APPLICATION OF

VISUAL SIMULATION IN

COMMUNICATION SYSTEMS

A Project Report

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND INSTRUMENTATION ENGINEERING

By

Ranjeet Mohapatra(10407016)

Sameer Ranjan Behera(10407006)

Under the guidance of

Prof. S.K.Patra

Department of Electronics & Instrumentation Engineering

National Institute of Technology

Rourkela,769008 (2007-2008)

APPLICATION OF

VISUAL SIMULATION IN

COMMUNICATION SYSTEMS

A Project Report

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND INSTRUMENTATION ENGINEERING

By

Ranjeet Mohapatra(10407016)

Sameer Ranjan Behera(10407006)

Under the guidance of

Prof. S.K.Patra

Department of Electronics & Instrumentation Engineering

National Institute of Technology

Rourkela,769008 (2007-2008)

National Institute of Technology

Rourkela

CERTIFICATE

This is to certify that the thesis entitled, "Application of Visual Simulation in communication systems" submitted by Sri Sameer Ranjan Behera and Sri Ranjeet Mohapatra in partial fulfillments for the requirements for the award of Bachelor of Technology Degree in Electronics & Instrumentation Engineering at National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

Prof. S. K. PATRA
Dept. of Electronics & Instrumentation Engg
National Institute of Technology
Rourkela - 769008

ACKNOWLEDGEMENT

We place on record and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our guide ***Prof. S.K.Patra,*** *Professor, Department of Electronics and instrumentation Engineering, National Institute of Technology, Rourkela, in bringing this report to a successful completion.*

We are grateful to ***Prof. G.Panda****, Head of the Department of Electronics and instrumentation Engineering, for permitting us to make use of the facilities available in the department to carry out the project successfully. Last but not the least we express our sincere thanks to all of our friends who have patiently extended all sorts of help for accomplishing this undertaking.*

Finally we extend our gratefulness to one and all who are directly or indirectly involved in the successful completion of this project work.

.

Ranjeet Mohapatra(10407016)

Sameer Ranjan Behera (10407006)

CONTENTS

LIST OF FIGURES

ABSTRACT

A communications system is a collection of individual communications networks, transmission systems, relay stations, tributary stations, and data terminal equipment (DTE) usually capable of interconnection and interoperation to form an integrated whole. The components of a communications system serve a common purpose, are technically compatible, use common procedures, respond to controls, and operate in unison. A typical communication link includes, at a minimum, three key elements: a transmitter, a communication medium (or channel), and a receiver. The ability to simulate all three of these elements is required in order to successfully model any end-to-end communication system. In order to achieve this target we have used a simulation software “VisSim” ,or Visual Simulator ,that allows us to use a graphical approach to simulation and modeling.

With graphical programming, the diagram *is* the source code, depicted as an arrangement of nodes connected by wires. Each piece of data flows through the wires, to be consumed by nodes that transform the data mathematically or perform some action such as I/O. The visual simulator allows us to model end-to-end communication systems at the signal or physical level. We use VisSim/ Comm to build both transmitter and receiver models, filters and equalizers, as well as channel models and coding techniques from a first principles perspective, by selecting and connecting predefined blocks. In this project work we simulate a variety of models including analog, digital and mixed mode designs, and quickly simulate their behavior using the VisSim/ Comm software and graphical programming.

CHAPTER 1

GENERAL INTRODUCTION

COMMUNICATION SYSTEMS

AND “VisSim”

1. COMMUNICATIONS SYSTEMS AND VisSim

A communications system is a collection of individual communications networks, transmission systems, relay stations, tributary stations, and data terminal equipment (DTE) usually capable of interconnection and interoperation to form an integrated whole.The components of a communications system serve a common purpose, are technically compatible, use common procedures, respond to controls, and operate in unison.As such any communications system consists of subsystems which work together to achieve a common link ,through achieving its own functionality.

A typical communication link includes, at a minimum, three key elements: a transmitter, a communication medium (or channel), and a receiver. The ability to simulate all three of these elements is required in order to successfully model any end-to-end communication system. The transmitter and receiver elements can in turn be further subdivided into sub-systems.These include a data source (analog or digital), an optional data encoder, a modulator, a demodulator, an optional data decoder, and a signal sink. To understand the process of such a communication we need to visualize or simulate such a link ,so as to have a better understanding of the process involved.We have used a simulation software “VisSim” ,or Visual Simulator ,that allows us to use a graphical approach to simulation and modeling.

With graphical programming, the diagram *is* the source code, depicted as an arrangement of nodes connected by wires. Each piece of data flows through the wires, to be consumed by nodes that transform the data mathematically or perform some action such as I/O.

The concept of a dataflow diagram (which, unlike a flowchart, shows the motion of data rather than the motion of logic) is nothing new. In fact, even the idea of letting a dataflow diagram be the sole input to a compiler or interpreter has been put into practice for years. A number of graphical programming tools are available today, each tailored to a particular industry.The tool in use, ”VisSim” ,has a special communication module that allows us to create accurate simulation environment of the communication system involved. It is a software program for modeling end-to-end communication systems at the signal or physical level. Execution is determined by the structure of a graphical block diagram on which the programmer connects

different function-nodes by drawing wires. These wires propagate signals and any subsystem can execute as soon as all its input data become available. Since this might be the case for multiple subsystems simultaneously, we are capable of parallel execution.

Simulation of communication channel and evaluating the performance requires accurate reconstruction of the channel and its subsystems. We have used graphical programming to create a visual simulation of communication systems using VisSim.The essential advantages in visual simulation are due to ease of modeling of:

Transmitter and Receiver Models

Communication system design can be divided into two categories: transmitter design and receiver design. VisSim/ Comm lets us build build both transmitter and receiver models, from a first principles perspective, by simply selecting and connecting predefined blocks. We simulate a variety of models including analog, digital and mixed mode designs, and quickly simulate their behavior. The VisSim/Comm block set provides a variety of modulators and demodulators, including standard analog, PSK, QAM and differential formats. .

Channel Models

VisSim/Comm includes a variety of predefined channel models supporting both fixed and mobile service scenarios. Included are fading, multipath, bandlimited, and gaussian noise models. Further all VisSim/Comm blocks, can modify model parameters to suit their specific needs.

Filter and Equalizer Design

VisSim/Comm supports a wide range of customizable filters, including FIR, IIR, gaussian, raised cosine and root raised cosine filters. Additional blocks, such as the complex FFT block, make it easy to view gain and phase responses of any filter. Furthermore, for designs that require adaptive filters, fractionally-spaced LMS equalizer blocks are included.

Predicting System Performance

Once designed, a transmitter or receiver model can be simulated to determine its performance under a variety of operating conditions. VisSim/Comm highly interactive interface makes it easy to perform ‘what if’ simulations and carry out performance trade-offs. For example, in analog modulation we can keep amplitude modulation and frequency modulation side by side and evaluate their envelope shapes , simultaneously.

However like any other approach to coding, graphical programming is not a panacea that meets all software needs. Besides the obviously more expensive hardware required to create and view dataflow diagrams, there are far fewer cheap or free software tools available. Despite their ability to be compiled, graphical programs still rely on hefty runtime libraries that may slow performance. Additionally, the dataflow model proves unsettling and unproductive for some coders and inappropriate for some jobs.

Thus the graphical programming approach used in Vissim eases the simulation by creating a platform for visual implementation of such communication systems .The visual presentation of their ideas is direct and refreshing. The ability to prototype rapidly and call on a wide range of industry-specific libraries leads to productivity increase for certain tasks.

CHAPTER 2

ANALOG MODULATION

AMPLITUDE MODULATION

FREQUENCY MODULATION

COMBINATION OF AM AND FM

2.ANALOG MODULATION

2.1Amplitude modulation (AM)

Amplitude modulation is a technique used in electronic communication, most commonly for transmitting information via a radio carrier wave. AM works by varying the strength of the transmitted signal in relation to the information being sent. For example, changes in the signal strength can be used to reflect the sounds to be reproduced by a speaker, or to specify the light intensity of television pixels. In its basic form, amplitude modulation produces a signal with power concentrated at the carrier frequency and in two adjacent sidebands. Each sideband is equal in bandwidth to that of the modulating signal and is a mirror image of the other. Amplitude modulation that results in two sidebands and a carrier is often called *double sideband amplitude modulation* (DSB-AM). Amplitude modulation is inefficient in terms of power usage and much of it is wasted. At least two-thirds of the power is concentrated in the carrier signal, which carries no useful information (beyond the fact that a signal is present); the remaining power is split between two identical sidebands, though only one of these is needed since they contain identical information.

