals are estimated using Schmidl Cox. and the estimated offset is corrected.
4. These frequency offset corrected samples are de-spreaded using User1 and User2 de-spreading codes
5. The phase offset in the de- spreaded data is estimated and corrected.
6. Then the residual frequency error is corrected by using LMS algorithm.
7. The residual frequency offset corrected data are QAM demodulated
8. BER is calculated for the QAM demodulated data

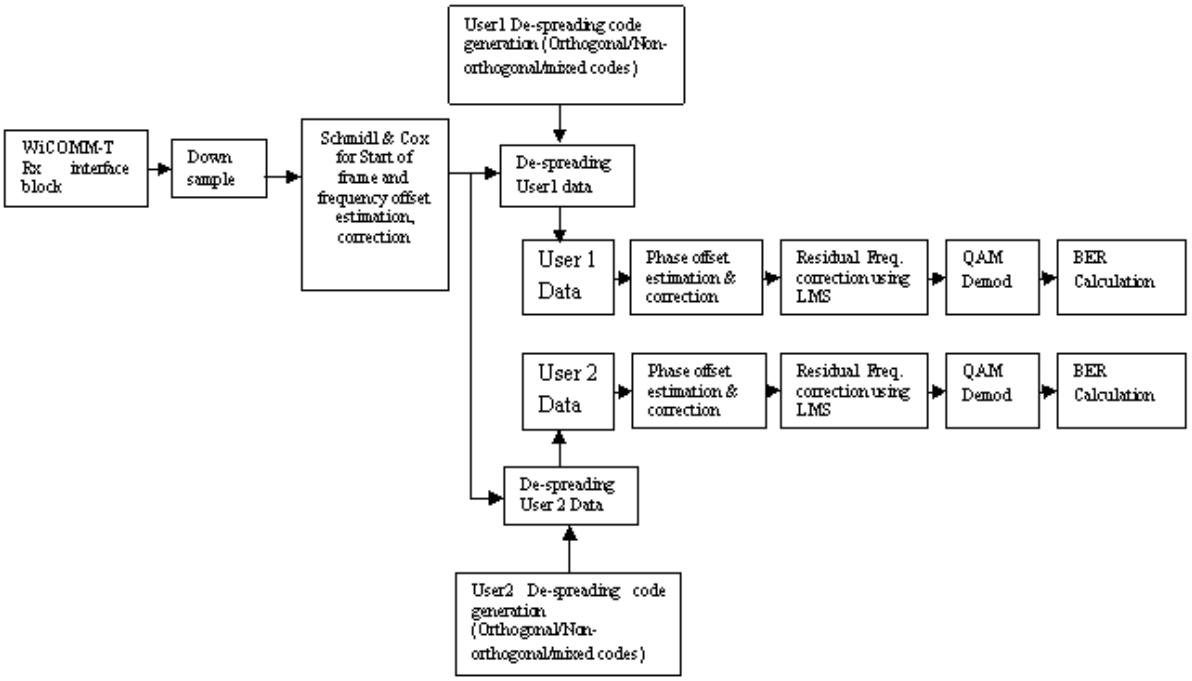


Fig : Block diagram of CDMA Receiver

Procedure

Note: Refer Appendix A on how to setup WiCOMM-T and Appendix B on how to generate the modem samples, vary the parameters, transmit, receive and analyze the received modem samples etc. The following are the default values used in this experiment.

Default values are given below:
Chip length = 32
Samples per symbol = 8
Data length = 12
No of frames = 10
Sync sequence length = 800
Spreading code : Orthogonal
Delay : zero
User Power level : User1 as one & User2 as zero.

1. Connect WiCOMM-T in baseband loop back.
2. Select CDMAPART1 from the Experiments popup menu in EXPERIMENT window.
3. Choose the orthogonal spreading code.
4. Keep the power of User1 (A1) as 1 and power of User2 (A2) as 0 for single user condition.
5. Generate the transmitter modem sample.
6. Transmit and receive the modem sample through WiCOMM-T.

7. Analyze the received modem samples.
8. Observe the various plots generated by MATLAB and tabulate the BER Value.
9. Introduce delay between two users.
10. Vary the power of User1 and User2 and repeat steps 5 to 8.
11. Change the spreading code to Non-orthogonal & Mixed code and repeat steps 3 to 10.
12. Connect WiCOMM-T in IF loop back.
13. Repeat steps 2 to 11 for IF loop back
14. Connect 2 WiCOMM-Ts in baseband level.
15. Repeat steps 2 to 11 for baseband communication
16. Connect the 2 WiCOMM-Ts in IF level
17. Repeat steps 2 to 11 for IF communication

Note: For running this experiment between two WiCOMM-Ts such that one will be transmitter and other will be receiver, 'data1.bin', 'data2.bin'generated by transmitter Matlab file under 'C:\WiCOMM-T\EXPERIMENTS\CDMAPART1\REF_Data' directory should be copied to receiver 'C:\WiCOMM-T\EXPERIMENTS\CDMAPART1 \REF_Data' directory since receiver Matlab code refers 'data1.bin'& 'data2.bin' file for finding pilot symbols & BER calculations.

Observation
Correlation

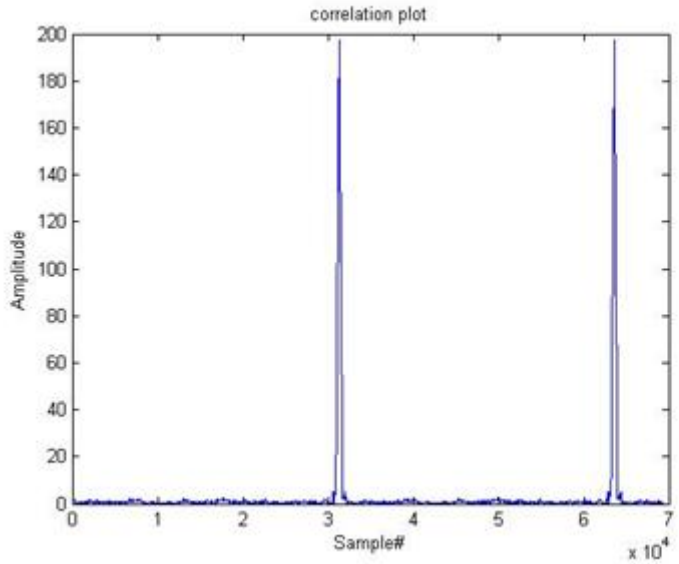


Fig :Correlation Plot to find the start of the frame

To find out the start of Frame information in the received signals we use Schmidl & Cox method. In addition to the start of the frame information Schmidl & Cox gives the coarse frequency offset of the received signal. This estimated coarse frequency offset is corrected and are then de-spreaded using User1 and User2 de-spreading codes. The peaks shown in above Figure shows the start of the Frame information.

Constellation diagram –
Orthogonal Spreading code:

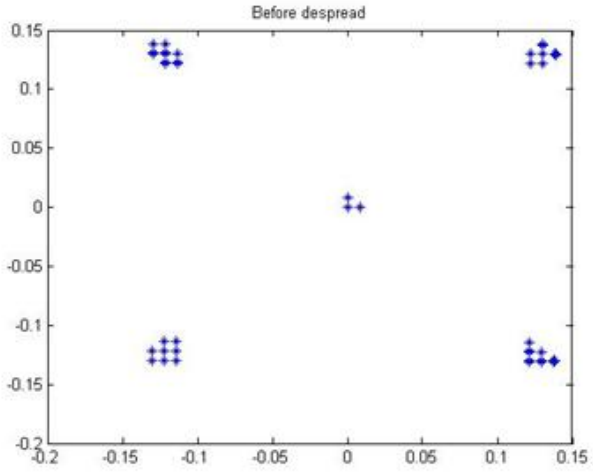


Fig 1:Constellation after frequency offset correction & before despreading – single User (BB loopback)

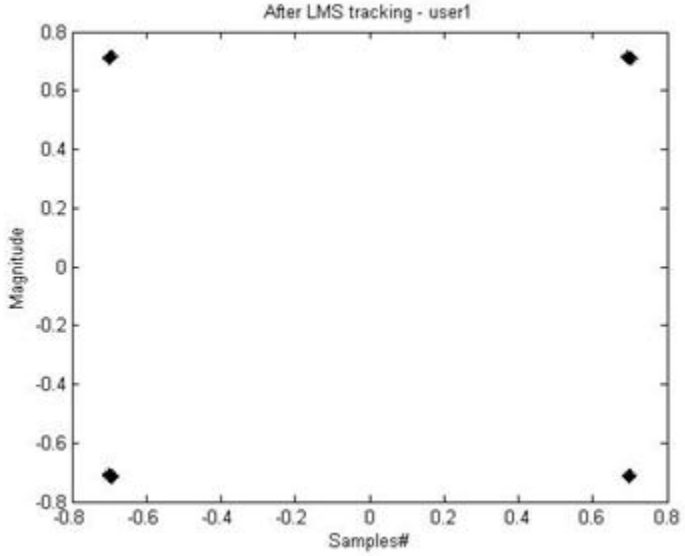


Fig 2: Constellation after despreading & LMS -– single User (BB loopback)

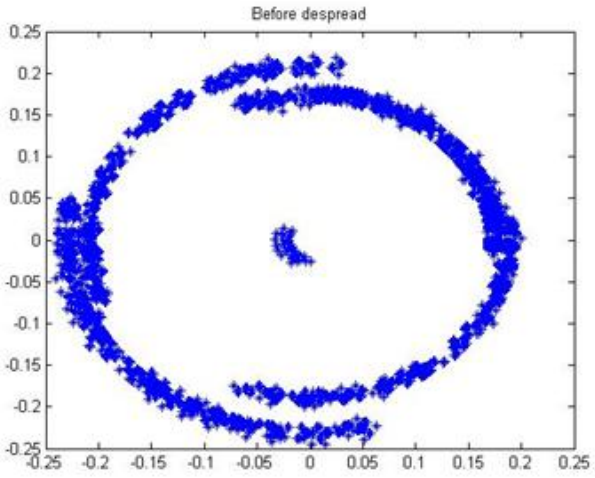


Fig 3 : Constellation after frequency offset correction & before despreading – single User (IF loopback)

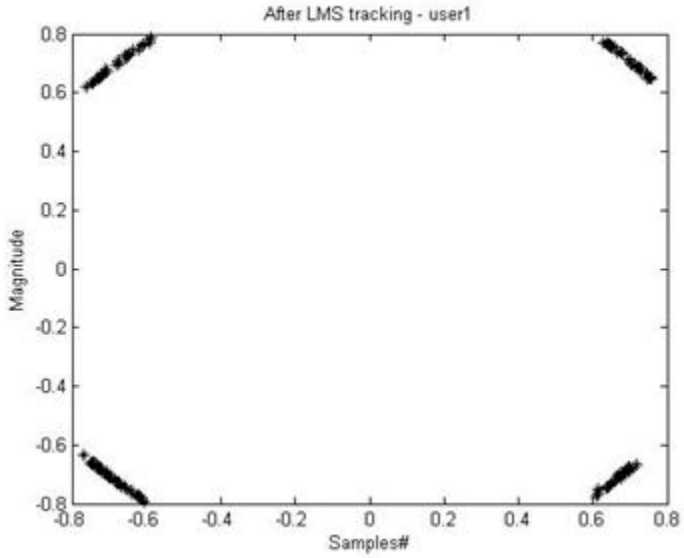


Fig 4 : Constellation after despreading & LMS – single User (IF loopback)

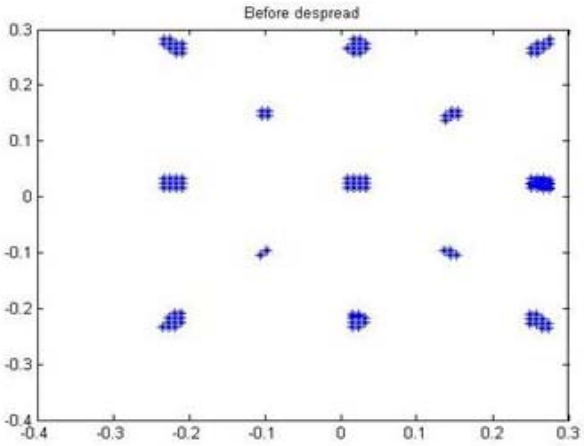


Fig 5 : Constellation after frequency offset correction & before despreading – two users at same power level with delay (BB loopback)

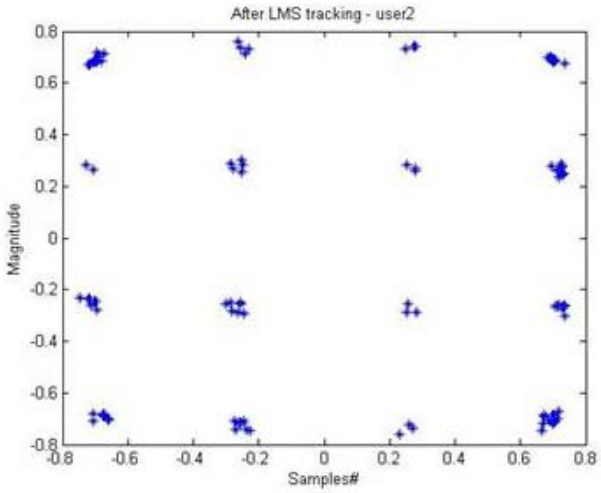


Fig 6 : Constellation after despreading & LMS – two users at same power level with delay (BB loopback)

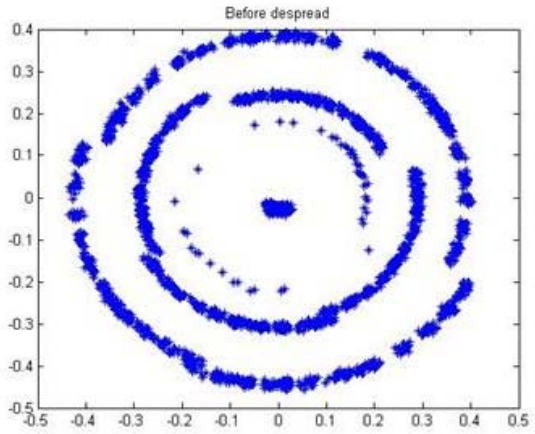


Fig 7 : Constellation after frequency offset correction & before despreading – two users at same power level with delay (IF loopback)

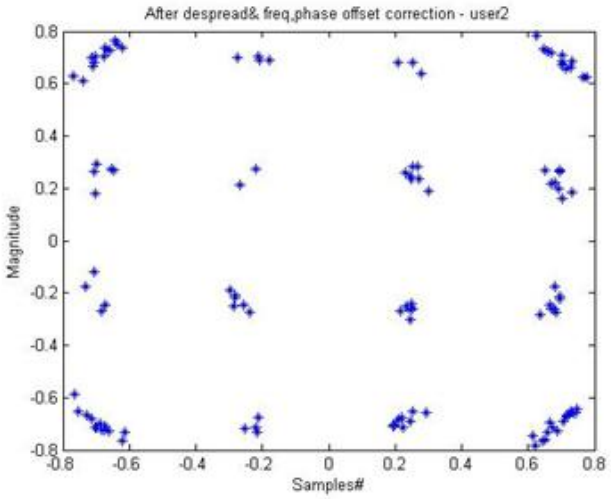


Fig 8 : Constellation after frequency despreading & LMS - two users at same power level with delay (IF loopback)

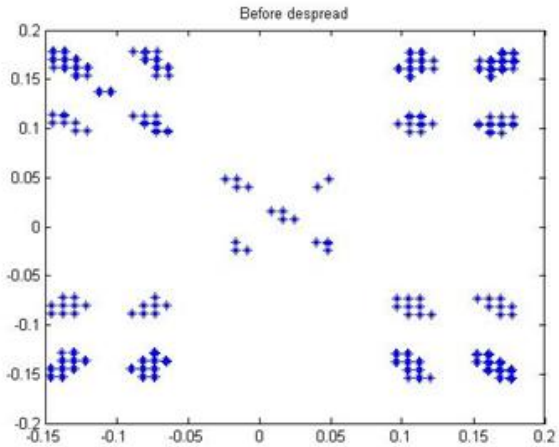


Fig 9 : Constellation after frequency despreading & LMS - two users at same power level with delay (IF loopback)

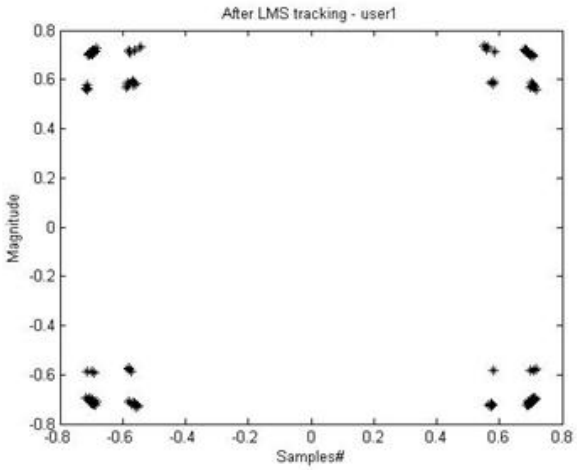


Fig 10: High power level user constellation after despreading & LMS - two users at different power level with delay (BB loopback)

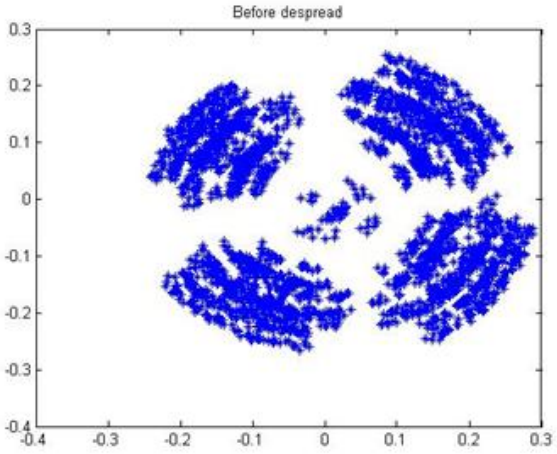


Fig 11: Constellation after frequencyoffset correction & before despreading - two users at same power level with delay (IF loopback)

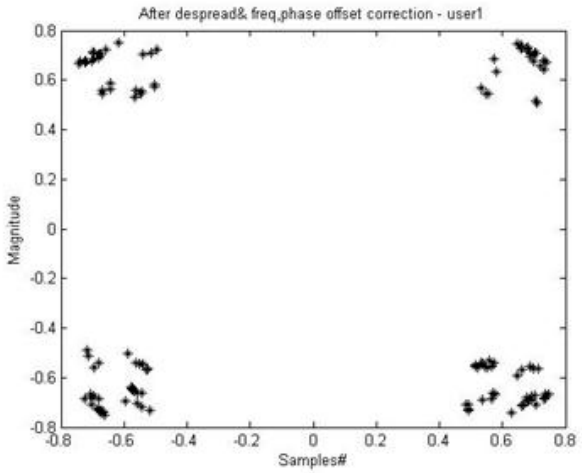


Fig 12: High power level user constellation after despreading & LMS - two users at different power level with delay (IF loopback)

1. Orthogonal Spreading code is generated using the same spreading sequence for the all the symbols for the particular user.

2. BER will be zero and the constellation will be four blobs in the single user condition since there is no interference from any other users as shown in the Fig 8.

3. When the two users are used with delay condition, the orthogonality is missed between the two users.

4. When both the users are having the same power level, with delay introduced between the users, the constellation will be having sixteen blobs. Each symbol of the one user interfered with the four possible symbols of the other user so that each blob in a quadrant becomes four blobs & hence getting 16 blobs as shown 2.and also these blobs are tighter because the cross correlation between the users for all the symbols is the same. Since the interference is not making the blobs to get shifted to other quadrants, we are getting zero BER for both the users.

5. When both the users are having different power level, (i.e one user power level is higher than the other user power level) with delay introduced between the users, the constellation will be having 16 blobs and the BER will be zero for the higher power level user as shown in Fig. 6. , where the lower power level user will be having higher BER and constellation will be poor. This is because of the interference of the higher power user with the lower power user. This condition is known as “Near Far Effect”. Figure 7 to 8 are taken with the Base band loopback and the IF loopback connection in WiCOMM-T for orthogonal spreading codes..

BER Calculation

Tabulate the values of the BER in the following table and analyze orthogonal spreading code to be used for different user conditions.

Users	Orthogonal Spreading code	
Single User		
Two users, same power with delay		
Two users, different power with delay		

EXPERIMENT 1.e

GLOBAL SYSTEM FOR MOBILE COMMUNICATION (GSM)

Aim

To study Gaussian Minimum Shift Keying (GMSK) modulation technique
To design a receiver using Viterbi algorithm
To study the BER using Viterbi

Concepts

- GMSK modulation
- Why GMSK modulation for GSM
- GMSK signal generation
- GSM transmitter
- GSM Receiver using Viterbi

Description

GMSK Modulation

Offset QPSK (OQPSK) is obtained from QPSK by delaying the Q data stream by 1 bit with respect to the I data stream. MSK is derived from OQPSK by replacing the rectangular pulses in amplitude with a half cycle sinusoidal pulse. MSK modulation makes the phase change linear and limited to $\pm\pi/2$ over a bit interval of T. Because of this linear phase change, the power spectral density has low side lobes that help to control adjacent channel interference. In MSK when the half sinusoidal pulse is replaced by Gaussian Pulse shape then the modulation is Gaussian Minimum Shift Keying (GMSK)

Why GMSK Modulation for GSM?

The phase of the transmitted signal in GMSK scheme is continuous and smoothed by a Gaussian filter. This results in more compact spectrum which enables better utilization of the available frequency spectrum. The side lobe energy for GMSK is less and hence channel spacing can be tighter. The compact spectrum is beneficial in a mobile communication scenario where the operators pay premium for bandwidth. Phase modulation, further, makes the transmitted signal to have constant envelope. The constant envelope property enables employing lower cost class C power amplifiers at the receiver end thereby reducing the overall cost.

GMSK Signal generation

To generate the GMSK signals the input data stream is first passed through a Gaussian Low pass filter with a Time-Bandwidth product (BT) of 0.3. This filter deliberately introduces ISI

spreading the bits over a period of 3 bits. The impulse response of the Gaussian low pass filter is given by:

$$g(t)=\frac{1}{2T}\left[Q\left(2\pi BT\frac{\left(t-\frac{T}{2}\right)}{T\sqrt{\ln 2}}\right)-Q\left(2\pi BT\frac{\left(t+\frac{T}{2}\right)}{T\sqrt{\ln 2}}\right)\right]$$

where Q function is defined by $s(n)=\cos(\phi(n))+j\sin(\phi(n))$

The phase of the modulated signal can then be got from the expression:

$$\phi(t)=\sum b_i h\pi\int_{-\infty}^{t-iT} g(u)du$$

where h is the modulation factor (=0.5 for GMSK) and b_i is the modulating NRZ data

The final GMSK signal is represented as

$$s(t)=\cos(\phi(t))+j\sin(\phi(t))$$

The phase response function, $q(n)$, is $q(t)$ sampled at F_s, extending from -1.5T to ∞

$$q(t)=h\pi\int^{t} g(u)$$

GSM Transmitter

Each GSM transmitter frame consists of 156.25 symbols. Six such frames constitute a hyper frame. Ten hyper frames repeated one after the other constitute the transmitted information. Total number of samples transmitted is $N_{samples}$ = 8 x 156.25 x 6 x 10 = 75000. The frame structure of the GSM transmitter consists of first 2 frames for the identification. They are the FCCH (Frequency Control Channel) and the SCH (Synchronization Channel). The remaining 4 frames carry the actual data to be transmitted.

- The FCCH consists of a 148 '0' bits followed by 8.25 random guard bits. It is mainly used to estimate the frequency difference between received and transmitted frequencies $S_4(n)$.

- The SCH channel has a known 64 bit sequence with good correlation properties. Hence this channel is used for frame synchronization. (In our case we use the whole SCH frame for synchronization)
- The traffic channel contains the data to be decoded.
- In this experiment, the parameters are estimated under noise free conditions.

GSM Receiver

GMSK signals can be detected in many ways. Optimal GMSK detection can be performed using MLSE, which is nonlinear and highly complex. Here for bit recovery Viterbi algorithm is used

Frequency Synchronization

Take samples of the received data, and calculate the FFT. The difference between the most dominant frequency component of the transmitted and received spectrum will give us the frequency offset between the transmitter and receiver. Necessary corrections are performed on the received data.

Frame Synchronization

Correlate the received data with the actual transmitted SCH channel and look for the peaks. The location of peak helps in identifying the beginning of the SCH channel. The beginning of the FCCH and the traffic channels are also identified.

Offset Phase Estimation

Carrier phase offset estimation is done with the help of FCCH channel. The received FCCH channel, previously identified through the frame synchronization, is decimated by a factor of 8 and the sequence $S_4(n)$ is chosen [i.e. the samples S(4), S(12), S(20) are selected]. Deterministic autocorrelation is performed over this set of data to estimate the carrier phase offset. The necessary phase corrections are made to the received data. The received data is now ready for demodulation of the traffic channels.

Traffic Channel Demodulation

The demodulation algorithm previously described is applied individually to each of the traffic channels to receive the transmitted data.

MATLAB Code implementation

Transmitter

1. Random data to be transmitted is generated.
2. FCCH and SCH channel are generated and then added to the random data.
3. The added data is then sent through Gaussian filter.

4. The Gaussian pulse shaped data is given to the WiCOMM-T Tx interface block to send through the WiCOMM-T.

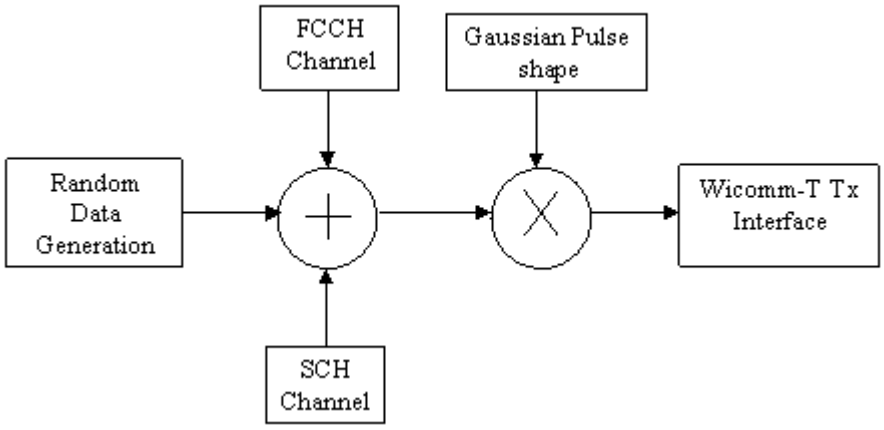


Fig. : Transmitter block diagram in MATLAB

Receiver

1. The samples are received from the WiCOMM-T Rx interface block
2. Frequency offset of the received samples are estimated and then corrected.
3. Identification of the SCH, FCCH channels and the burst data are done in the frequency offset corrected samples.
4. The phase offset is estimated and corrected.
5. The phase offset corrected samples are convolved with the Matched filter to recover the Burst data.
6. BER is calculated for various values of SNR and is plotted against the theoretical value.

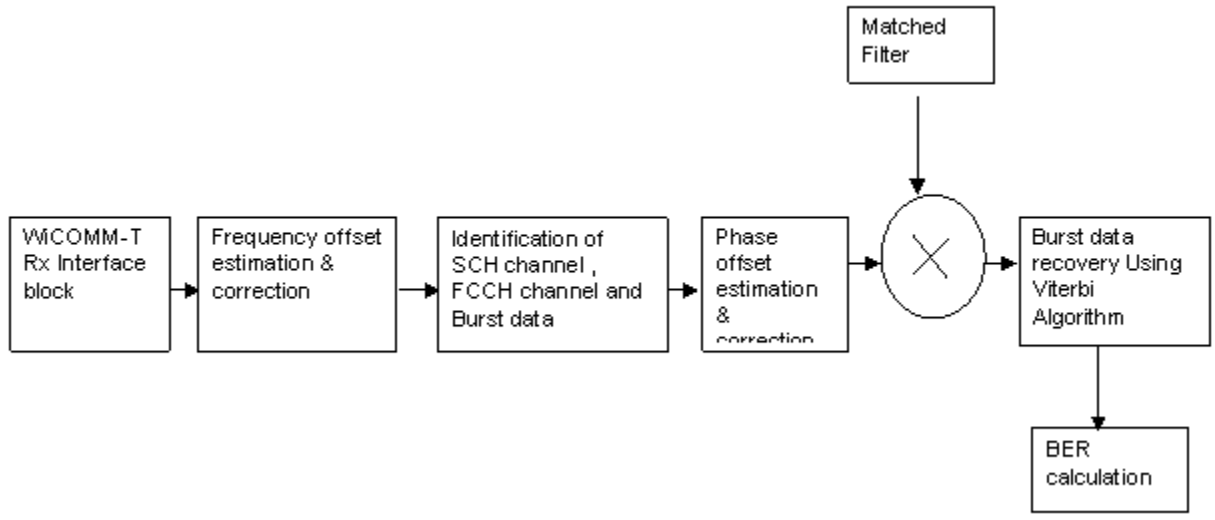


Fig. : Receiver block diagram in MATLAB

Procedure

Note: Refer Appendix A on how to setup WiCOMM-T and Appendix B on how to generate the modem samples, vary the parameters, transmit, receive and analyze the received modem samples etc.

1. Connect WiCOMM-T in baseband loop back with the sampling rate set to 2MBps.
2. Generate the transmitter modem sample.
3. Transmit and receive the modem sample through WiCOMM-T and analyse the received modem samples.
4. Observe various plots generated by MATLAB.
5. Connect WiCOMM-T in IF loop-back and repeat steps 2 to 4
6. Connect 2 WiCOMM-Ts such that one as transmitter and other as receiver in baseband and in IF and repeat steps 2 to 4

Note: For running this experiment between two WiCOMM-Ts such that one will be transmitter and other will be receiver, 'bits.bin', generated by transmitter Matlab file under 'C:\WiCOMM-T\EXPERIMENTS\GMSK\REF_Data' directory should be copied to receiver 'C:\WiCOMM-T\EXPERIMENTS\GMSK\REF_Data' directory since receiver Matlab code refers 'bits.bin' file for synchronization & BER calculation.

Result:
Thus the experiment was performed successfully.

EXPERIMENT 1.f

SPREAD SPECTRUM – DSSS MODULATION & DEMODULATION

INTRODUCTION:

Recall that when a sinusoidal carrier is DSBSC modulated by a message, the two signals are multiplied together. Recall also that the resulting DSBSC signal consists of two sets of sidebands but no carrier.

When the DSBSC signal is demodulated using product detection, both sidebands re multiplied with a local carrier that must be synchronized to the transmitter's carrier that is , it has the same frequency and phase. Doing so produces two messages that are in phase with each other and so add to form a single bigger message.

Direct sequence spread spectrum is a variation of DSBSC modulation scheme with a pulse train for the carrier instead of a simple sinewave. This may sound radical until you remember that pulse trans are actually made up of a theoretically infinite number of sinewaves. That being the cse, spread spectrum is really the DSBSC modulation of a theoretically infinite number of sinusoidal carrier signals. The result is a theoretically infinite number of pairs of tiny sidebands about a suppressed carrier.

In practice, not all of these sidebands have any energy of significance. However, the fact that the message information is distributed across so many of them makes spread spectrum signals difficult to deliberately interfere with "jam". To do so , you have to upset a significant number of the sidebands which is difficult considering their number .

Spread spectrum signals are demodulated in the same way as DSBSC signals using a product detector. Importantly, the product detectors local carrier signal must contain all the sinewaves that m ake up transmitter's pulse train at the same frequency and phase. If this is not done, the tiny demodulated signals will be at the wrong frequency and phase and so they wont add up to reproduce the original message. Instead, they'll produce a garbage signal that looks like noise.

The only way to obtain the right number of sinewaves at the right frequency and phase at the receiver is to use a pulse train with an identical sequence to that used by the transmitter. Moreover, it must be synchronized. This issue gives spread spectrum another of its advantages over other modulation schemes. The transmitted signal is effectively encrypted.

Of course, with trial and error its possible for an unauthorized person to guess the correct PN sequence to use for their receiver. However, this can be made difficult by making the sequence longer before it repeats itself. Longer sequences can produce more combinations of unique codes which would take longer to guess using a trial and error approach. To illustrate this point , an 8 bit code has 256 combinations while a 20 bit code has 1048575 combinations. A 256 bit code has $1.1579*10^{77}$ combinations.

Increasing the sequence's chip-length has another advantage. To explain, the total energy in a spread spectrum signal is distributed between all of the tiny DSBSC that make it up. A mathematical technique called Fourier Analysis shows that the greater the number of chips in a sequence before repeating, the greater the number of sinewaves of significance needed to make it.