Carrier Wave:

$$c(t) = C \cdot \sin(\omega_c t + \phi_c),$$

Waveform to be transmitted:

$$m(t) = M \cdot \cos(\omega_m t + \phi).$$

Hence ,the net amplitude modulated wave is of the form

$$= [A + M \cdot \cos(\omega_m t + \phi)] \cdot \sin(\omega_c t).$$

Modulation Index : As with other modulation indices, in AM, this quantity, also called *modulation depth*, indicates by how much the modulated variable varies around its 'original' level. For AM, it relates to the variations in the carrier amplitude and is defined as:

$$h = \frac{\text{peak value of } m(t)}{A} = \frac{M}{A},$$

Now we will simulate the amplitude modulation using VisSim. The main parameters that are needed here are:

1. Input Signal(which is a combination of many sine waves)

2. AM Modulator (which modulates the input signal)

3. Complex to Real (converts the complex quantity into real & imaginary part)

The Input Signal consists of a no. of sine waves where we can change the amplitudes of the sine waves by the following method:

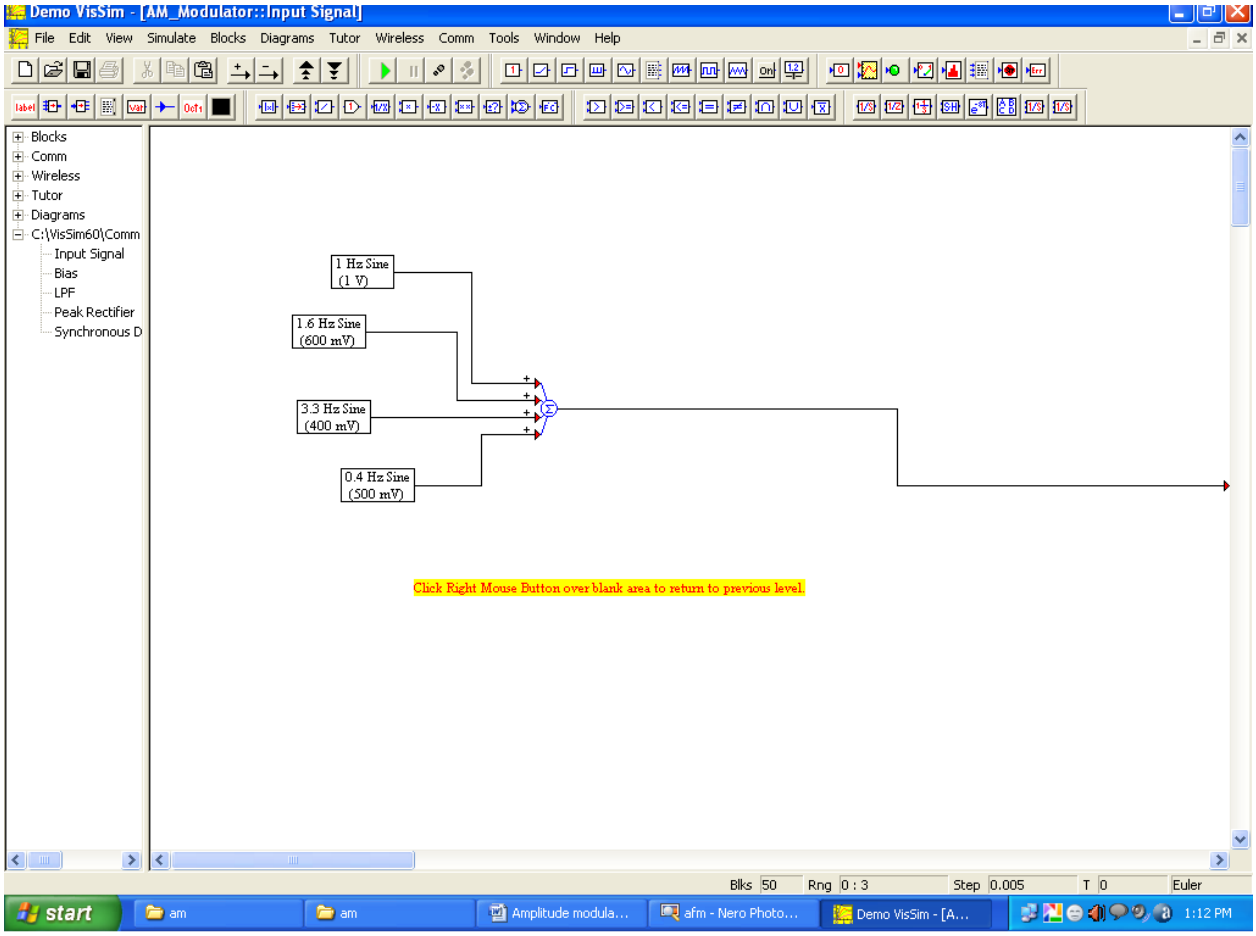


Fig 2.1

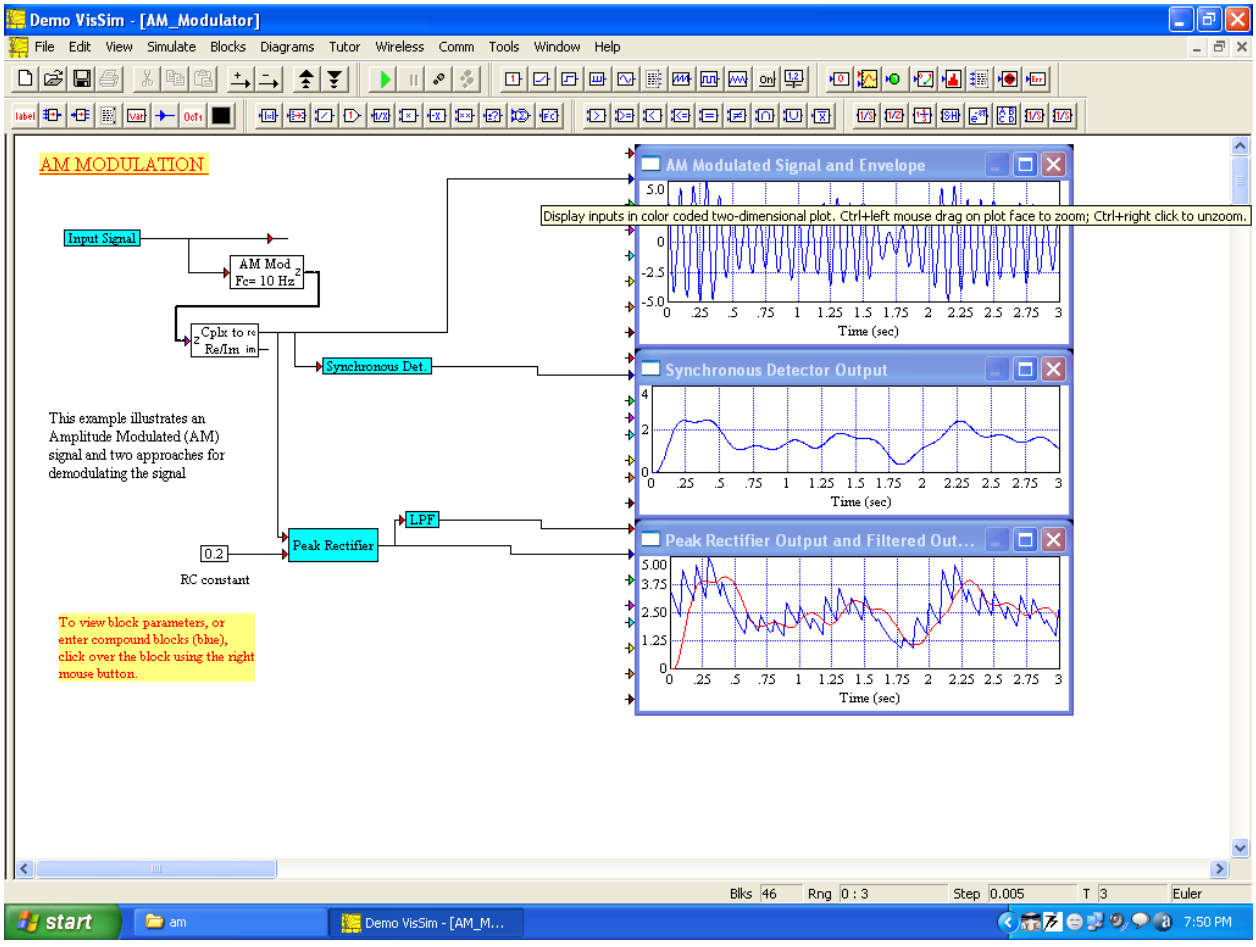
The AM Modulator has the following parameters (that can also be changed according to our wish):

1. Carrier frequency
2. Amplitude
3. Initial Phase
4. Modulation Factor

Here there are basically two types of Phase Output Modes:

1. Wrapped[0,2pi]
2. Unwrapped

Fig-2.2 AM modulation and demodulation



The outputs that we get here are basically:

1. AM Modulated Signal and Envelope
2. Synchronous detector Output
3. Peak Rectifier Output and Filtered Output

2.2 FREQUENCY MODULATION

In telecommunications, **frequency modulation** (**FM**) conveys information over a carrier wave by varying its frequency (contrast this with amplitude modulation, in which the amplitude of the carrier is varied while its frequency remains constant). In analog applications, the instantaneous frequency of the carrier is directly proportional to the instantaneous value of the input signal. Digital data can be sent by shifting the carrier's frequency among a set of discrete values, a technique known as frequency-shift keying.