That being the case, using more chips in the transmitter's PN sequence products more DSBSC signals and so the signal's total energy is distributed more thinly between them. This in turn means that the individual signals are many and extremely small. Infact , if the PN sequence is long enough, all of these DSBSC signals are smaller than the background electrical noise that's always present in free space. This fact gives spread spectrum yet another important advantage. The signal is difficult to detect.

Spread spectrum finds use in several digital applications including: CDMA mobile phone technology, cordless phones. The global positioning system and two of the 805.11 Wi-fi standards

OBJECTIVE:

In this experiment you'll use the Emona Telecoms-Telecoms-trainer 101 generate a DSSS signal by implementing its mathematical model. You'll then use a product detector (with a stolen carrier) to reproduce the message. Once done, you'll examine the importance of using the correct PN sequence for the local carrier and difficulty of jamming DSSS signals.

Equipment

1. Emona Telecoms Trainer 101 (plus power pack
2. Dual channel 20 MHz oscilloscope
3. Two Emona Telecoms Trainer 101 oscilloscope leads
4. Assorted Emona Telecoms Trainer 101 patch leads.

Procedure :

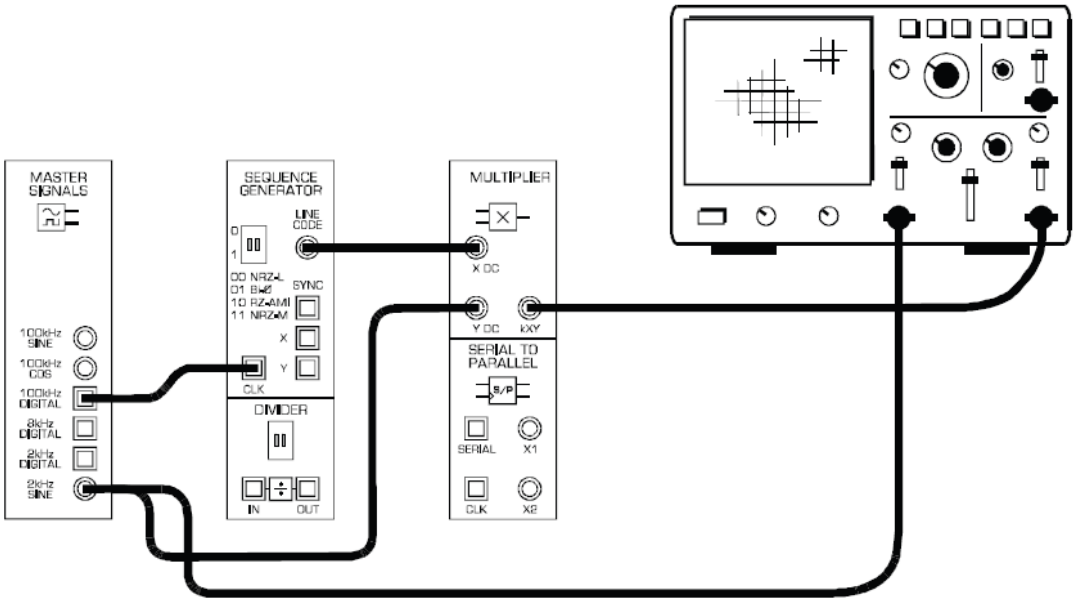
Part A

As DSSS is basically just DSBSC with a pulse train for the carrier instead of a simple sinusoid, it can be generated by implementing the mathematical model for DSBSC.

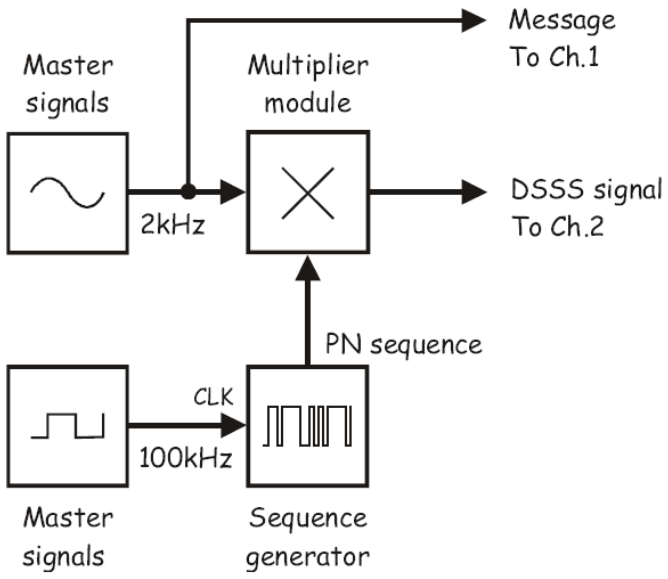
1. Gather a set of the equipment listed on the previous page.
2. Set up the scope per the instructions in experiment 1.
    a. The Trigger source control is set to the CHI position
    b. The Moe control is set to the CHI position.
3. Set the scope's Trigger source coupling control the t he HF REJ position.
4. Locate the sequence Generator module and set its dip-switches to 00

a. To do this, push both switches up.

5. Connect the set-up shown in Figure 1 below.
   a. Note : Insert the black plugs of the oscilloscope leads into a ground (GND) socket.



The set up i*N FIG 1* can be represented by the block diagram in Fig 2 below. It multiplies the 2 kHz sinewave message with a PN sequence modeled by the sequence Generator's 32 bit pulse train output.

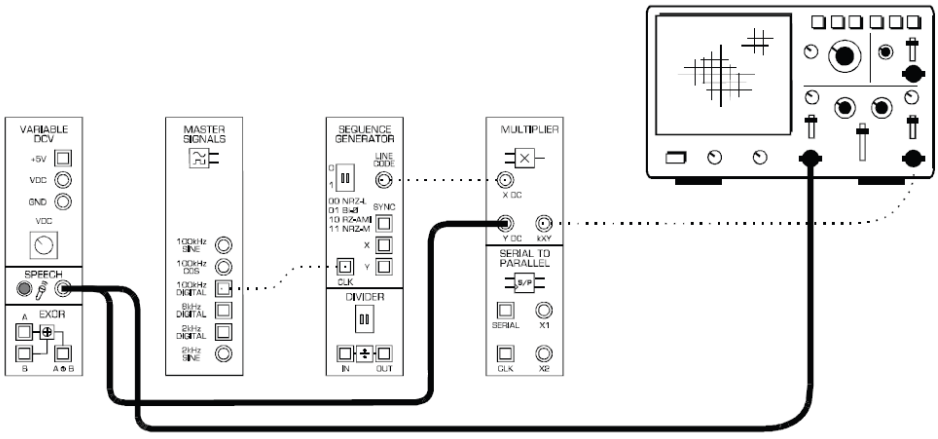


6. Adjust the scope's Time base control to view two or more cycles of the Master signals modules 2 kHz sine output.
7. Set the scope's Mode control to the DUAL position to view the DSSS signal out of the Multiplier Module as well as the message signal.
8. Adjust the scope's Vertical Attenuations controls to the appropriate settings for the signals.
9. Draw the two waveforms to scale in the space provided on the next page leaving room to draw a third waveform.
    a. Tip: Draw the message signal in the upper third of the graph and DSSS signal in the middle third.
10. Use the scope's channel 1 Vertical position control to overlay the message with the DSSS signal's envelope's and compare them.

Part B – Generating a DSSS signal using speech.

So far , this experiment has generated a DSSS signal using a sinewave for the message. The next part of the experiment lets you see what a DSSS signal looks like when modulated by speech.

11. Disconnect the plugs to the Master signals module's 2kHz sine output.
12. Connect them to the speech module's output as shown in fig 3 below.
    a. Remember Dotted lines sow leads already in place.

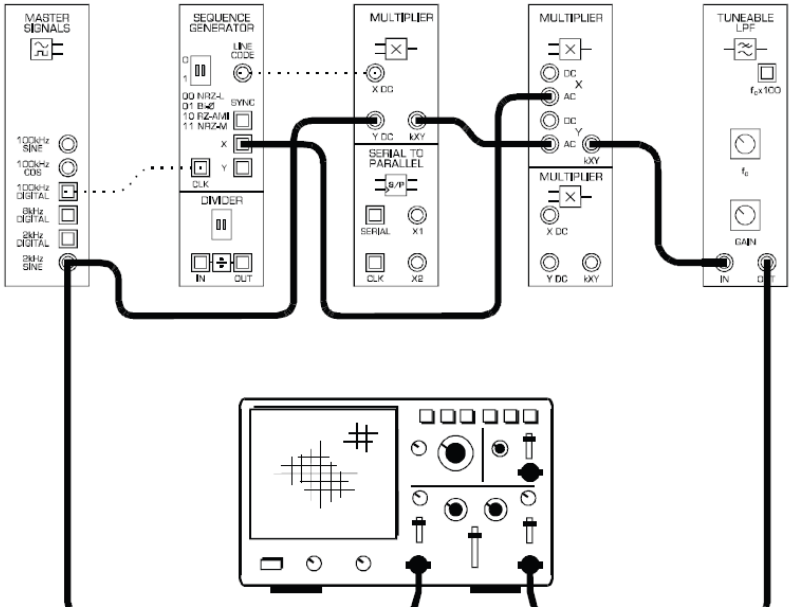


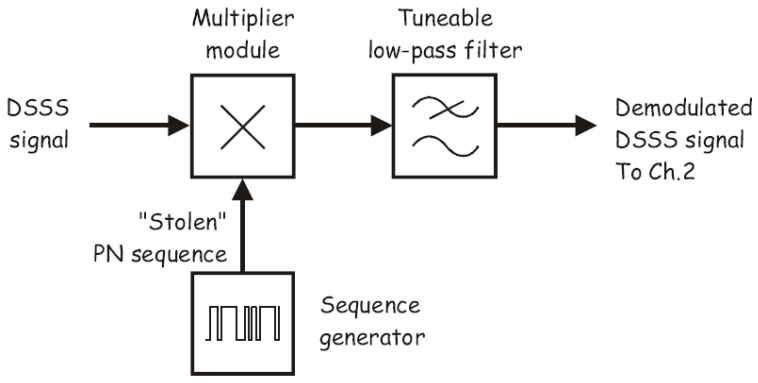
13. Set the scope's time base control to the 2ms/div position.
14. Talk, sing or hum while watching the scope's display.

Part C

Using the product detector to recover the message.

15. Return the scope's Time base control to its original position.
16. Locate the Tunable low pass filter module and set its Gain control to about the middle of its travel.
17. Turn the tunable low pass filter modules cut-off frequency adjust control fully anti-clockwise.
18. Disconnect the plugs to the speech module's output and modify the set-up as shown in Fig 4 below



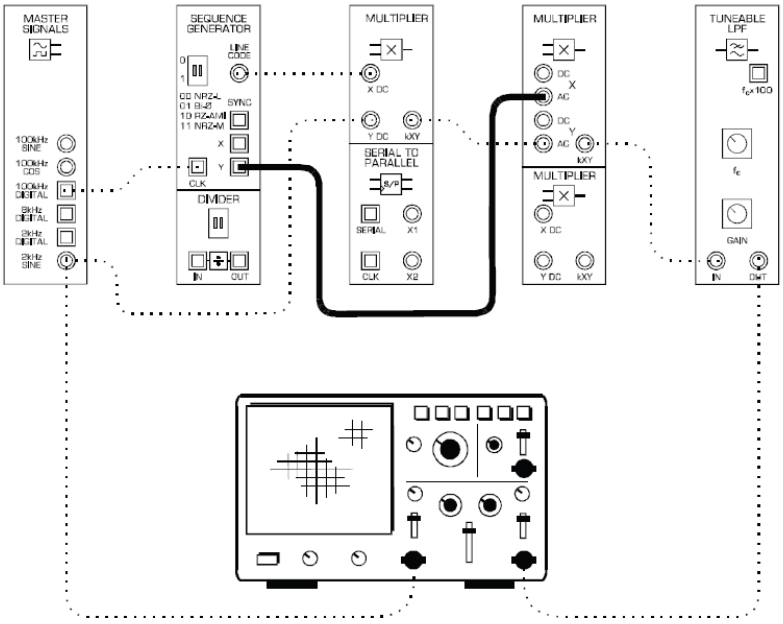


19. Slowly turn the Tunable low-pass Filter module's cut – off Frequency control clockwise while watching the scope's display and stop when it's at about half its travel.

20. Draw the demodulated DSSS signal to scale in the space that you left on the graph paper.

Recall that the message can only be recovered by the product detector if an identical PN sequence to the DSSS modulator's carrier is used. The next part of the experiment demonstrates. This.

21. Modify the setup as shown in Fig 7 below to make the demodulator's local carrier a different PN sequence to the transmitter's carrier.



22. Compare the message with the product detector's new output.

Part D

DSSS and deliberate interference (Jamming)

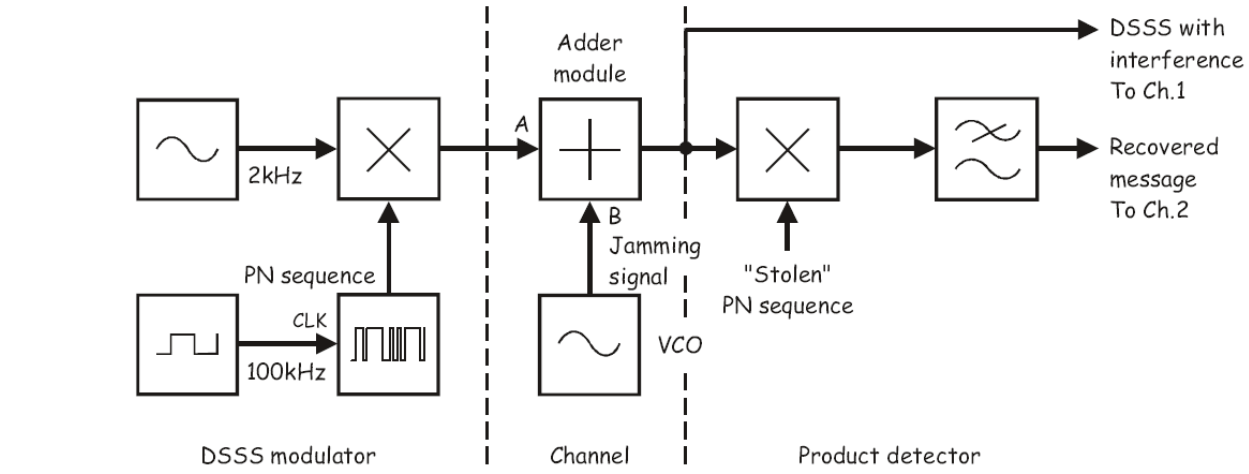
Interferences occurs when an unwanted electrical signal gets added to the transmitted signal and changes it enough to change the recovered message. Electrical noise is a significant source of unintentional interference.

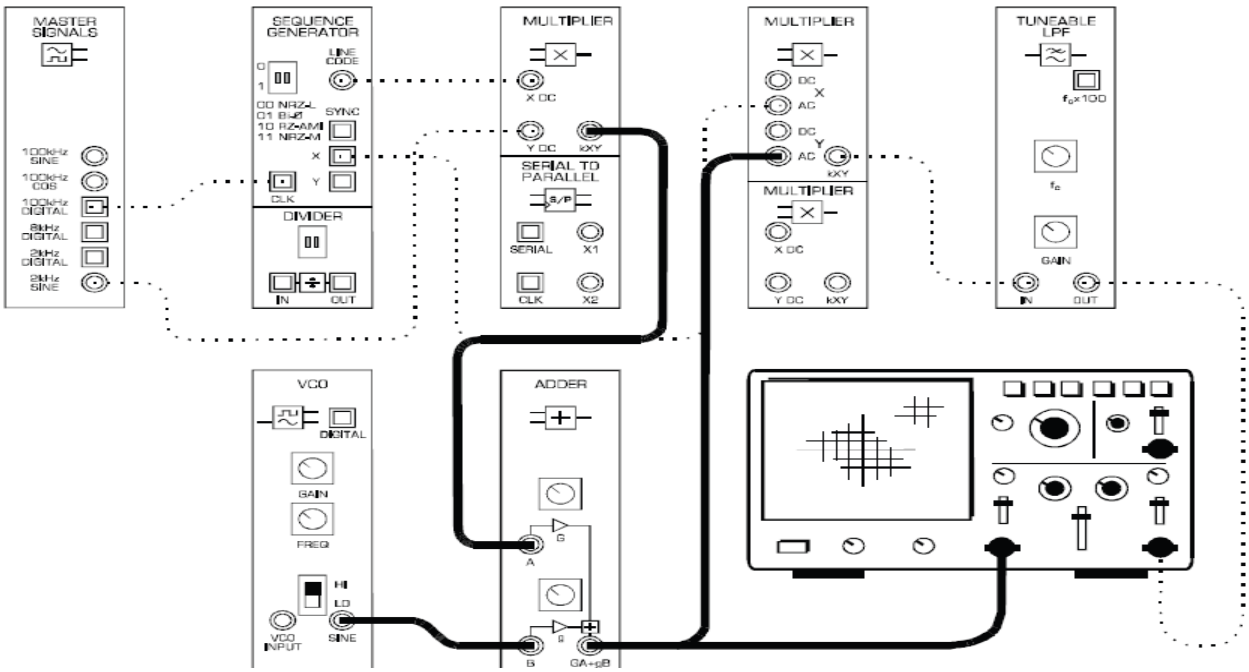
However, sometimes noise is deliberately added to the transmitted signal for the purpose of interfering or Jamming it. The next part of the experiment models deliberate interference to show how spread spectrum signals are highly resistant to it.

23. Move the patch lead from the sequence Generator's Y output back to its X output.
   a. Note : The product detector should now be recovering the message again.
24. Locate the VCO module and set its Range control the HI position.
25. Set the VCO modules Frequency Adjust control to about the middle of its travel.
26. Locate the Adder Module and turn its g control fully anti clockwise.

27. Set the Adder module's G control to about the middle of its travel.

28. Modify the setup as shown in Fig 8 below.



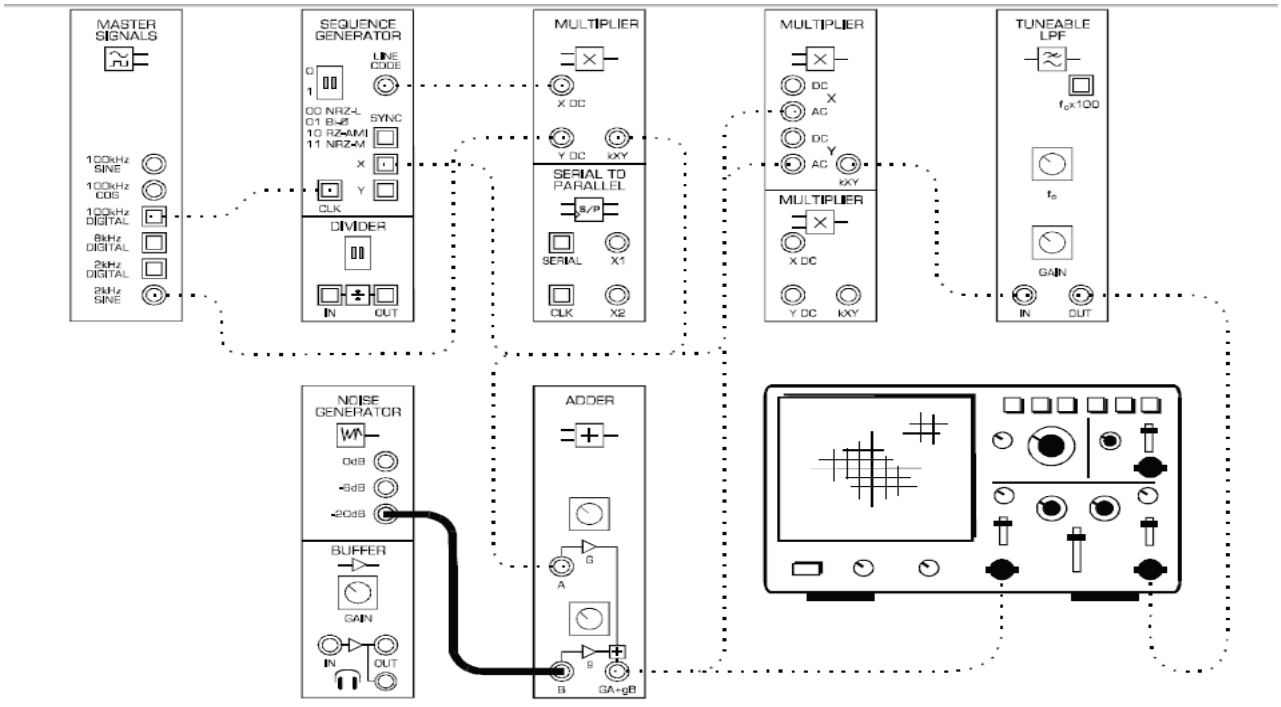


This modification forces the VCO module's output to sweep continuously through a wide range of Frequencies.

29. compare the two signals. Notice that the DSSS signal with interference is very distorted but the recovered message is only mildly affected.

An even more sophisticated approach to jamming involves using jamming signals at once to increase the chances of upsetting the transmitted signal. The next part of the experiment let's you see how spread spectrum handles this.

30. Modify the setup shown in Fig 11 below. This modification uses the Noise generator module to model a jamming signal that consists of thousands of frequencies.
31. Compare the two signal. Notice that the DSSS signal with interference is distorted but the recovered message is only mildly affected.
32. Increase the strength of the broadband jamming signal by connecting the Adder.
33. Compare the DSSS signal and the recovered message.
34. Increase the strength of the broadband jamming signal even more by connecting the Adder module's B input to the Noise Generator module's 0dB output.
35. Compare the two signals. Notice how distorted DSSS signal is but how little the recovered message is affected.
36. Modify the Setup as shown.



Result:

Thus the Experiment was performed Successfully.

WIRELESS PATH LOSS COMPUTATIONS

STUDY OF PROPAGATION PATH LOSS MODELS: INDOOR & OUTDOOR

EXPERIMENT 2.a
FREESPACE PROPAGATION – PATH LOSS MODEL

Aim:

To determine the freespace loss and the power received using Matlab program.

Theory:

The free space path loss, also known as FSPL is the loss in signal strength that occurs when an electromagnetic wave travels over a line of sight path in free space. In these circumstances there are no obstacles that might cause the signal to be reflected refracted, or that might cause additional attenuation.

The free space path loss calculations only look at the loss of the path itself and do not contain any factors relating to the transmitter power, antenna gains or the receiver sensitivity levels.

To understand the reasons for the free space path loss, it is possible to imagine a signal spreading out from a transmitter. It will move away from the source spreading out in the form of a sphere. As it does so, the surface area of the sphere increases. As this will follow the law of the conservation of energy, as the surface area of the sphere increases, so the intensity of the signal must decrease.

As a result of this it is found that the signal decreases in a way that is inversely proportional to the square of the distance from the source of the radio signal

Free space path loss formula

The free space path loss formula or free space path loss equation is quite simple to use. Not only is the path loss proportional to the square of the distance between the transmitter and receiver, but the signal level is also proportional to the square of the frequency in use.

$$FSPL = (4\pi d/ \lambda)2 \quad = (4\pi df/ c)2$$

FSPL is the Free space path loss
d is the distance of the receiver from the transmitter (metres)
λ is the signal wavelength (metres)
f is the signal frequency (Hertz)
c is the speed of light in a vacuum (metres per second)

The free space path loss formula is applicable to situations where only the electromagnetic wave is present, i.e. for far field situations. It does not hold true for near field situations.

Decibel version of free space path loss equation

Most RF comparisons and measurements are performed in decibels. This gives an easy and consistent method to compare the signal levels present at various points. Accordingly it is very convenient to express the free space path loss formula, FSPL, in terms of decibels..

FSPL (dB) = 20 $\log_{10}$ (d) + 20 $\log_{10}$ (f) + 32.44

Where:
d is the distance of the receiver from the transmitter (km)
f is the signal frequency (MHz)

Affect of antenna gain on path loss equation

The equation above does not include any component for antenna gains. It is assumed that the antenna gain is unity for both the transmitter. In reality, though, all antennas will have a certain amount of gain and this will affect the overall affect. Any antenna gain will reduce the "loss" when compared to a unity gain system. The figures for antenna gain are relative to an isotropic source, i.e. an antenna that radiates equally in all directions.

FSPL (dB) = 20 $\log_{10}$ (d) + 20 $\log_{10}$ (f) + 32.44 -Gtx - Grx

Where:
Gtx is the gain of the transmitter antenna relative to an isotropic source (dBi)
Grx is the gain of the receiver antenna relative to an isotropic source (dBi)

The free space path loss equation or formula given above, is an essential tool that is required when making calculations for radio and wireless systems either manually or within applications such as wireless survey tools, etc. By using the free space path loss equation, it is possible to determine the signal strengths that may be expected in many scenarios. While the free space path loss formula is not fully applicable where there are other interactions, e.g. reflection, refraction, etc as are present in most real life applications, the equation can nevertheless be used to give an indication of what may be expected. It is obviously fully applicable to satellite systems where the paths conform closely to the totally free space scenarios

Power Received :

[Pr] = [pt] + [Gt] + [Gr] – [FSPL]

Pr – Received power
Pt – Transmitted power
Gt – Gain of the transmitting antenna
Gr – Gain of the receiving antenna

Program:

```
clc;
close all;
clear all;
f=input('enter the frequency in Mhz: ');
L=300/f;                                          %calculating wavelength
disp('thus the wavelength is: ');
L                                                 %displaying wavelength
d=input('enter the distance in km: ');
Gt=input('enter the transmitting antenna gain in db: ');
Gr=input('enter the receiving antenna gain in db: ');
Wt=input('enter the transmitted power in db: ');
ls=32.44+20*log10(d)+20*log10(f);            %calculating path loss
disp(sprintf('%s %d %s','the path loss is:',ls,'db'));%displaying path loss
Wr=Wt+Gt+Gr-ls;                                   %calculating recieved power in db
disp(sprintf('%s %d %s','the recieved power is:',Wr,'db'));
wr=10^(Wr/10);                                    %calculating recieved power in watts
disp(sprintf('%s %d %s','the recieved power is:',wr,'watts'));
                                                  %displaying recieved power in watts
```

Result:

The program for power received by an antenna and path loss in Free space propagation was simulated successfully.

EXPERIMENT 2.b
LINK BUDGET EQUATION – SATELLITE COMMUNICATION

Aim :

To write a Matlab program to calculate the link budget for satellite communication.

Theory :

A link budget is an accounting of all the gains and losses in a transmission system. The link budget looks at the elements that will determine the signal strength arriving at the receiver. The link budget may include the following items:

- Transmitter power.
- Antenna gains (receiver and transmitter).
- Antenna feeder losses (receiver and transmitter).
- Path losses.
- Receiver sensitivity (although this is not part of the actual link budget, it is necessary to know this to enable any pass fail criteria to be applied.

Where the losses may vary with time, e.g. fading, and allowance must be made within the link budget for this - often the worst case may be taken, or alternatively an acceptance of periods of increased bit error rate (for digital signals) or degraded signal to noise ratio for analogue systems.

Received power (dBm) = Transmitted power (dBm) + gains (db) - losses (dB)

The basic calculation to determine the link budget is quite straightforward. It is mainly a matter of accounting for all the different losses and gains between the transmitter and the receiver.

Losses = FSL + AML + RFL + PL + AA

FSL = Freespace loss

AML = Antenna Misalignment loss

RFL=Receiver Feeder loss

PL=Polarization Loss

AA = Atmospheric Absorption.

Carrier to Noise Ratio – Uplink

$$CNR_u=EIRP_u+GTR_u-Loss_u+228.6$$

Carrier to Noise Ratio – Uplink

$$CNR_d=EIRP_d+GTR-Loss_d+228.6$$

Overall Carrier to Noise Ratio

$CNR_{overall}=CNR_u \text{ X } CNR_d \text{ / } (CNR_u+CNR_d)$

Program:

```
clc;
close all;
clear all;
pt=input('enter the input power in watts:');
Pt=10*log10(pt) %calculating transmitted power in db
gt=input('enter the transmitting antenna gain in db:');
gs=input('enter the recieving antenna gain in db:');
EIRP=Pt+gt      %calculating EIRP
d=input('enter the distance in km:');
f=input('enter the frequency in mhz:');
fsl=32.4+20*log10(d)+20*log10(f) %calculating path loss
rfl=input('enter the reciever feeder loss in db:');
aa=input('enter the atmospheric absorption in db:');
aml=input('enter the antenna misalignment loss in db:');
pl=input('enter the polarization loss in db:');
losses=fsl+rfl+aa+aml+pl;  %calculating total losses
disp(sprintf('%s %f %s','total loss',losses,'db'));
P=EIRP+gs-losses;          %calculating power recieved
disp(sprintf('%s %f %s','Total recieved power =',P,'db'));
```

Result:

The Matlab program for calculating the link budget was simulated successfully.

EXPERIMENT 2.c
CARRIER TO NOISE RATIO – SATELLITE COMMUNICATION

Aim :

To write a Matlab program to calculate the link budget for satellite communication and also to calculate the Carrier to noise ratio for uplink and downlink and also the overall carrier to noise ratio.

Theory :

A link budget is an accounting of all the gains and losses in a transmission system. The link budget looks at the elements that will determine the signal strength arriving at the receiver. The link budget may include the following items:

- Transmitter power.
- Antenna gains (receiver and transmitter).
- Antenna feeder losses (receiver and transmitter).
- Path losses.
- Receiver sensitivity (although this is not part of the actual link budget, it is necessary to know this to enable any pass fail criteria to be applied.

Where the losses may vary with time, e.g. fading, and allowance must be made within the link budget for this - often the worst case may be taken, or alternatively an acceptance of periods of increased bit error rate (for digital signals) or degraded signal to noise ratio for analogue systems.

Received power (dBm) = Transmitted power (dBm) + gains (db) - losses (dB)

The basic calculation to determine the link budget is quite straightforward. It is mainly a matter of accounting for all the different losses and gains between the transmitter and the receiver.

Losses = FSL + AML + RFL + PL + AA

FSL = Freespace loss

AML = Antenna Misalignment loss

RFL=Receiver Feeder loss

PL=Polarization Loss

AA = Atmospheric Absorption.

Carrier to Noise Ratio – Uplink

$$CNR_u=EIRP_u+GTR_u-Loss_u+228.6$$

Carrier to Noise Ratio – Uplink

$$CNR_d=EIRP_d+GTR-Loss_d+228.6$$

Overall Carrier to Noise Ratio

$CNR_{overall}=CNR_u \text{ X } CNR_d \text{ / } (CNR_u+CNR_d)$

Program:

```
clc;
clear all;
close all;
EIRPu=input('Enter the uplink EIRP:');
EIRPd=input('Enter the downlink EIRP:');
GTRu=input('Enter the uplink G/T:');
GTRd=input('Enter the downlink G/T:');
FSLu=input('Enter the uplink FSL:');
FSLd=input('Enter the downlink FSL:');
RFLu=input('Enter the uplink RFL:');
RFLd=input('Enter the downlink RFL:');
AAu=input('Enter the uplink AA:');
AAd=input('Enter the downlink AA:');
AMLu=input('Enter the uplink AML:');
AMLd=input('Enter the downlink AML:');
Lossu=FSLu+RFLu+AAu+AMLu    %calculating total losses in UPLINK
Lossd=FSLd+RFLd+AAd+AMLd    %calculating total losses in DOWNLINK
CNRu=EIRPu+GTRu-Lossu+228.6; %calculating CNR of UPLINK
disp(sprintf('%s %f %s','total carrier to noise ratio for uplink is:',CNRu,'decilog'));
CNRd=EIRPd+GTRd-Lossd+228.6; %calculating CNR of DOWNLINK
disp(sprintf('%s %f %s','total carrier to noise ratio for downlink is:',CNRd,'decilog'));
CNRt=CNRu*CNRd/(CNRu+CNRd);  %calculating total CNR
disp(sprintf('%s %f %s','total carrier to noise ratio is:',CNRt,'decilog'));
```

Result:

The program for power received by an antenna and path loss in Free space propagation was simulated successfully.

EXPERIMENT 2.d
OUTDOOR PROPAGATION MODEL - OKUMURA MODEL

Aim:

To write a Matlab program to calculate the median path loss for Okumura model for outdoor propagation.

Theory:

The Okumura model for Urban Areas is a Radio propagation model that was built using the data collected in the city of Tokyo, Japan. The model is ideal for using in cities with many urban structures but not many tall blocking structures. The model served as a base for the Hata Model.

Okumura model was built into three modes. The ones for urban, suburban and open areas. The model for urban areas was built first and used as the base for others.