Suppose the baseband data signal to be transmitted is

$$x_m(t)$$

and is restricted in amplitude to be

$$|x_m(t)| \leq 1,$$

and the sinusoidal carrier is

$$x_c(t) = A\cos(2\pi f_c t)$$

where f_c is the carrier's base frequency and A is an arbitrary amplitude. The modulator combines the carrier with the baseband data signal to get the transmitted signal,

$$y(t) = A\cos\left(2\pi \int_0^t f(\tau)\,d\tau\right) = A\cos\left(2\pi \int_0^t \left[f_c + f_\Delta x_m(\tau)\right]\,d\tau\right)$$

where $f(\tau) = f_c + f_\Delta x_m(\tau).$

Modulation Index :As with other modulation indices, in FM this quantity indicates by how much the modulated variable varies around its unmodulated level. For FM, it relates to the variations in the frequency of the carrier signal where f_m is the highest modulating frequency of $x_m(t)$. If , the modulation is called *narrowband FM*, and its bandwidth is approximately $2f_m$. If , the

modulation is called *wideband FM* and its bandwidth is approximately $2f_\Delta$. While wideband FM uses more bandwidth, it can improve signal-to-noise ratio significantly.

Now we will simulate the amplitude modulation using VisSim. The main parameters that are needed here are:

1. Source or the Input Signal(which is a combination of many sine waves)
2. FM Modulator (which modulates the input signal)
3. FM Demodulator
4. Complex to Real (converts the complex quantity into real & imaginary part)

Here also we can change the values of the parameters according to our requirement.

FM modulation

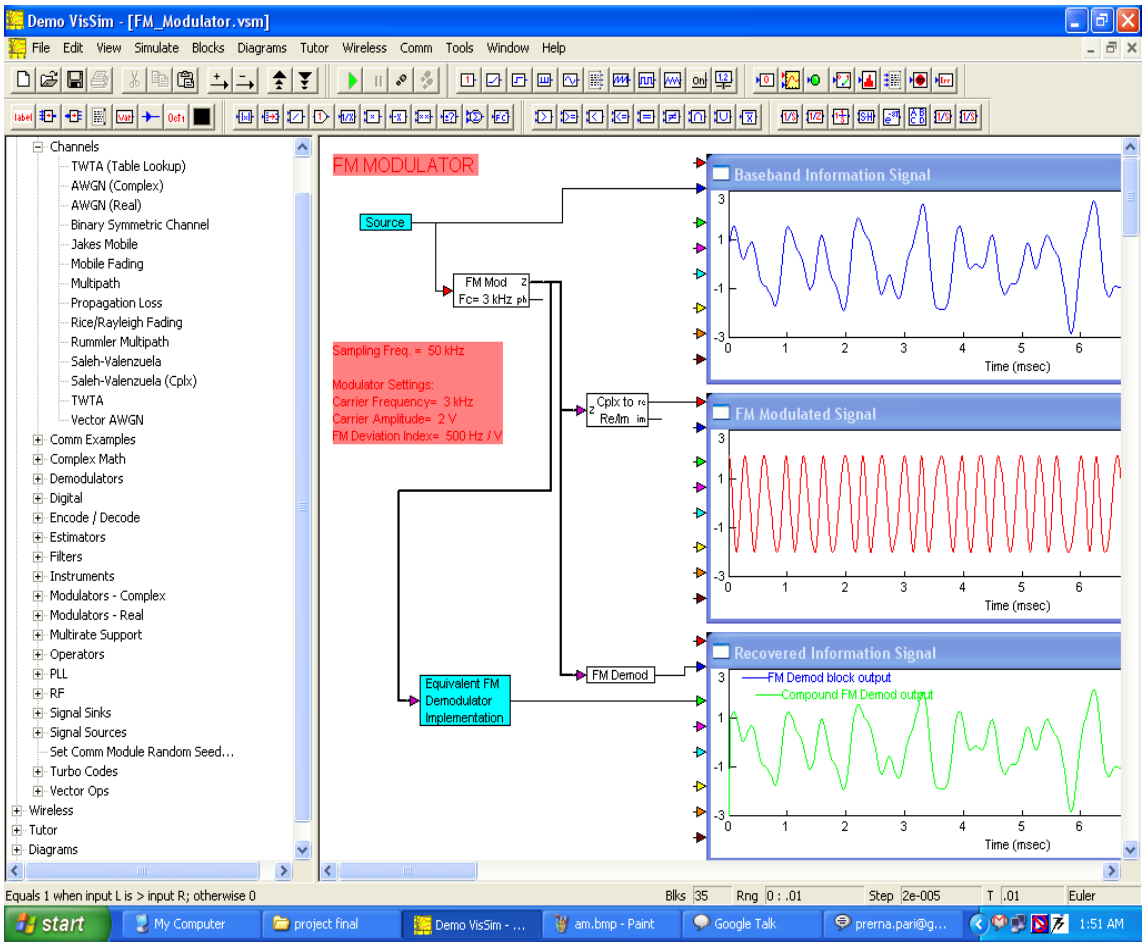


Fig 2.3

The outputs that we get here are basically:

1. Baseband Information Signal
2. FM Modulated Signal
3. Recovered Information Signal

2.3 Combination of AM and FM:

The major advantage that we can have with VisSim is that we can plot the Input Signal, The AM Modulated Signal and the FM Modulated signal simultaneously which helps us in comparing the two outputs with a single source

Here the basic components that are involved:

1. Input Signal(which is a combination of many sine waves)
2. AM Modulator (which modulates the input signal w.r.t. Amplitude)
3. FM Modulator(which modulates the input signal w.r.t. Frequency)
4. Complex to Real (converts the complex quantity into real & imaginary part)

The outputs that we achieve here are:

1. AM Modulated Signal and Envelope
2. FM Modulated Signal

AM Vs FM

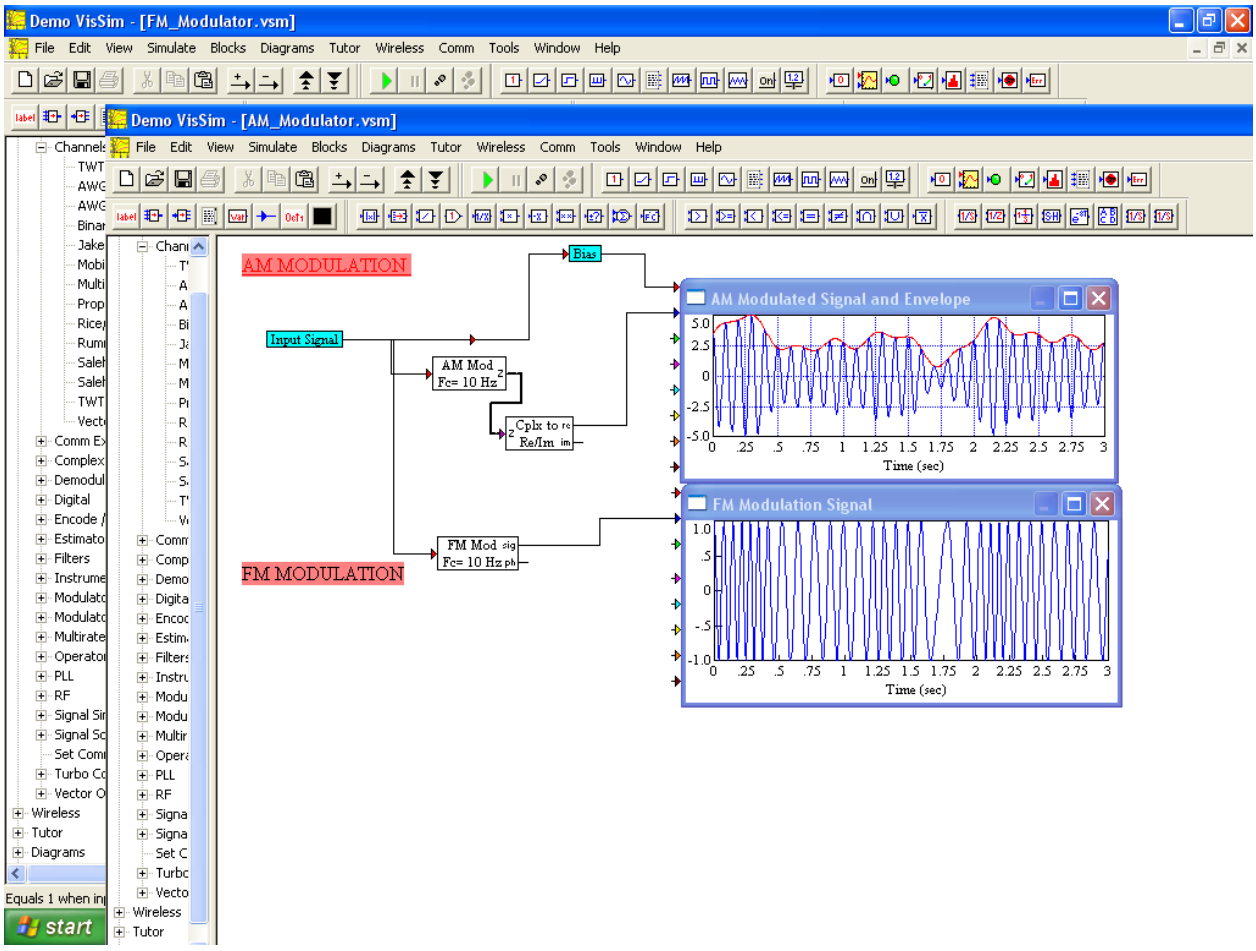


Fig 2.4

CHAPTER 3

DIGITAL CIRCUITS

COUNTERS

MULTIPLEXERS

FLIP FLOPS

3. DIGITAL CIRCUITS

3.1COUNTERS

In digital logic and computing, a counter is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal. In practice, there are two main types of counters:

- up counters which increase (increment) in value
- down counters which decrease (decrement) in value

A few major designs of counters are

- Asynchronous (ripple) counters
- Synchronous counters
- Decade counters
- Up-Down counters
- Ring counters

3.1.1 Asynchronous (ripple) counters

The simplest counter circuit is a single D-type flip flop, with its D (data) input fed from its own inverted output. This circuit can store one bit, and hence can count from zero to one before it overflows (starts over from 0). This counter will increment once for every clock cycle and takes two clock cycles to overflow, so every cycle it will alternate between a transition from 0 to 1 and a transition from 1 to 0. Notice that this creates a new clock with a 50% duty cycle at exactly half the frequency of the input clock. If this output is then used as the clock signal for a similarly arranged D flip flop (remembering to invert the output to the input), you will get another 1 bit counter that counts half as fast. Putting them together yields a two bit counter:

cycle Q1 Q0 (Q1:Q0)dec

0 0 0 0

1 0 1 1

3.1.2 Synchronous counters

Where a stable count value is important across several bits, which is the case in most counter systems, synchronous counters are used. These also use flip-flops, either the D-type or the more complex J-K type, but here, each stage is clocked simultaneously by a common clock signal. Logic gates between each stage of the circuit control data flow from stage to stage so that the desired count behaviour is realised. Synchronous counters can be designed to count up or down, or both according to a direction input, and may be presettable via a set of parallel "jam" inputs. Most types of hardware-based counter are of this type.

3.1.3 Decade counters

Decade counters are a kind of counter that counts in tens rather than having a binary representation. Each output will go high in turn, starting over after ten outputs have occurred. This type of circuit finds applications in multiplexers and demultiplexers, or wherever a scanning type of behaviour is useful. Similar counters with different numbers of outputs are also common.

3.1.4 Up-Down Counters

It is a combination of up counter and down counter, counting in straight binary sequence. There is an up-down selector. If this value is kept high, counter increments binary value and if the value is low, then counter starts decrementing the count. The Down counters are made by using the complemented output to act as the clock for the next flip-flop in the case of Asynchronous counters.

3.1.5 Ring Counters

A ring counter is a counter that counts up and when it reaches the last number that is designed to count up to, it will reset itself back to the first number. For example, a ring counter that is designed using 3 JK Flip Flops will count starting from 001 to 010 to 100 and back to 001. It will repeat itself in a 'Ring' shape and thus the name Ring Counter is given.

Here we will now simulate the counter with the help of VisSim. The example that we take into consideration is the Binary Counter. In a Binary Counter each bit represent either ‘0’ or ‘1’. If it

is a 4 bit Binary Counter then it can calculate upto 15 and as soon as it counts 15 the counters again resets to '0'.Like in a decade counter we can count from 0-9. The basic components that we need here are:

- Input data stream
- 4-bit counter(which can counts up to 15)

The output that we will get here are the:

- Counter output
- Carry Flag

Binary counter

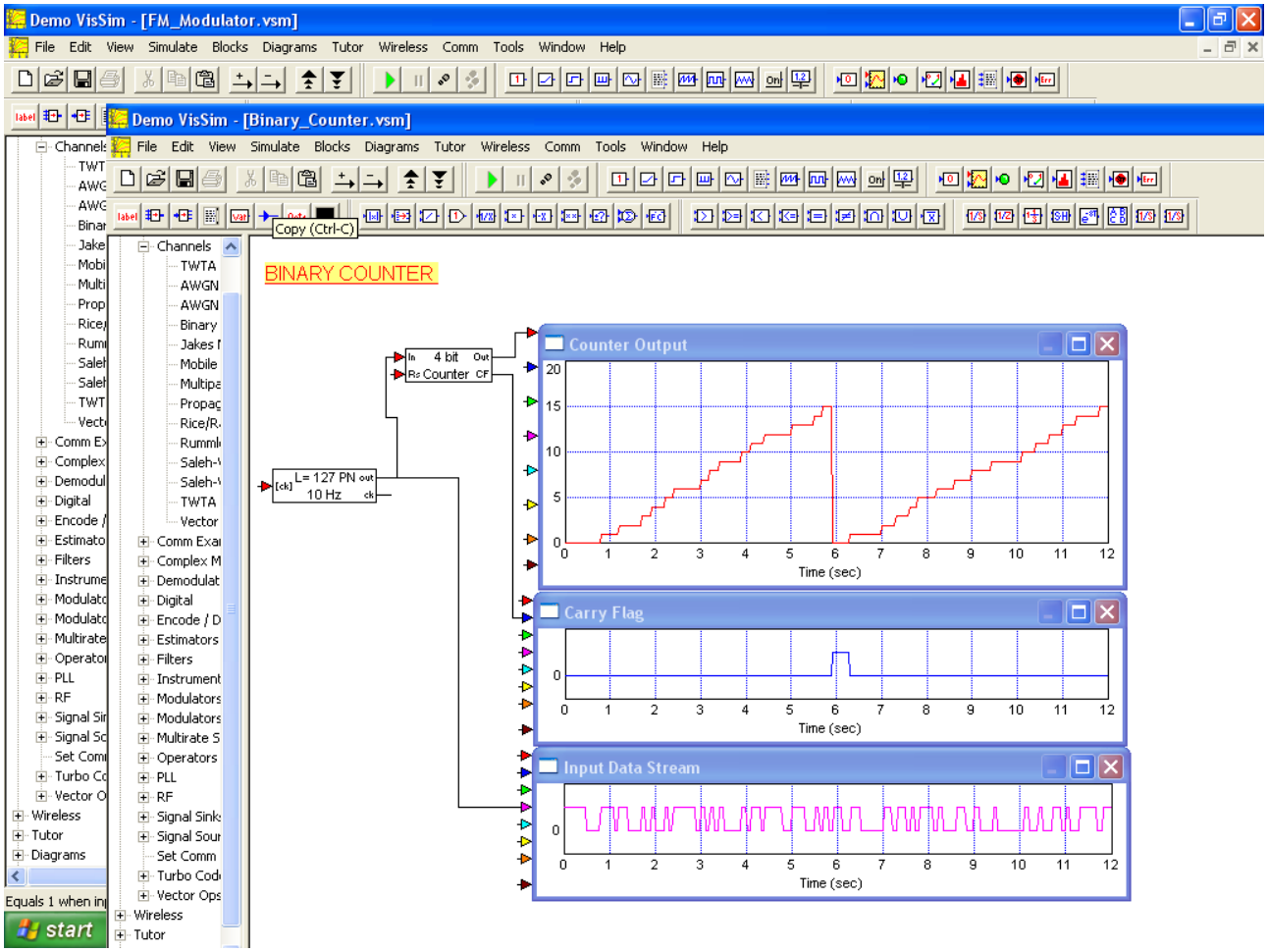


Fig 3.1

3.2 MULTIPLEXER

In electronics, a multiplexer or mux (occasionally the term muldex is also found, for a combination multiplexer-demultiplexer) is a device that performs multiplexing; it selects one of many analog or digital input signals and outputs that into a single line. An electronic multiplexer makes it possible for several signals to share one expensive device or other resource, for example one A/D converter or one communication line, instead of having one device per input signal.

In electronics, a demultiplexer (or demux) is a device taking a single input signal and selecting one of many data-output-lines, which is connected to the single input. A multiplexer is often used with a complementary demultiplexer on the receiving end. An electronic multiplexer can be considered as a multiple-input, single-output switch, and a demultiplexer as a single-input, multiple-output switch.