Coverage

Frequency = 150 MHz to 1920 MHz
Mobile Station Antenna Height: between 1 m and 10 m
Base station Antenna Height: between 30 m and 1000 m
Link distance: between 1 km and 100 km

Mathematical formulation

The Okumura model is formally expressed as:

$$L = L_{FSL} + A_{MU} - H_{MG} - H_{BG} - \sum K_{CORRECTION}$$

where,

L = The median path loss. Unit: Decibel (dB)
L_{FSL} = The Free Space Loss. Unit: Decibel(dB)
A_{MU} = Median attenuation.Unit: Decibel(dB)
H_{MG} = Mobile station antenna height gain factor.
H_{BG} = Base station antenna height gain factor.
$K_{correction}$ = Correction factor gain (such as type of environment, water surfaces, isolated obstacle etc.)

Okumura model does not provide a mean to measure the Free space loss. However, any standard method for calculating the free space loss can be used.

Program:

```
clc;
clear all;
close all;
Lfsl=input('enter the free space loss:');
Amu=input('enter the median attenuation value:');
Hmg=input('enter the Mobile station antenna height gain factor:');
Hbg=input('enter the Base station antenna height gain factor:');
Kc=input('enter the Correction factor gain:');
L=Lfsl+Amu-Hmg-Hbg-Kc;  %calculating median path loss
disp(sprintf('%s %f %s','the median path loss:',L,'dB'));
```

Result:

The program for Okumura Model – Outdoor Propagation was simulated successfully.

EXPERIMENT 2.e

OUTDOOR PROPAGATION MODEL - HATA MODEL

Aim:

To write a Matlab program to calculate the median path loss for Hata model for outdoor propagation.

Theory:

In wireless communication, the Hata Model for Urban Areas, also known as the *Okumura-Hata model* for being a developed version of the Okumura Model, is the most widely used radio frequency propagation model for predicting the behaviour of cellular transmissions in built up areas. This model incorporates the graphical information from Okumura model and develops it further to realize the effects of diffraction, reflection and scattering caused by city structures. This model also has two more varieties for transmission in Suburban Areas and Open Areas.

Hata Model predicts the total path loss along a link of terrestrial microwave or other type of cellular communications.

This particular version of the Hata model is applicable to the radio propagation within urban areas.

This model is suited for both point-to-point and broadcast transmissions and it is based on extensive empirical measurements taken.

PCS is another extension of the Hata model. The Walfisch and Bertoni Model is further advanced.

Coverage

Frequency: 150 MHz to 1500 MHz
Mobile Station Antenna Height: between 1 m and 10 m
Base station Antenna Height: between 30 m and 200 m
Link distance: between 1 km and 20 km.

Mathematical formulation

Hata Model for Urban Areas is formulated as:

$$\mathbf{L_U = 69.55 + 26.16 \log f - 13.82 \log h_B - C_H + [\,44.9 - 6.55 \log h_B]\, \log d.}$$

For small or medium sized city,

$$\mathbf{C_H = 0.8 + (1.1 \log f - 0.7)\, h_M - 1.56 \log f.}$$

and for large cities,

$$\mathbf{C_H = 8.29\ (\log (1.54\, h_M))^2 - 1.1\,, \qquad \text{if } 150 \le f \le 200}$$

$$\mathbf{C_H = 3.2\ (\log (11.75\, h_M))^2 - 4.97\,, \qquad \text{if } 200 \le f \le 1500}$$

Where,

$\mathbf{L_U}$ = Path loss in Urban Areas (dB)
h_B= Height of base station Antenna. (m)
h_M = Height of mobile station Antenna. (m)
f= Frequency of Transmission (MHz).
C_H = Antenna height correction factor
d= Distance between the base and mobile stations (km).

The term "small city" means a city where the mobile antenna height not more than 10 meters. i.e.
$1 \leq \ h_M \ \leq \ 10m$

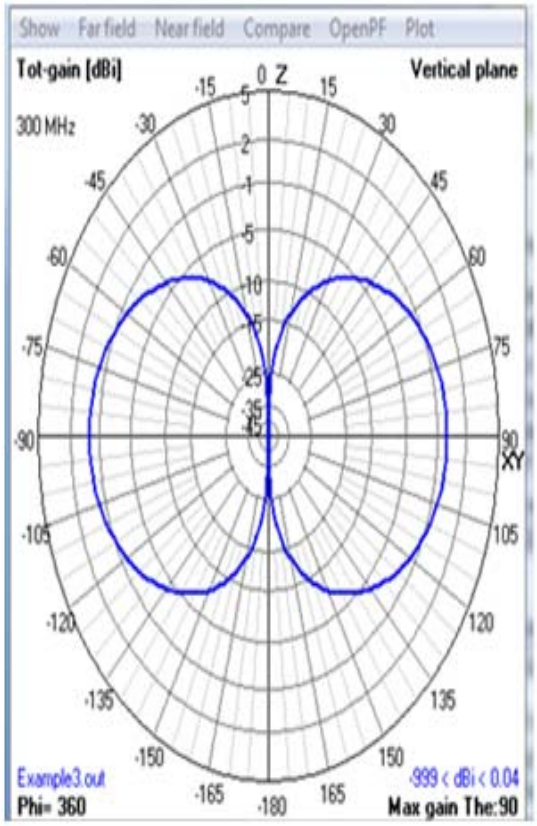
Program:

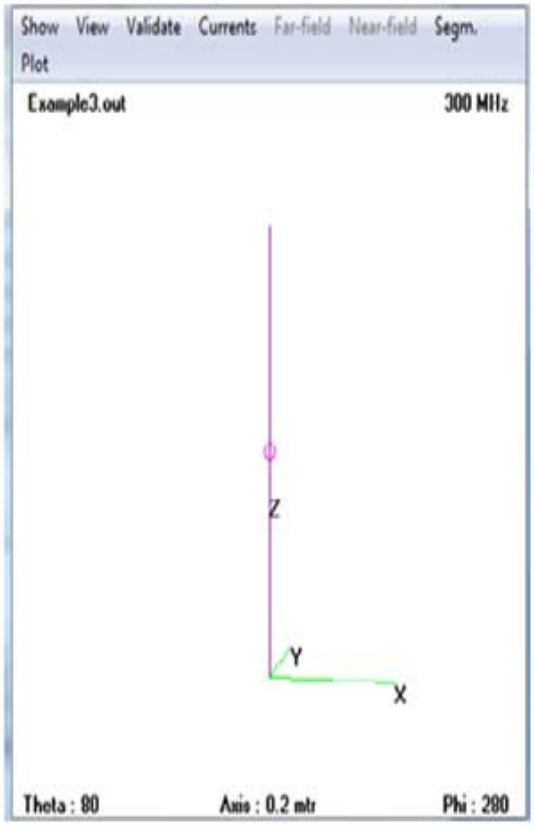
```
clc;
clear all;
close all;
f=input('enter the frequency of transmisson in mhz:');
Hb=input('enter the height of base station Antenna in meter:');
Hm=input('enter the height of mobile station Antenna in meter:');
d=input('enter the distance between the base and mobile stations:');
n=input('enter 0 for small city and 1 for large city:');
if n==0
   ch=0.8+(1.1*log10(f)-0.7)*Hm-1.56*log10(f);
else
  if f>=150 && f<=200
   ch=8.29*(log10(1.54*Hm))^.2-1.1;
  else
     if f>=200 && f<=1500
        ch=3.2*(log10(11.75*Hm))^.2-4.97;
     end;
  end;
end;
Lu=69.55+26.26*log10(f)-13.82*log10(Hb)-ch+(44.9-6.55*log10(Hb))*log10(d);
disp(sprintf('%s %f %s','Path loss in Urban Areas=',Lu,'db'));
```

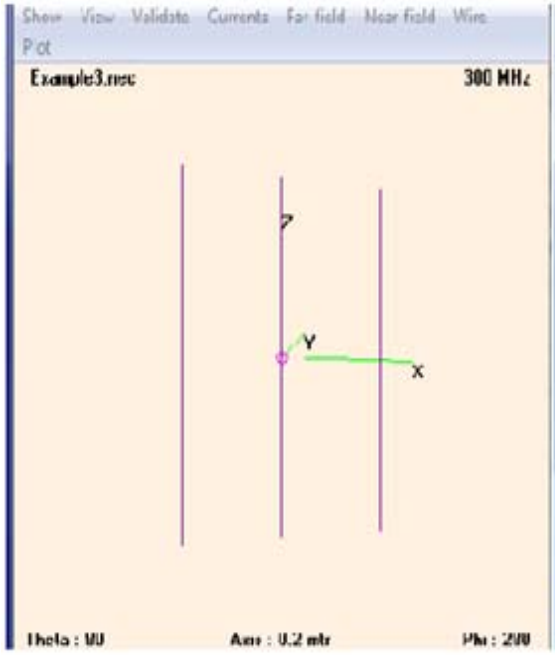
Result:

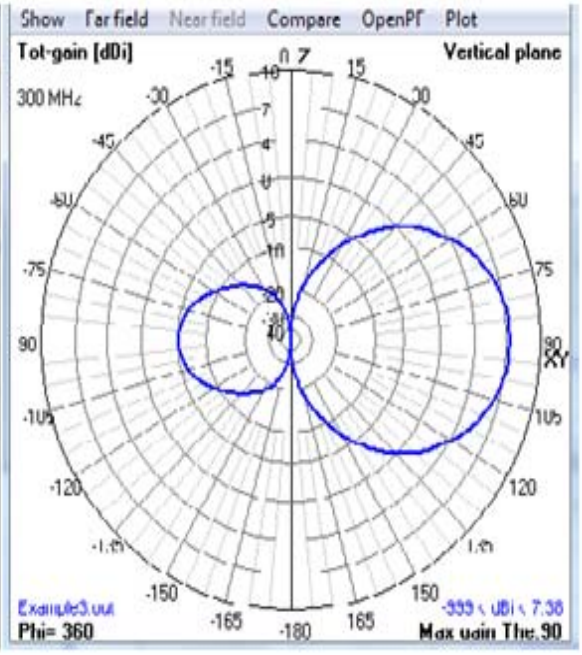
The program for Hata Model – Outdoor Propagation was simulated successfully.

OUTPUT:


Show Far field Near field Compare OpenPF Plot
Tot-gain [dBi]
Vertical plane
300 MHz
0 Z
XY
Example3.out
Phi= 360
-999 < dBi < 0.04
Max gain The:90


Show View Validate Currents Far-field Near-field Segm.
Plot
Example3.out
300 MHz
Z
Y
X
Theta : 80
Axis : 0.2 mtr
Phi : 280


Show View Validate Currents Far field Near field Wire
Plot
Example3.nec
300 MHz
Z
Y
X


Show Far field Near field Compare OpenPF Plot
Tot-gain [dBi]
Vertical plane
300 MHz
0 Z
XY
Example3.out
Phi= 360
Max gain The:90

ANTENNA DESIGN CONCEPT (USING 4NEC2)

NUMERICAL ELECTROMAGNETICS CODE - 4NEC2

Introduction :

NEC is the result of efforts by G. J. Burke and A. J. Poggio of Lawrence Livermore Laboratory.The Numerical Electromagnetics code (NEC-2) is a computer code for analyzing the electromagnetic response of an arbitrary structure consisting of wires and surfaces in free space or over a ground plane. The analysis is accomplished by the numerical solution of integral equations for induced currents. The excitation may be an incident plane wave or a voltage source on a wire, while the output may include current and charge density, electric or magnetic field in the vicinity of the structure, and radiated fields. NEC-2 includes several features not contained in NEC-1, including an accurate method for modeling grounds, based on the Sommerfeld integrals, and an option to modify a structure without repeating the complete solution.

The integral equation approach is best suited to structures with dimensions up to several wavelengths. Although there is no theoretical size limit, the numerical solution requires a matrix equation of increasing order as the structure size is increased relative to wavelength. Hence, modeling very large structures may require more computer time and file storage than is practical on a particular machine. In such cases standard high-frequency approximations such as geometrical optics, physical optics, or geometrical theory of diffraction may be more suitable than the integral equation approach used in NEC-2.

Getting Started :

Contents:

1) Create an antenna model using 'Geometry Edit'.
2) Show structure; generate data and view currents and phase distribution.
3) Generate Far-field data and view 2D polar and 3D far field patterns.
4) Generate frequency loop graphical-data.
5) Optimize antenna performance.
6) Sweep antenna variables.
7) Generate and view Near-field data.

- Create an antenna model using 'Geometry Edit'

As a starting-point we will create a basic 20-meter dipole fed with 50-ohm feedline. To show none-metric unit usage we consider ourselves for a moment as an US-citizen using Inch and Feet as the basic length unit....

The below specifications apply:

- antenna height 70 feet
- wire length 33.7 feet

- wire radius #12 AWG
- feedline length 69 feet (electrical length)

First we open one of the existing example files (e.g. 36dip.nec). If not already done so, specify 'Geometry Edit' as the prefered Edit method using the 'Settings' menu on the 'Main' window. Furthermore select 'Feet' as the length-unit and 'Inch/Awg' as the 'radius-unit' using this same 'Settings' menu.

When done, select 'Edit -> Input-file' on the 'Main' window or use the <F6> key to start 'Geometry-edit'. A picture of the selected example file is displayed. If not already set, select 'Options -> Set Segmentation -> Medium' on the Edit window to set medium segmentation density

To create a new model, Select 'File -> New' on the Edit-window.

- Setting design frequency

When starting a new model, initially on the lower right, frequency 'data' is displayed. This because one of the first things we will have to do is specifying the antenna design-frequency. Enter 14.15 (MHz) in the 'frequency' text-box on the right part of the window. When entered, click the 'wire' button on top of the window (the one with the single line in it). Notice the grid-size changing from .025 to .5 feet, correspon- ding to about half a wavelength for the window-width. Furthermore the default 3D- display view is now set to 2 dimensional XZ plane. (The Y-axis is pointing backwards)

- Add new wire(s)

To start adding a new wire, click the 'Add' button. The mouse-pointer changes to a cross-hair, indicating 'Add-mode' is activated. The Y-position text-box on the right is now highlighted. If required you can specify a certain 'depth' position, but for now we will stay in the XZ-plane for an Y-position equal zero.

When you will try to locate a mouse-pointer position for which the height Z equals 70 feet, you shouldn't succeed, because the grid-size is too small to cover a Z posi- tion of 70 feet. First increase grid-size to 1 feet by clicking on the left arrow for the 'Zoom' scroll-bar. When done, point somewhere inside the picture-box (that part of the window where the antenna structure is displayed), hold down the right mouse-button and move the X-axis to almost at the bottom of the picture-box. Now you should be able to locate a position for which Z equals 70 feet somewhere in the upper region of the picture-box.

Because we want to create a line at Z=70 feet with a length of 33.7 feet we will have to locate a point for which Z = 70 and X = -33.7/2 = 16.85 feet. However, because the 'Snap to grid' box is checked you won't succeed in this. For now we will locate a position for which X equals -17 feet.

To start drawing the wire, click and hold down the left mouse-button and drag the mouse-pointer to the second position for which Z=70 and X=17 feet, then release the mouse-button. Because this is the first wire added to the model, a pop-up window is displayed asking for the initial/default wire diameter. Use the supplied default of

.05 inch. Now the first wire, the dipole itself, is created. On the right of the picture-box all data belonging to this wire is listed. You can edit the end-1 or end-2 coor- dinates text-boxes to further refine the end positions. You will also notice that the number of segments is set to 25, corresponding to 'medium segmentation'.

The second wire, to connect the other end of the feedline, is done the same way by drawing a line from x=-1 to X=+1 for a height Z=1. This second wire is automatically divided into 3 segments. Because we only need this wire to attach one end of the feed-line, manually change the number of segments to 1.

Don't worry in case you did not position the wire-ends at the right coordinates. To move wire-ends, click the 'pointer' button and place the mouse-pointer on the wire- end to move. The mouse-pointer should now change to four-arrows, meaning you can move the wire end. Hold down the left-mouse button and move the wire end to the required position. As an alternative you can also directly edit the end-1 or -2 XYZ values.

- Add feed/transmission-line(s)

Next we will have to add the feedline. But, before doing so we need some knowledge about how wires are identified in Nec-2/4. All wires are assigned a unique tag-nr. Mostly the tag-number equals the wire-number. Ater delete, copy or paste operations however this sequence may have changed. Tag-numbers should still be unique. You may use 'Resequence tag-numbers' in the 'Option' menu to make tagnumbers equal to the corresponding wire-numbers again. Each voltage/current-source, transmission-line or RLC-load (see below) is 'assigned' to a wire using this unique tag-number. To specify the position of the source, TR-line or load on the specified wire a segment-number between 1 and the nr-of-segments for the wire is used.

Using Geometry-edit these tag- and segment-numbers are automatically assigned. It is allowed to change these number manually. When doing so please note how these tag- and segment-numbers are used within Nec-2/4.

Adding/creating a transmission-line is done by clicking the 'TR-line' button (the one with the ladder picture). If not in 'Add-mode', click the 'Add' button to start adding a new Tr-line. Locate the mouse-pointer on the middle of the first wire and click and hold down your left mouse-button and move the mouse-pointer to the middle of the second wire. When reached release the mouse-button. When positioning was not too rude a new transmission-line is now added. If not, try again. Note also that we did not take the velocity-factor into account, we just used an electrical length of 70-1=96 feet.

- Add voltage source

To prevent loosing the changes, backup the model using 'File->Save as' and choose a folder- and file-name for your new model. The next thing to do is add a voltage-source. hile still in Add-mode, click the 'Source button' (right of the 'Wire button'). Next click and hold down your left mouse-button somewhere in the picture-box. At the current mouse-pointer position a new source-object is displayed. Drag the

source-object to the middle of the second wire, just between the two lower wires-ends of the feedline and release the mouse- button. When properly positioned a new source is now added. If not try again.

For now we will set a default voltage-source of 1+j0 volt (1V @ 0 deg.)

Select the 'pointer' button to switch back to select-mode mode. The mouse-pointer changes to the default arrow-pointer indicating 'Select-mode' is active.

- Add wire-conductivity

We use copper wire for our antenna, so we will have to include this in our model (the default is perfect wire with zero losses). To do this, click the 'Loading button' (the one with the RLC symbols), click somewhere in the picture-box and drag the new load-object to any place on the first (upper) wire and release the mouse- button. The new load-object is now 'connected' to the first wire.

The default load however is a lumped load. To change this to a distributed/wire-load, change the 'Par-RLC' selection for the Load-data on the right of the screen to 'Wire-ld'. The box shape on the first wire should now have changed to a red line- segment. The initial conductivity is set to 10000 mho/m. Change this to 'Copper' by using the 'G (mho/m)' selection-box on the right of the picture.

To specify wire-conductivity for the whole structure, first change from 'spot load' to 'single-wire' (see lower right) . Notice how the whole wire becomes 'wire-loaded'. Next change to 'Whole struct'. The 'Wire-conductivity' is not visible any more. This would not deliver us additional information because the whole structureis now loaded (both wires). To enable wire-loading display for the complete structure use 'Option -> Show wire loading'.

For now we have added all required objects. Switch back to 'Select-mode'.

- Select/move objects

The attentive user will have noticed that we did not yet explicitely specify our wire radius (half the diameter). We will do so now. While in 'Select-mode', click the wire button and select the upper wire (wire 1). The wire color will change to red (if not already set), indicating this wire is selected for modification. In the wire-data on the right part of the screen select #12 as the wire-radius for wire 1. Select wire 2 and change the radius also to #12.

Also click the 'Trans-line' button to set the 'Char-Imp./Z0' to 50 ohms.

To move a wire, click the 'Wire' button again and select the required wire. When the mouse-pointer is over the selected wire the pointer changes from the default to a two-point or four-point indicator. When a two-point indicator is visible one can move the whole wire at once. When a four-point indicator is visible one can move the corresponding wire-end. Move a wire(end) by

clicking on the wire(end) and hold down the left mouse-pointer while dragging it to a new position and releasing the mouse- button.

When XY, XZ or XY plane (2D) is selected you can move a wire(end) to any place inside the picture-box. When 3D-view is selected however be careful with this, because a moved wire(end) automatically connects to the another near wire-end. To undo the latest move-action use the 'Edit' menu or move the mouse-pointer over the wire(-end) till a two/four-point mouse-pointer becomes visible and click the right mouse-button. A pop-up window is now displayed in which you can select 'Undo move'. The same principles apply for moving sources, loads and transmission-lines. However these objects can only be moved from one wire(segment) to another wire(segment).

You can textually change/edit XYZ-, wire-, tag- or segment-position by selecting the required object and modify the 'object' data on the right of the screen.

If required, backup or save your model by using 'File->Save(as)' to be able to restore a previous model in case you made a serious mistake.

- Specify ground parameters

By default a new model is located in 'Free space'. To model an antenna over ground, select the right-most 'Ground Params' button and select between Free-space, Perfect-, Finite- or SomNec-ground. For now we will use the finite-ground, also know as 'fast- ground'. Change from 'User-specified' to 'Average' (Clay/Forest) ground. Conductivity is now automatically set to 0.005 Siemens and 'Diel-const' is set to 13.

To view the corresponding NEC-syntax for all wires and other objects we created,
use 'Options->View Nec data'.

- Run NEC-engine and create far-field pattern.

To run the NEC-engine and evaluate your model, click the 'Run Nec-engine' button (the one with the calculator picture) or push <F7>. A new pop-up window is displayed asking you for additional settings. To create a full 3D far-field pattern, select the second option 'far-field pattern', specify 'Full' and a 5 degree resolution.

Then click <Generate>.

If the DirectX based version of 4nec2 is installed, push <F9> to visualize the new antenna-structure Select 'Pattern' or push the 'R' key to see the 3D far-field pattern.

You can return back to your model by pushing <F6> or clicking the 'Geometry-edit' window. You may alter your model (e.g. set to 'Free-space') and recalculate to see the results of your changes.

For another example of creating a T-antenna on a box, see appendix A at the end of this document.

- Show structure, generate data and view currents and phase distribution.

In this next example it is explained, how to open a NEC antenna model, view/edit (4)nec(2) input-file data the traditional way, generate NEC-output, examine and validate structure geometry and display the current- and phase-distribution along the structure. Furthermore some of the more general menu-bar options as available on the different 4nec2 forms/windows are discussed.

After starting the 4nec2 program by double clicking on the 4nec2 shortcut or on the 4nec2.exe program-file, a file selecttion window is displayed. This initial window is used to select the (4)nec(2) antenna model file to open and work with. In this first example please locate the file ..\4nec2\example1.nec and click the open button.

If no NEC-output is generated yet for the selected file, the data loaded into 4nec2 will be that for the (4)nec(2) input-file. The wire geometry structure specified in this file is displayed on the 'geometry' form. You may use the F2 or F3 key's to bring the 'Main (F2)' or the 'Geometry (F3)' form to the foreground. To indicate that the you are currently viewing the input-file data, the background for the 'Geometry' form is displayed in a none white color. Note also that in this case, most of the fields on the 'Main (F2)' form are empty.

You may use the arrow key's to rotate, shift or zoom the structure, or the Page-up and Page-down key's to zoom-in or -out. To shift the structure up/down or left/right you can also use the Control key together with one of the arrow-key's. Use the 'Home' key to reset the geometry form. If you have installed the 4nec2X extended version you could use the F9 key to view your nec-model using real-time 3D rendering techniques. Use your mouse-buttons (left, right or both) to rotate, shift and zoom the model.

- Generate Far-field data and view 2D polar and 3D far field patterns.

To generate a 3D far-field pattern, press the F7 key and select the second option called 'Far-field pattern'. In the lower half of the form, additional fields are displayed to specify a certain pattern-resolution and a check-box to include the surface wave into the combined far-field pattern.

The pattern-resolution specified how fine or course the generated pattern is. Fur- thermore this affects 4nec2 memory-usage and NEC processing-time. For 'simple' antennas like our dipole a resolution of 5 or 10 degrees will be fine. For multi- element antennas like the 'emeyagi.nec' a reolution of 1 degree may be needed. For now, the 'surface wave' option should not be selected. The option boxes on the right should be set to 'Default pattern'. Experienced NEC-users can use one of the other options or even use the 'more' button to get extra options.

When the 'Generate' button is pushed, the NEC-engine starts and new output data is generated. After the calculations are done a third form called the 'Pattern' form is displayed. In this form the 2D horizontal or vertical polar far-field patterns are made available. If this form is on top, with

the arrow-keys you can select the pattern for different theta or phi angles. With the 'G'(eometry) key or the 'Show-> Structure' the geometry structure is displayed on the pattern form.

To view the 3D pattern, select the 'Geometry' form (F3) and push the 'R' key or use the 'Show->Near/Far-field' option. You may use the mouse-buttons or the arrow- and page-up/down keys to move, rotate or zoom the 3D pattern. If the 3D pattern on the 'Geometry' form is enabled and the 'Pattern' form is selected (F4), the color for the 3D pattern changes and the 2D pattern for the selected theta (elevation) or phi (azimuth) angle is highlighted. This helps you to understand where the selected 2D pattern is located in the full 3D-pattern.

On the 'Pattern' form you can use the 'L' key t switch between linear and (semi) logarithmic scaling. By default the pattern is normalized for maximum gain for the current Theta(elevation)/Phi(azimuth) angle. The Max-gain value is displayed in the upper left corner. To normalize against the overall maximum gain, press the <Home> key. To disable all normalization, press the <Home> key again. A third push will bring you back to the efault state.

To get the gain and angle for a particular point on the pattern it is possible to select a point on the pattern line with the mouse and click the right mouse button. Use the 'I'(nfo) key or 'Show->Info' the get additional information about maximum gain, front to back ratio and beam-width. By default the 'total field' is displayed, to view the other generated patterns use the ','(<) and '.'(>) key's.

Use the 3D-viewer/<F9> (4nec2X only) to view the far-field data in 3D perspective.

- Generate frequency loop graphical-data

To generate frequency loop (frequency sweep) data, Enter the F7 key, and select 'Use frequency loop'. With this calculation-option line-chart graphs are generated for Forward-gain, Front-to-Back ratio, Front-to-Rear-ratio, SWR and input impedance.

When selecting this option additional input-boxes appear. For now we select the 'Gain' option. Please enter a frequency start-value of 3.5, a stop value of 4 and a step-size of .02 Mhz. Enter a value of 90 for the Phi angle and a value of 55 for the Theta angle, and click the 'Generate' button.

When calculations are done a third window is displayed called the 'Line-chart (F5)' Window. In this window you can switch between "S"(SWR), "G"(Gain) and "I"(impedance) display. Use the "L" key to switch between linear and logarithmic Y axis scaling. Use the "F" key to change to X-axis scaling. By default the SWR, R-in and Z-in graphs are set to logarithmic, the others default to linear. When linear scaling is set you can use the 'Up','Down', 'Page-up' and 'Page-Down' keys to move and zoom the graph. Use the 'Tab' key to select one or both graphs.

4nec2 also has the possibility to display the input impedances on a Smith chart. Enter the F11 key to select this option. Use the cursor keys to select a specific frequency. More experienced users may use the <Shift> key in conjunction with the cursor keys to 'add' a certain length of feedline. Use <Home> to (de)normalize. To view the changing for, for example, the vertical far-

field pattern when fre- quency increases from 3 to 30 Mhz, please enter F7, 'use frequency loop' and select the 'Ver'tical option. Enter 3, 30 and .5 for frequency start, stop and step-size. Again enter 55 for the Theta- and 90 for the Phi-angle of our direction of interest. Click 'Generate' and when calculations are done, you can 'walk' through the different vertical far-field patterns on the 'Pattern' (F4) form with the 'Left' and 'Right' arrow key's.

Note: Select the Nec2dSX engine for increased accuracy when running a frequency-loop using SomNec ground settings.

- Optimize antenna performance.

In this example again the 'Example3.nec' input file is used, but now we will opti- mize antenna-performance. As a first try we will use the traditional hill-climbing optimizer and optimize radiator length for resonance. To do this, start the Optimi- zer by entering the F12 key. A new window appears with a number of selection- and input-boxes. First we set the traditional optimizer by selecting 'Optimize' in the Function-box and 'Default' in the Option-box. After this we select the variable(s) we want to optimize, by clicking on the 'len' variable in the list-box with the 'variables' heading. The selected variable(s) will show-up in the right list-box.

Furthermore you must select one or more antenna properties to optimize, together with their "importance" (weighting factor, contributing in the total result). To optimize for resonance, please enter a value of 100 (%) in the 'X-in' box, (all other values must be set to zero) meaning only the Reactive component con- tributes for 100% in the total result. (FOM, figure of merit). To get resonance, this property must be minimized. This is the default for the 'X-in' property. (Click with the right mouse key on one of the property-boxes to change this default target)

After clicking the 'Start' button the optimizing process starts and the button text changes to 'Stop'.

In the upper right box (value sensitivity), the selected variables together with the direction and relative amount in which they are changed are displayed. In the lower left box (calculated results) the calculated property values are displayed for each new optimization step, together with the calculated overall result (Res%) and the step-size used. In the lower right box (variabele values) the corresponding variable value(s) is/are listed, so it is possible to follow the optimizing process.

After some time the process should stop with the message 'Optimized in XX steps', indicating the optimization is ready. To premature abort the process, you may click the 'Stop' button. It is possible the process is not immediately halted. If so, please wait till the active calculation step is ready. Sometimes it may be necessary to click the button once more. After the process is ready/aborted, you may change the variables or properties and continue optimization by clicking the 'Resume' button.

If the optimization results are OK, you may use the 'Update NEC-file' button to up- date your NEC-file with the new variable value(s). Use 'Exit' to quit the optimizer without saving.

In the same way you can optimize for Forward-Gain, Front-to-back- and/or Front-to- Rear-ratio. If one or more of these properties are selected, you must also specify the Forward- and backward-gain angle(s) for which the Gain has to be calculated for.

For quick optimizations, a resolution of "0" (zero) could be used. In this case only the Gain for the specified Forward- and backward-angles are calculated and no additional Front-to-Rear data is calculated.

For more precise optimizations, a none-zero resolution (e.g. 5 degrees) could be set. Now a complete 3D pattern is calculated for each optimization step, so the difference between the Forward lobe and the largest side-lobe in the backward 180 degree part of the pattern is calculated and displayed as the Front-to-rear ratio.

If optimizing for Gain or F/B, one may also specify a delta Theta and Phi for the forward- and/or the backward-angle. If a none-zero value is specified, the gain is averaged over the range between Phi-delta_phi and Phi+delta_phi. The same holds for the Theta angle.

Mostly optimization is performed for total-gain. If required, however you may opti- mize for horizontal/vertical-gain or E-theta/E-phi. Optimization with included sur- face-wave is also possible.

Variable changes are reflected on the Geometry view. To view them, after starting the optimization process, please move and/or resize the optimizer window to the lower left part of the screen. If optimization is done for Gain and a none-zero resolution is set, the far-field pattern changes are also reflected on the Geometry (if 3D pattern is enabled) and the Pattern form. The optimization steps are logged in the optimzer.log log-file. This file can be viewed with 'Show -> Optimizer log' in the Geometry window.

- Generate and view Near-field data

As an example to generate near-field data, the file NearFld.nec is included in the package. To use this example, please load this file and push the F7 key to get the 'Generate' window. Select 'Use original file', to start the calculation. This input-file already contains the required NE card is, so it is not yet necessary to specify any near-field parameters. Calculations will take some time, because almost 30.000 near-field points are calculated.

When calculations are done, the Pattern windows is displayed with the 'Near-field' lay-out. Initially you mostly will see a blue plane with on the left a color-bar telling on the left, telling you what field-strength is represented by a certain color. The maximum value will be in the range > 1e+4 volts/m. This is due to the fact that one or more of the calculation points will be very close (or maybe on) a geometry wire with high RF-voltage/current.

To get rid of these 'unusable' high values, please use 'Near-field -> Ignore high values" or push the <Del> key. You are asked for maximum field-strength. Please enter a value of 250 (V/m). The display should now change to a more colorful view. The color-bar on the left is also updated.

Another way to change the color-resolution, is to use the <Alt> together with the <Page-up> or <Page-down> keys. This will change the linear scale to a more logarith- ic scale. Use 'Show -> Geometry' or the "G" key to display the geometry- structure.

Initially the field-strength on the XY plane for a certain Z value is displayed. To change the Z-value use the cursor-left or -right key's. To change between XY, YZ and XZ plane, use the spacebar.

On the Geometry form you are also able to display the Near-field points. Use 'Show -> Near/far field" or push the "R" key. Use the spacebar to switch between 3D view, or 2D XY, YZ or XZ view. Use <Shift> plus the arrow-keys to change the 2D plane coordinates. Use <Page-up>/<Page-down> to change the dot-size.

On the 'Pattern' form on the left and on the right of the 2D plane, two black indicator lines are displayed. These indicators are used to select a certain Y- or Z value for displaying the field-strength using a line-chart. Use the Up/Down arrow keys to change position, use <Page-Up>/<Page-down> to switch between 2D and line- chart viewing.

Use the 3D-viewer/<F9> (4nec2X only) or the "Plot -> 3D" menu-bar command (if gnuplot is availle) to view the far-field data in 3D perspective.