Multiplexer

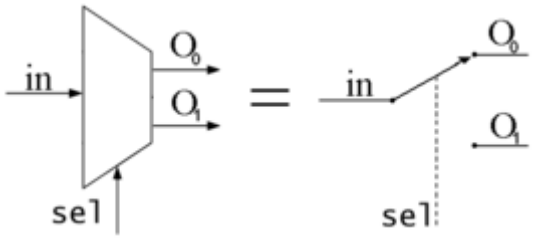


Fig 3.2

Demultiplexer

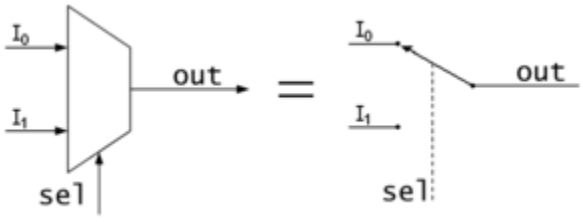


Fig 3.3

Multiplexer simulation

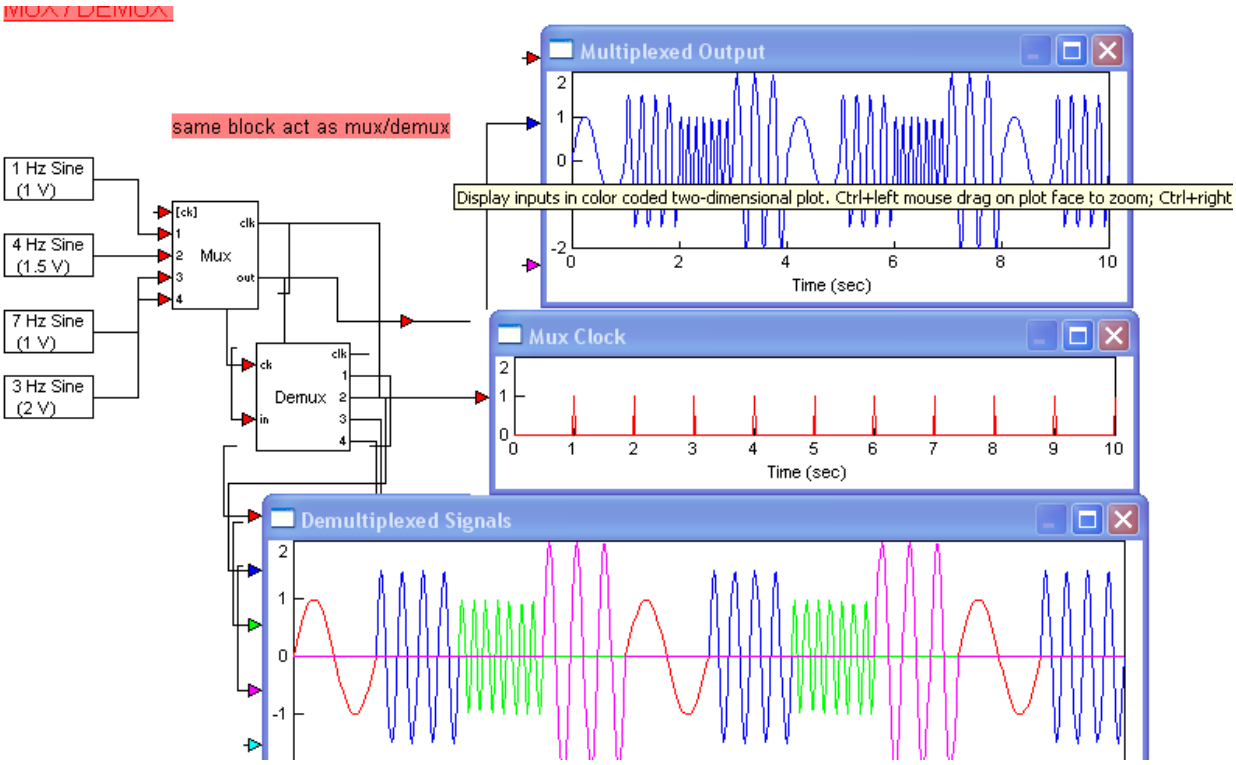


Fig 3.4

3.3 FLIP FLOPS

In digital circuits, a flip-flop is a kind of bistable multivibrator, an electronic circuit which has two stable states and thereby is capable of serving as one bit of memory. Today, the term flip-flop has come to generally denote non-transparent (clocked or edge-triggered) devices, while the simpler transparent ones are often referred to as latches.

A flip-flop is controlled by (usually) one or two control signals and/or a gate or clock signal. The output often includes the complement as well as the normal output. As flip-flops are implemented electronically, they require power and ground connections. Flip-flops can be either simple (transparent) or clocked. Simple flip-flops can be built by two cross-coupled inverting elements – transistors, or NAND, or NOR-gates – perhaps augmented by some enable/disable (gating) mechanism. Clocked devices are specially designed for synchronous (time-discrete) systems and therefore one such device ignores its inputs except **at** the transition of a dedicated clock signal (known as clocking, pulsing, or strobing). This causes the flip-flop to either change or retain its output signal based upon the values of the input signals at the transition. Some flip-flops change output on the rising edge of the clock, others on the falling edge.

3.3.1 JK Flip Flop:

The **JK** flip-flop augments the behavior of the SR flip-flop by interpreting the S = R = 1 condition as a "flip" or toggle command. Specifically, the combination J = 1, K = 0 is a command to set the flip-flop; the combination J = 0, K = 1 is a command to reset the flip-flop; and the combination J = K = 1 is a command to toggle the flip-flop, i.e., change its output to the logical complement of its current value. Setting J = K = 0 does NOT result in a D flip-flop, but rather, will hold the current state. To synthesize a D flip-flop, simply set K equal to the complement of J. The JK flip-flop is therefore a universal flip-flop, because it can be configured to work as an SR flip-flop, a D flip-flop, or a T flip-flop.

JK flip flop

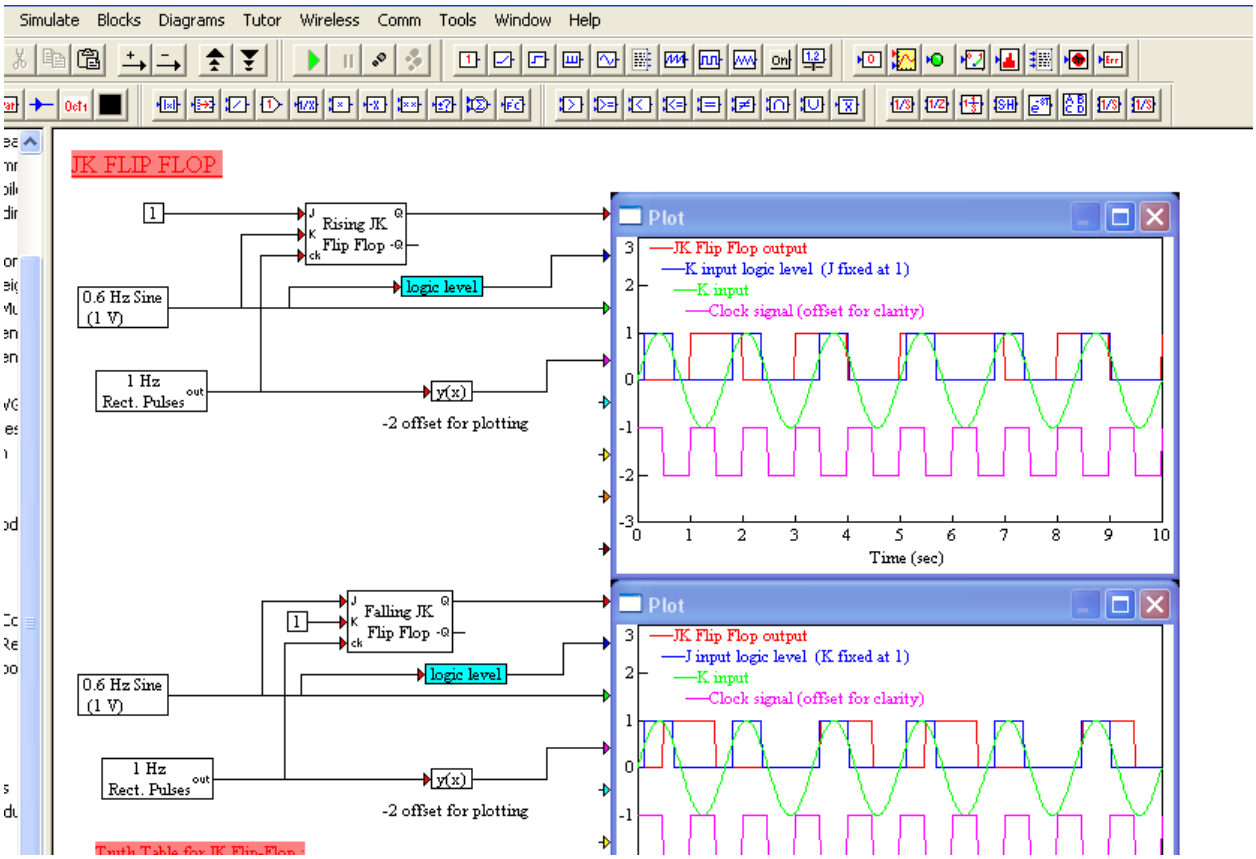


Fig 3.5

3.3.2 D flip-flop

The Q output always takes on the state of the D input at the moment of a rising clock edge, and never at any other time. It is called the **D** flip-flop for this reason, since the output takes the value of the **D** input or Data input, and Delays it by one clock count. The D flip-flop can be interpreted as a primitive memory cell, zero-order hold, or delay line.

Truth table of D flip flop

Clock	**D**	**Q**	**Q_{prev}**
Rising edge	0	0	X
Rising edge	1	1	X

Table 3.1

D flip flop

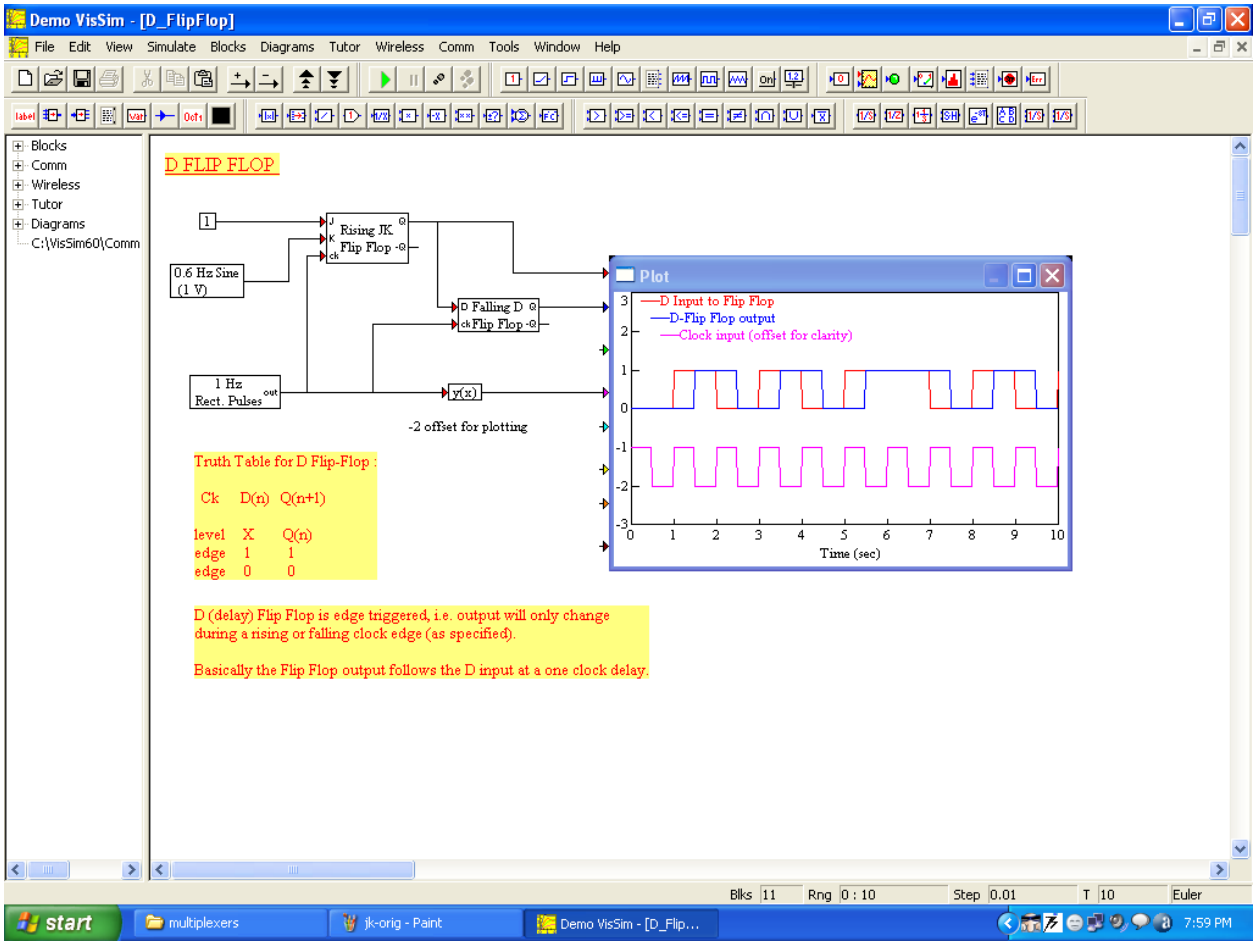


Fig 3.6

CHAPTER 4

FILTERS AND EQUALIZERS

EQUALIZERS

FILTERS

4. FILTERS AND EQUALIZERS

4.1 Equalizers

In a communication system, the transmitter sends the information over an RF channel. The channel distorts the transmitted signal befores it reaches the receiver. The receiver ”task” is to figure out what signal was transmitted and turn the received signal in understandable information. Equalization is a technique to improve received signal quality and link performance over a noisy communication channel. Equalization compensates for intersymbol interference created by multipath due to time dispersive channels. An equalizer within a receiver compensates for the average range of channel amplitude and phase characteristics. Equalizers must be adaptive since the channel is unknown and time varying generally. An adaptive equalizer is a filter that adaptively updates its coefficients in order to track a time-varying communication channel. It is frequently used with coherent modulations such as phase shift keying in wireless communications, mitigating the effects of multipath propagation and Doppler spreading. The channel equalizer models the impulse response of the radio channel and based on the estimate removes unwanted phenomena (for example echo) from the signal.

An equalization (EQ) filter, or an equalizer is a filter, usually adjustable, chiefly meant to compensate for the unequal frequency response of some other signal processing circuit or system.An EQ filter typically allows the user to adjust one or more parameters that determine the overall shape of the filter's transfer function. It is generally used to improve the fidelity of sound, to emphasize certain instruments, to remove undesired noises.Equalizers may be designed with peaking filters, shelving filters, bandpass filters, plop filters or high-pass and low-pass filters.

Shown below is the block diagram of a 5 tap adaptive filter that takes in input as well as error to adaptively equalize the channel.Further channel equalization of a QAM link has been simulated using VisSim.