EXPERIMENT 3.a

DIPOLE ANTENNA

Aim

To simulate a dipole antenna (λ, λ/4, λ/2, 3λ/2) for a particular frequency using 4NEC2

Parameter Required

Frequency is the required parameter for designing an antenna. Let the frequency be 300 MHz.

Calculations

Length of the dipole is given as
$L = \lambda/2$
where λ=wavelength of the antenna

$\lambda = c/f$
where c=speed of light in m/s
f= frequency in MHz

Thus,
$\lambda = c/f = 300/f(MHz) = 300/300 = 1m$
and,
$L = \lambda/2 = 1/2 = 0.5m$

Similar calculations can be done for λ, λ/4 and 3λ/2

Simulation

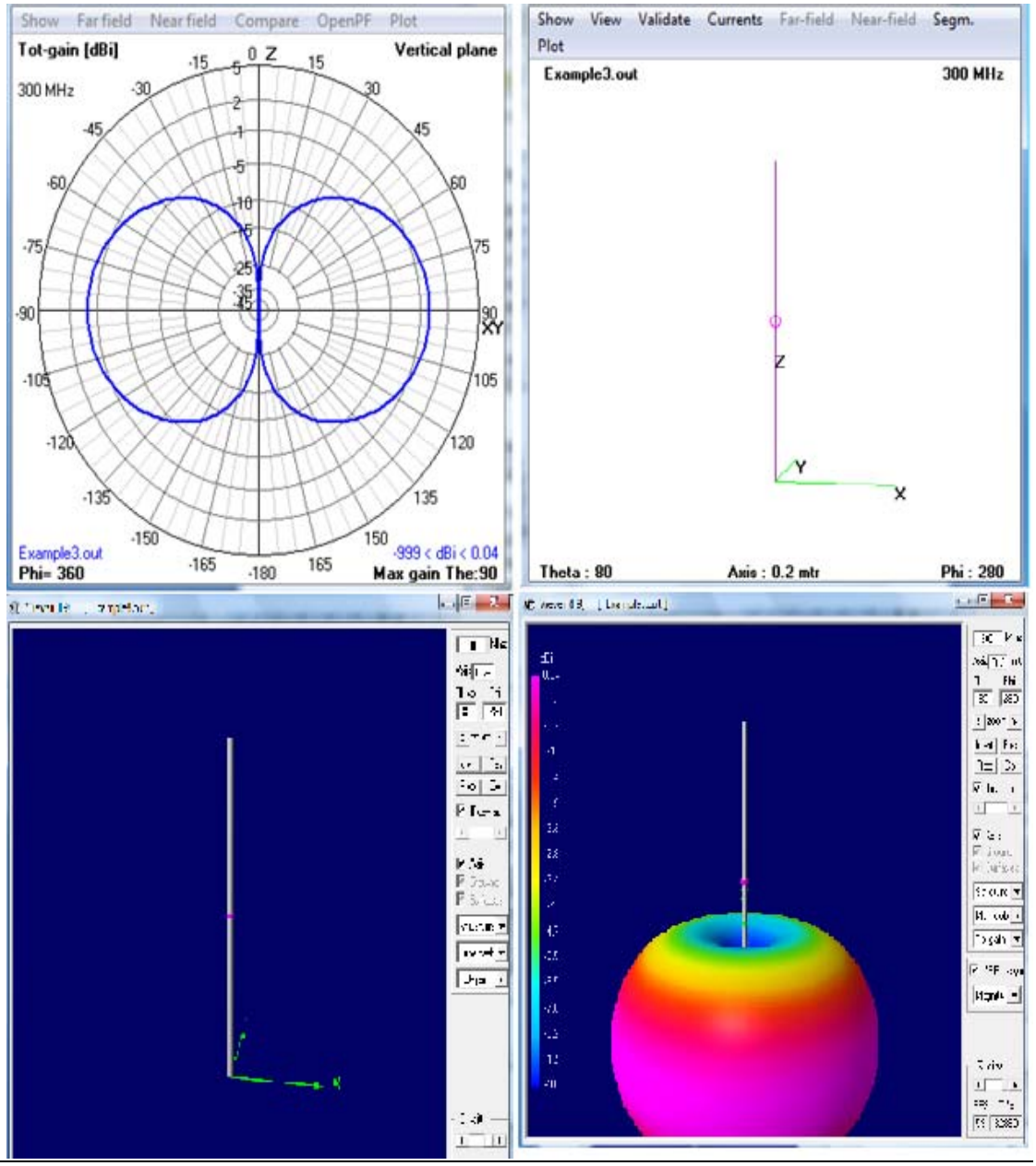
- Open the 4NEC2x main window. Click open file and select a pre-existing file to open it.
- Click Edit Input File, a geometry edit window will open.
- Now click on wire symbol and click add and then draw a wire of any length.
- Now specify the co-ordinates for the end 1 and end 2.(Note: Divide the length in two parts- one above the x-axis and other half below x-axis.
- Now click on source symbol and click add and add the source by dragging it to the centre of the wire. Specify source value.
- Click on frequency symbol and specify the frequency.
- Now click on generate button and select the options to be generated.

Result

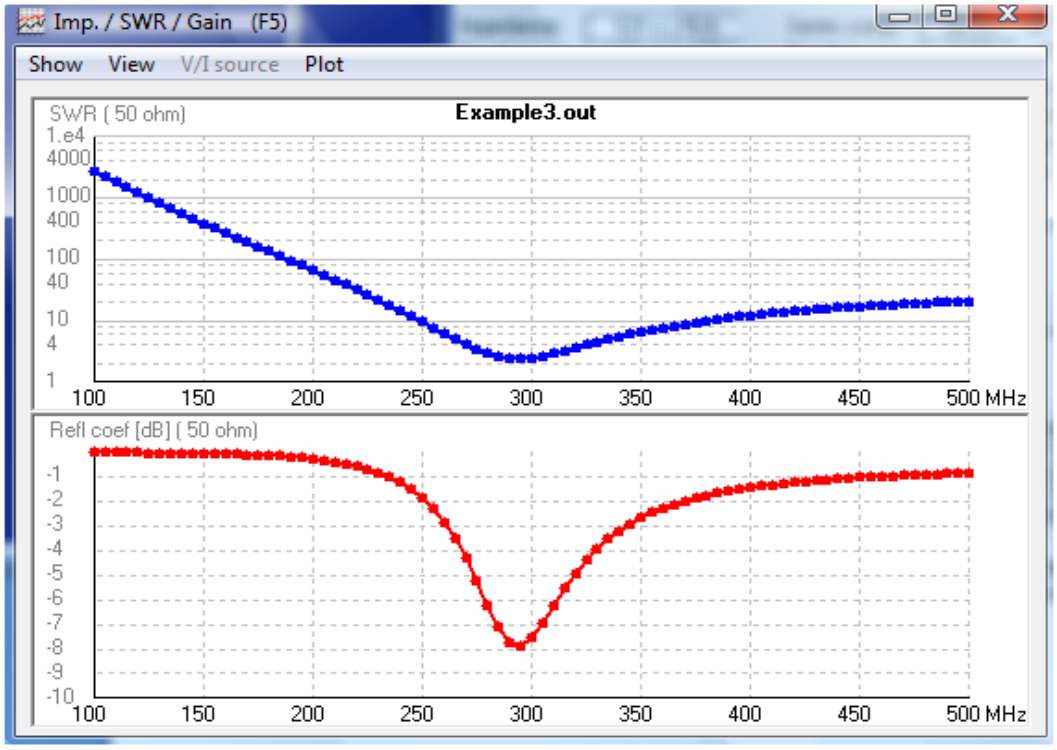
Thus the dipole antenna was simulated using the 4NEC2 software and various outputs could be inferred from the plots.

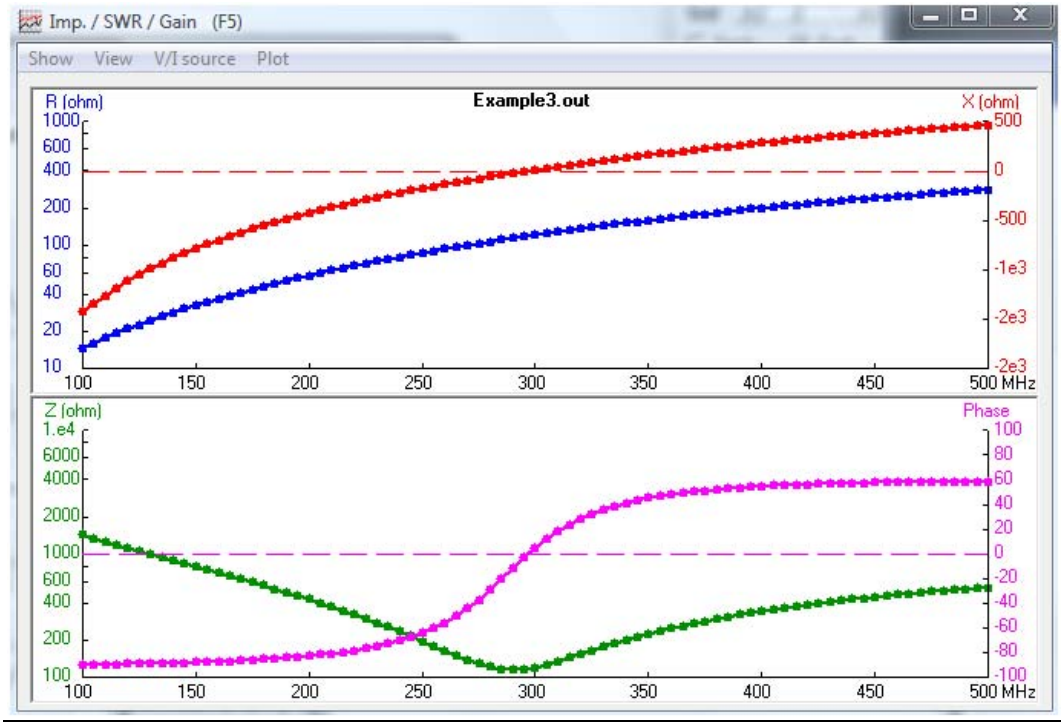
Output

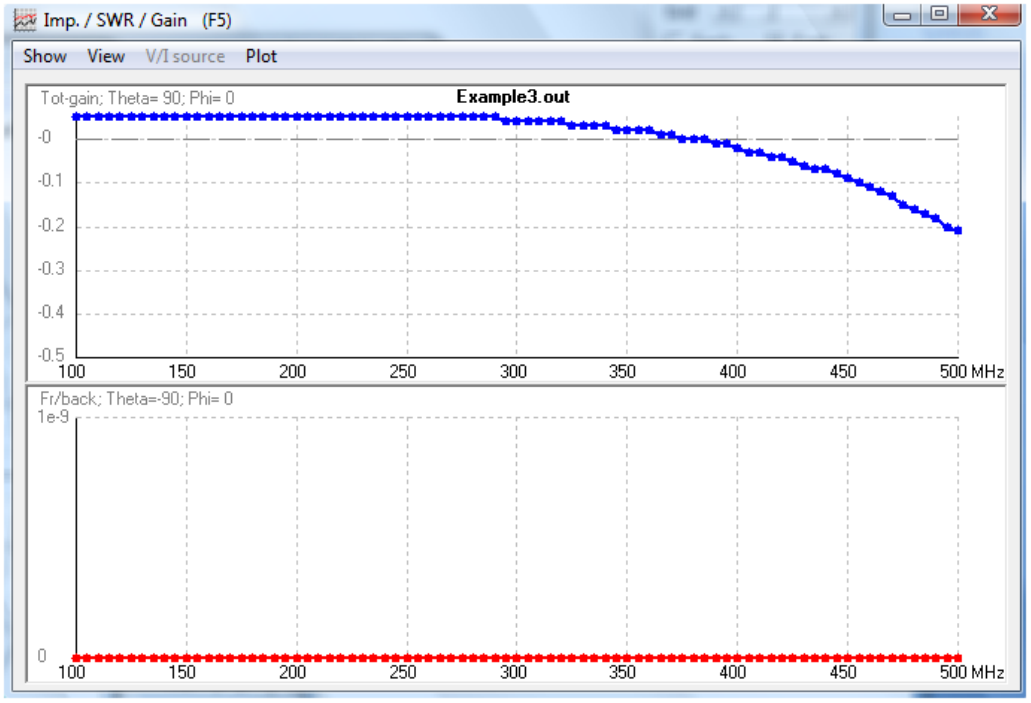
Far-field pattern (2D and 3D)



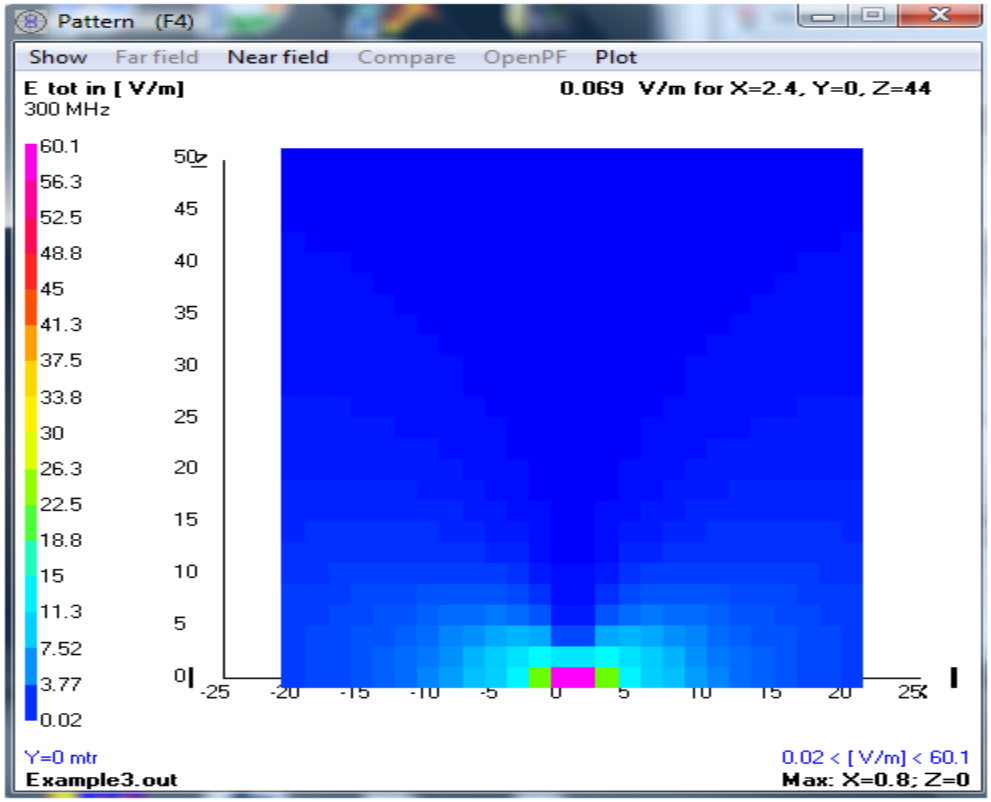
Frequency Sweep







Near Field Pattern



EXPERIMENT 3.b

YAGI-UDA ANTENNA (3 – ELEMENT)

Aim

To simulate a 3-element Yagi-Uda antenna for a particular frequency using 4NEC2

Parameter Required

Frequency is the required parameter for designing an antenna. Let the frequency be 300 MHz.