Equalizer

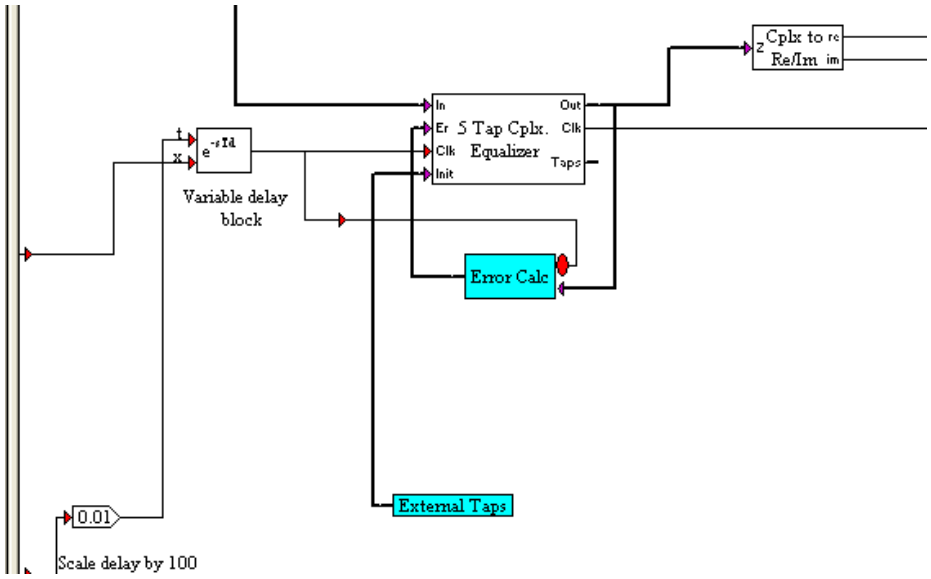
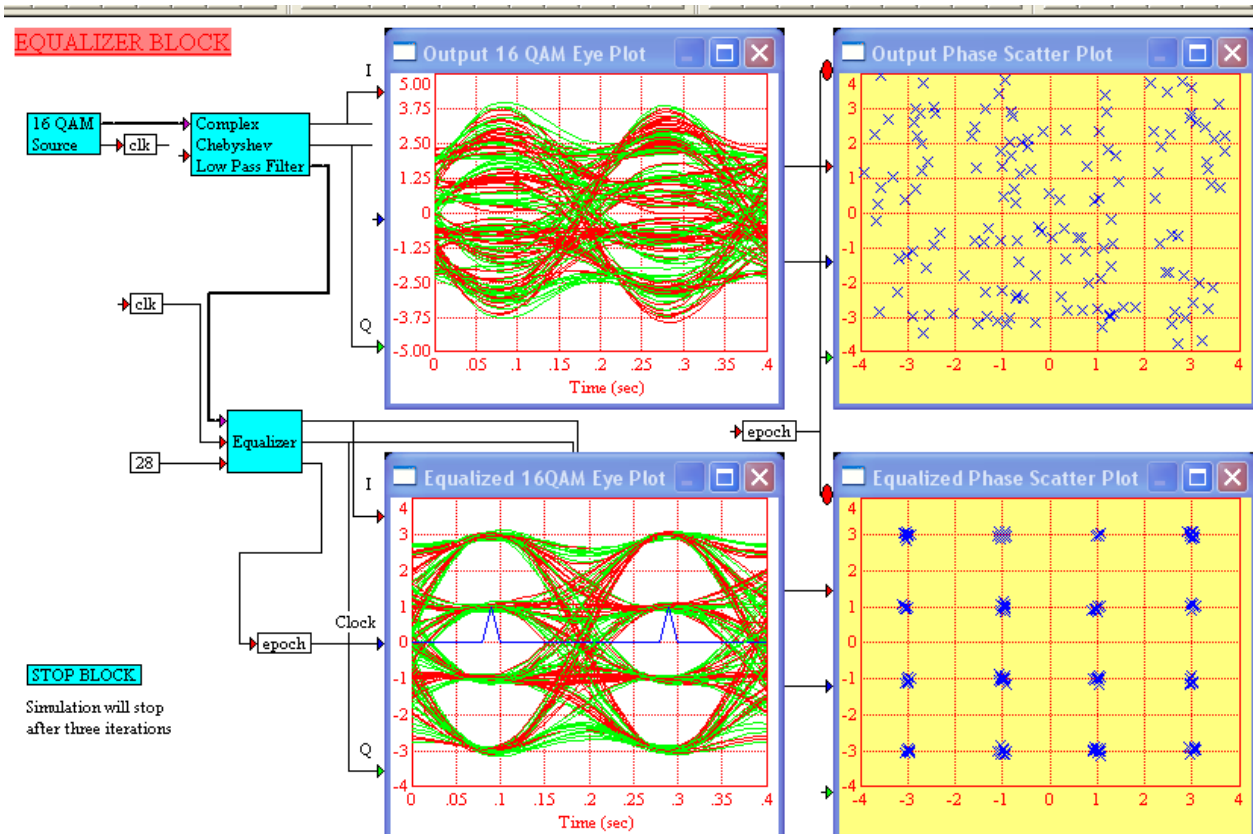


Fig :4.1

Fig-4.2Equalization of channel distortions



4.2 FILTERS

In electronics, a digital filter is any electronic filter that works by performing digital mathematical operations on an intermediate form of a signal. This is in contrast to older analog filters which work entirely in the analog realm and must rely on physical networks of electronic components (such as resistors, capacitors, transistors, etc.) to achieve the desired filtering effect.

Digital filters are implemented according two one of two basic principles, according to how they respond to an impulse:

- Infinite impulse response (IIR)
- Finite impulse response (FIR)

4.2.1 Infinite impulse response (IIR)

IIR is a property of signal processing systems. Systems with that property are known as *IIR systems* or when dealing with electronic filter systems as *IIR filters*. They have an impulse response function which is non-zero over an infinite length of time. This is in contrast to finite impulse response filters (FIR) which have fixed-duration impulse responses. The simplest analog IIR filter is an RC filter made up of a single resistor (R) feeding into a node shared with a single capacitor (C). This filter has an exponential impulse response characterized by an RC time constant.IIR filters may be implemented as either analog or digital filters. In digital IIR filters, the output feedback is immediately apparent in the equations defining the output. Note that unlike with FIR filters, in designing IIR filters it is necessary to carefully consider "time zero" case in which the outputs of the filter have not yet been clearly defined.Design of digital IIR filters is heavily dependent on that of their analog counterparts because there are plenty of resources, works and straightforward design methods concerning analog feedback filter design while there are hardly any for digital IIR filters. As a result, mostly, if a digital IIR filter is going to be implemented, first, an analog filter (e.g. Chebyshev filter, Butterworth filter, Elliptic filter) is designed and then it is converted to digital by applying discretization techniques such as Bilinear transform or Impulse invariance.

An IIR filter has the difference equation which defines how the input signal is related to the output signal:

where:

- P is the feedforward filter order
- b_i are the feedforward filter coefficients
- Q is the feedback filter order
- a_i are the feedback filter coefficients
- $x[n]$ is the input signal
- $y[n]$ is the output signal.

A more condensed form of the difference equation is:

$$y[n] = \sum_{i=0}^{P} b_i x[n-i] - \sum_{j=1}^{Q} a_j y[n-j]$$

which, when rearranged, becomes:

$$\sum_{j=0}^{Q} a_j y[n-j] = \sum_{i=0}^{P} b_i x[n-i]$$

if we let $a_0 = 1$.

To find the transfer function of the filter, we first take the Z-transform of each side of the above equation, where we use the time-shift property to obtain:

$$\sum_{j=0}^{Q} a_j z^{-j} Y(z) = \sum_{i=0}^{P} b_i z^{-i} X(z)$$

We define the transfer function to be:

$$H(z) = \frac{Y(z)}{X(z)}$$
$$= \frac{\sum_{i=0}^{P} b_i z^{-i}}{\sum_{j=0}^{Q} a_j z^{-j}}$$

Simple IIR filter

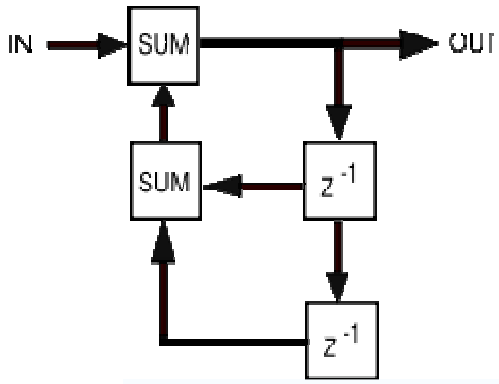


Fig 4.3

4.2.2 Finite impulse response (FIR)

FIR filter is a type of a digital filter. The impulse response, the filter's response to a input, is 'finite' because it settles to zero in a finite number of sample intervals. This is in contrast to infinite impulse response filters which have internal feedback and may continue to respond indefinitely. An Nth order FIR filter has a response to an impulse that is N+1 samples in duration.

An FIR filter can be understood be first stating the difference equation which defines how the input signal is related to the output signal

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N]$$

where $x[n]$ is the input signal, $y[n]$ is the output signal and b_i are the filter coefficients. N is known as the *filter order*; an N^{th}-order filter has $(N + 1)$ terms on the right-hand side; these are commonly referred to as *taps*.

The previous equation can also be expressed as a convolution of filter coefficients and the input signal.

$$y[n] = \sum_{i=0}^{N} b_i x[n-i].$$

To find the impulse response we set

$$x[n] = \delta[n]$$

where δ[*n*] is the Kronecker delta impulse. The impulse response for an FIR filter is the set of coefficients b_n, as follows

$$\begin{aligned} h[n] &= \sum_{i=0}^{N} b_i \delta[n-i] \\ &= b_n. \end{aligned}$$

for $n = 0$ to N.

The Z-transform of the impulse response yields the transfer function of the FIR filter

$$\begin{aligned} H(z) &= Z\{h[n]\} \\ &= \sum_{n=-\infty}^{\infty} h[n] z^{-n} \\ &= \sum_{n=0}^{N} b_n \, z^{-n}. \end{aligned}$$

Simple FIR filter

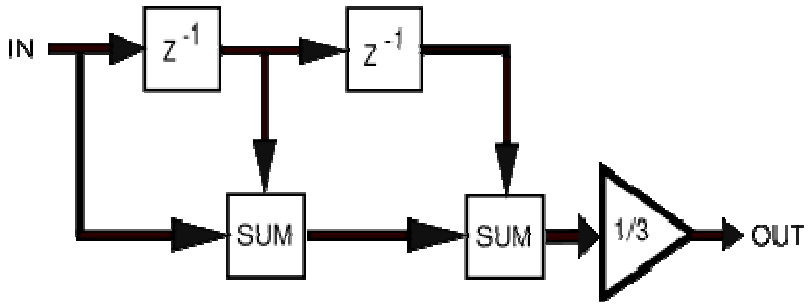


Fig 4.4

FIR filters are clearly bounded-input bounded-output (BIBO) stable, since the output is a sum of a finite number of finite multiples of the input values, so can be no greater than $\sum |b_i|$ times the largest value appearing in the input.