Calculations

Length of the driven element

$L = \lambda/2$
where λ=wavelength of the antenna.
$\lambda = c/f$
where c=speed of light ; f= frequency
Thus,
$\lambda = c/f = 300/f(\text{in mhz}) = 300/300 = 1m$
and,
$L = \lambda/2 = 1/2 = 0.5m$
Now we divide it by 2 so as to get the +ve z and –ve z axis coordinates.
Thus the z coordinates will be=0.5/2=0.25m

Length of the reflector

$L_R = L + 5\% \text{ of } L = 0.5 + 5\% \text{ of } 0.5 = 0.525m$
Now we divide it by 2 so as to get the +ve z and –ve z axis coordinates.
Thus the z coordinates will be=0.525/2=0.2625m

Length of the director

$L_D = L - 5\% \text{ of } L = 0.5 - 5\% \text{ of } 0.5 = 0.475m$
Now we divide it by 2 so as to get the +ve z and –ve z axis coordinates.
Thus the z coordinates will be=0.475/2=0.2375m

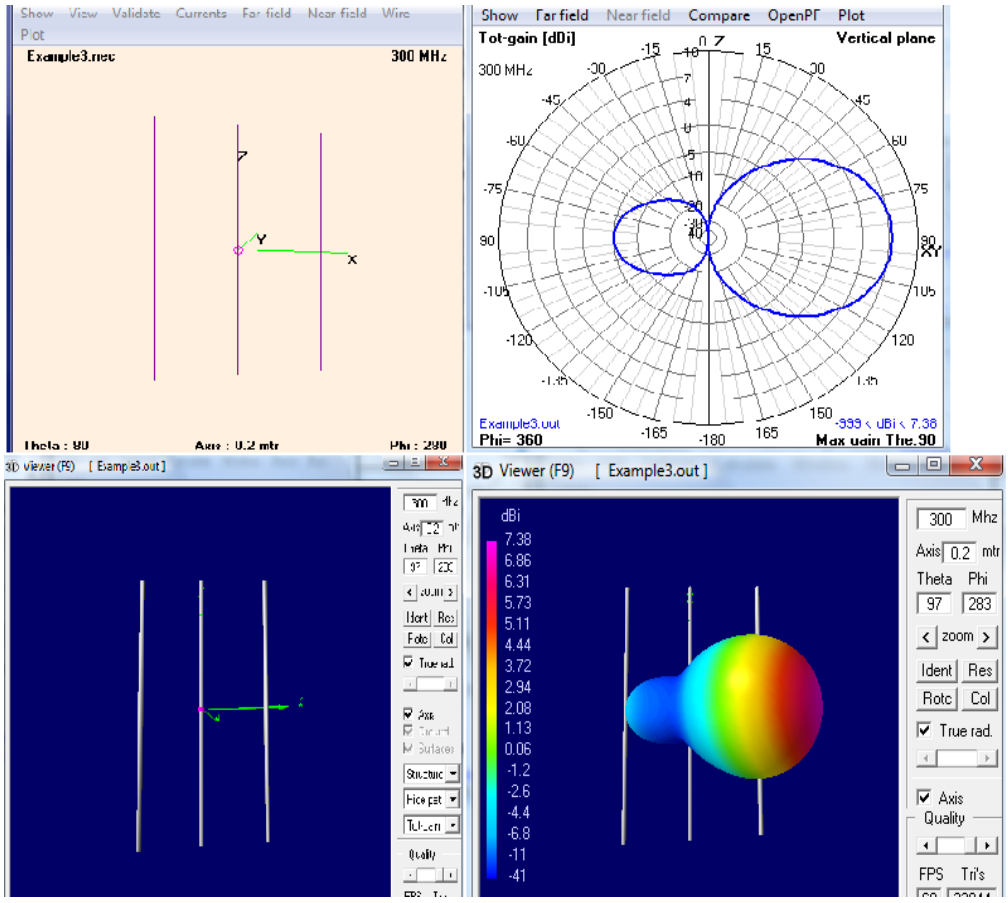
Simulation

- Open the 4NEC2x main window. Click open file and select a pre-existing file to open it.
- Click Edit Input File, a geometry edit window will open.
- Now click on wire symbol and click add and then draw a wire of any length.

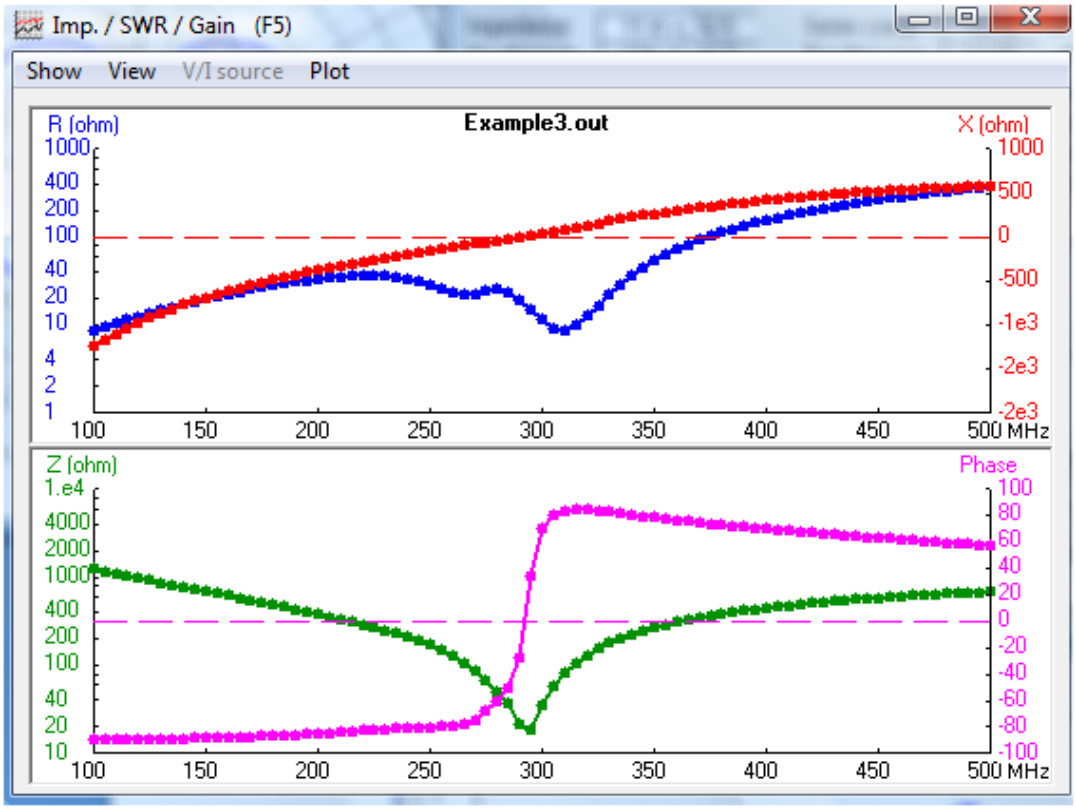
- Now specify the co-ordinates for the end 1 and end 2.(Note: Divide the length in two parts- one above the x-axis and other half below x-axis.
- Now click on source symbol and click add and add the source by dragging it to the center of the wire. Specify source value.
- Click on frequency symbol and specify the frequency.
- Now click on generate button and select the options to be generated.

Output

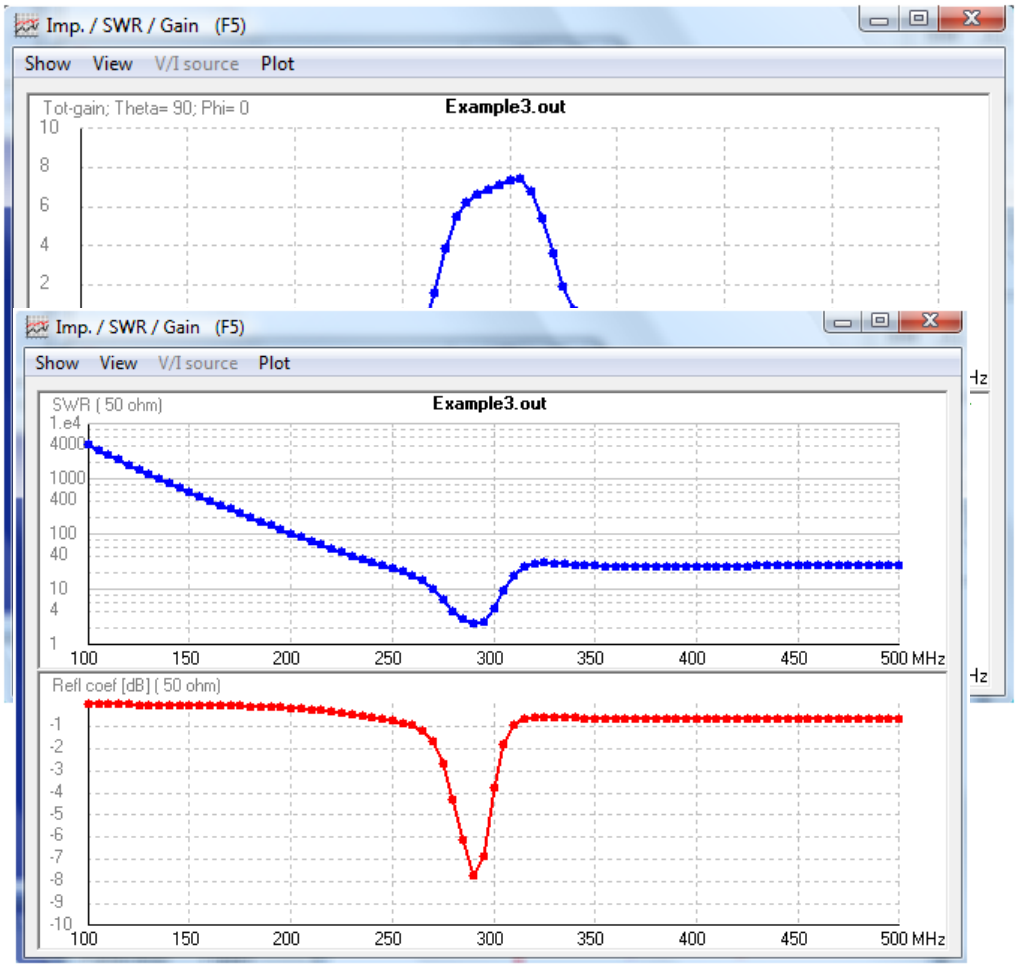
Far field Pattern (3-D and 2-D)

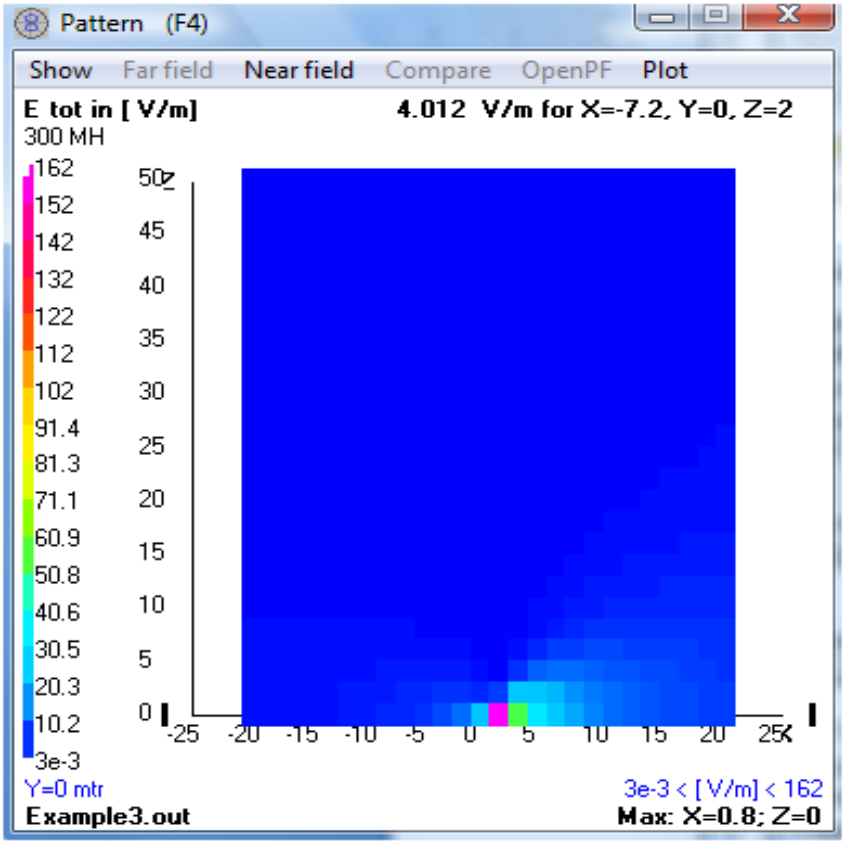


Frequency Sweep



Near Field Pattern





Result

Thus the 3- element Yagi-Uda antenna for 300MHz frequency was simulated using 4NEC2 and various outputs could be inferred from the plot.

EXPERIMENT 3.c

YAGI-UDA ANTENNA (5 – ELEMENT)

Aim

To simulate a 5-element Yagi-Uda antenna for a particular frequency using 4NEC2

Parameter Required

Frequency is the required parameter for designing an antenna. Let the frequency be 300 MHz.

Calculations

Length of the driven element

$L = \lambda/2$
where λ=wavelength of the antenna.
$\lambda = c/f$
where c=speed of light; f= frequency
Thus,
$\lambda = c/f = 300/f(\text{in mhz}) = 300/300 = 1m$
and,
$L = \lambda/2 = 1/2 = 0.5m$
Now we divide it by 2 so as to get the +ve z and –ve z axis coordinates.
Thus the z coordinates will be=0.5/2=0.25m

Length of the reflector

$L_R = L + 5\%$ of $L = 0.5 + 5\%$ of $0.5 = 0.525m$
Now we divide it by 2 so as to get the +ve z and –ve z axis coordinates.
Thus the z coordinates will be=0.525/2=0.2625m

Length of the director

$L_{D1} = L - 5\%$ of $L = 0.5 - 5\%$ of $0.5 = 0.475m$
Now we divide it by 2 so as to get the +ve z and –ve z axis coordinates.
Thus the z coordinates will be=0.475/2=0.2375m

$L_{D2} = L_{D1} - 5\%$ of L_{D1}
$= 0.475 - 5\%$of 0.475
$= 0.4512$ m

$L_{D3} = L_{D2} - 5\%$ of L_{D2}
$= 0.4512 - 5\%$of 0.4512
$= 0.4286$ m

Simulation

- Open the 4NEC2x main window. Click open file and select a pre-existing file to open it.
- Click Edit Input File, a geometry edit window will open.
- Now click on wire symbol and click add and then draw a wire oany length.
- Now specify the co-ordinates for the end 1 and end 2.(Note: Divide the length in two parts- one above the x-axis and other half below x-axis.
- Now click on source symbol and click add and add the source by dragging it to the center of the wire. Specify source value.
- Click on frequency symbol and specify the frequency.
- Now click on generate button and select the options to be generated.

EXPERIMENT 3.d

YAGI-UDA ANTENNA (7 – ELEMENT)

Aim

To simulate a 7-element Yagi-Uda antenna for a particular frequency using 4NEC2

Parameter Required

Frequency is the required parameter for designing an antenna. Let the frequency be 300 MHz.

Calculations

Length of the driven element

$L = \lambda/2$
where λ=wavelength of the antenna.
$\lambda = c/f$
where c=speed of light; f= frequency
Thus,
$\lambda = c/f = 300/f(\text{in mhz}) = 300/300 = 1m$
and,
$L = \lambda/2 = 1/2 = 0.5m$
Now we divide it by 2 so as to get the +ve z and –ve z axis coordinates.
Thus the z coordinates will be=0.5/2=0.25m

Length of the reflector

$L_R = L + 5\%$ of $L = 0.5 + 5\%$ of $0.5 = 0.525m$
Now we divide it by 2 so as to get the +ve z and –ve z axis coordinates.
Thus the z coordinates will be=0.525/2=0.2625m

Length of the director

$L_D = L - 5\%$ of $L = 0.5 - 5\%$ of $0.5 = 0.475m$
Now we divide it by 2 so as to get the +ve z and –ve z axis coordinates.
Thus the z coordinates will be=0.475/2=0.2375m

$L_{D2} = L_{D1} - 5\%$ of L_{D1}
$= 0.475 - 5\%$of 0.475
$=0.4512$ m

$L_{D3} = L_{D2} - 5\%$ of L_{D2}
$= 0.4512 - 5\%$of 0.4512
$=0.4286$ m

$L_{D4} = L_{D3} - 5\%$ of L_{D3}

= 0.4286-5%of 0.4286
=0.4072 m

L_{D5} = 0.4072-5% of 0.4072
=0.3868 m

Simulation

- Open the 4NEC2x main window. Click open file and select a pre-existing file to open it.
- Click Edit Input File, a geometry edit window will open.
- Now click on wire symbol and click add and then draw a wire oany length.
- Now specify the co-ordinates for the end 1 and end 2.(Note: Divide the length in two parts- one above the x-axis and other half below x-axis.
- Now click on source symbol and click add and add the source by dragging it to the center of the wire. Specify source value.
- Click on frequency symbol and specify the frequency.
- Now click on generate button and select the options to be generated.

Review Questions

1. What is free space?
2. Write expression for received power.
3. What is meant by path loss?
4. What is the relation between received power and distance?
5. Why received power mostly calculated in dBm?
6. For all the wireless communication why is it necessary to calculate free space propagation.
7. How loss will be evaluated in free space propagation?

Review Questions

1. What is the frequency used in uplink and downlink?
2. Why it is necessary to calculate link budget power equation?
3. What are the losses encountered in satellite communication?
4. Write the formula to calculate the freespace loss.