An FIR filter has a number of useful properties which sometimes make it preferable to an infinite impulse response filter. FIR filters:

- Are inherently stable. This is due to the fact that all the poles are located at the origin and thus are located within the unit circle.
- Require no feedback. This means that any rounding errors are not compounded by summed iterations. The same relative error occurs in each calculation.
- They can be designed to be linear phase, which means the phase change is proportional to the frequency. This is usually desired for phase-sensitive applications, for example crossover filters, and mastering, where transparent filtering is adequate.

Simulation of a FIR Filter using the help of Vissim is shown below. The basic blocks that we take into consideration are

- A complex FFT Block
- A band pass Filter(FIR)

The fft block converts data from time domain to frequency domain. The fft block computes an *n*-sample FFT at every simulation time step, where *n* is the length of the input vector .If the input to the fft block is not an integral power of 2, automatic zero padding is performed to make the input vector size an integral power of 2. This is a standard procedure in FFT computation. The output of the fft block is Fourier coefficients.

The FIR Band pass Filter block implements a Finite Impulse Response (FIR) filter. It employs the windowing method for filter design and allows you to implement low pass, high pass, band pass, band stop, raised cosine, root raised cosine, Hilbert and Gaussian filters with your choice of window function. A snapshot of the tunable parameters has been shown below

FIR filter parameters

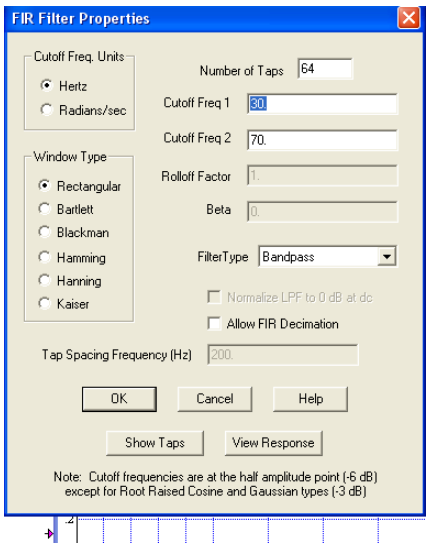


Fig 4.5

FIR filter Simulation

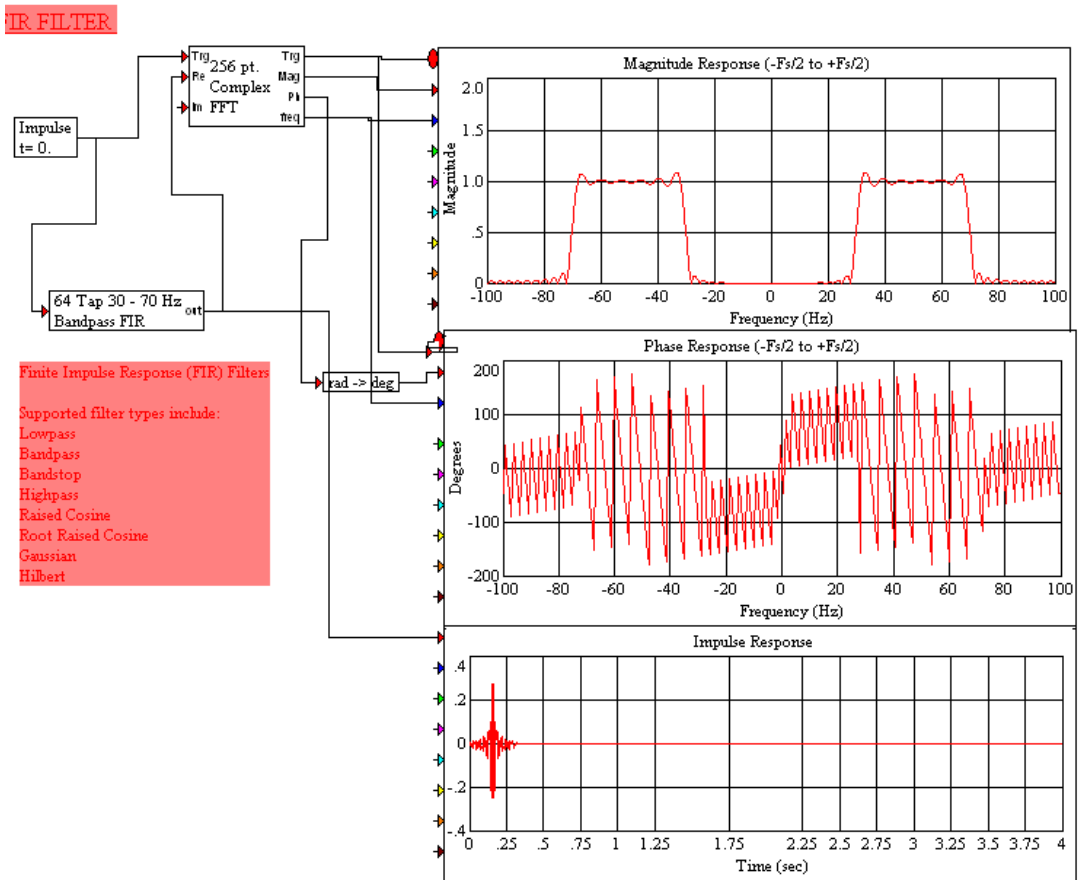


Fig 4.6

The IIR implementation uses analog filter prototypes, including Butterworth and Chebyshev designs. The desired filter is implemented using bilinear transformation to map the *s*-domain analog design to the digital *z*-domain. Shown below are two IIR filters based on the Butterworth and Chebyshev designs.

IIR filter simulation

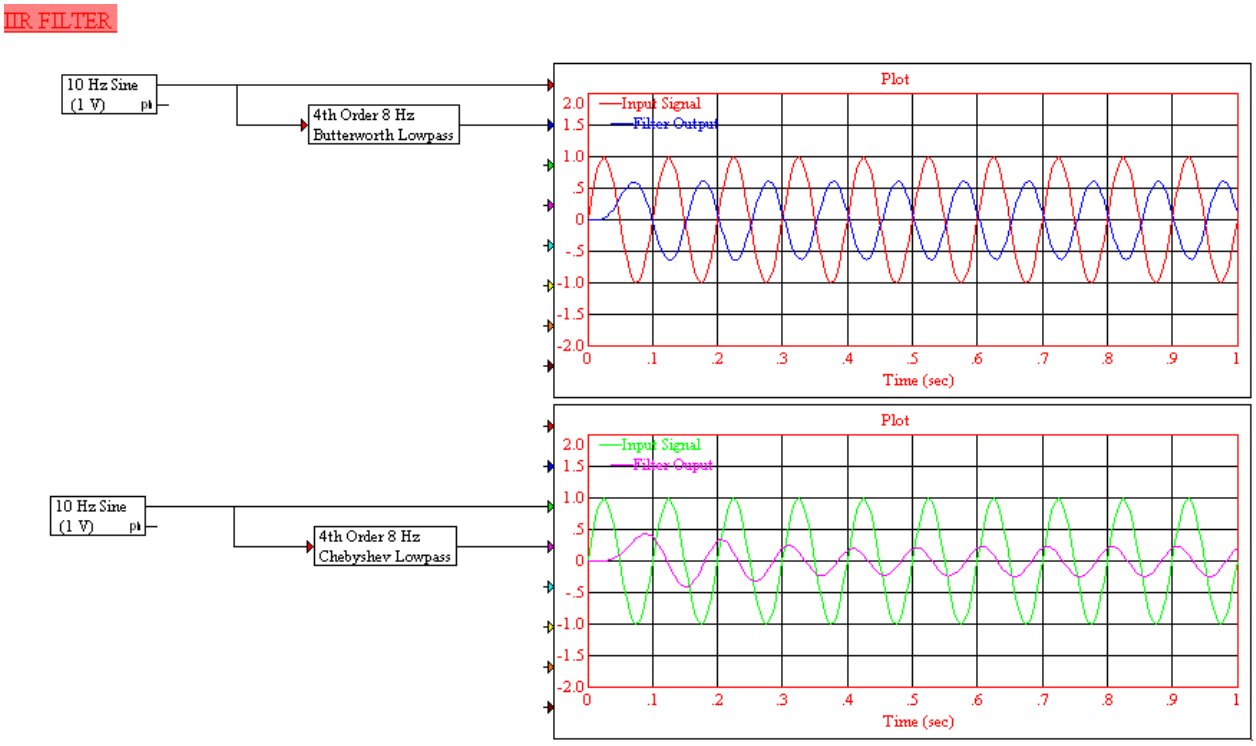


Fig 4.7

CHAPTER 5

COMMUNICATION

CHANNEL SIMULATION

TRANSMISSION TECHNIQUES

TURBO CODES

5 COMMUNICATION

5.1CHANNEL SIMULATION

5.1.1 Multipath propagation

A mobile channel is often characterised by multipath propagation. What really happens is that the presence of reflecting objects and scaterers in the channel creates a constantly changing environment that dissipates the signal energy in amplitude phase and time. These effects result in mutltiple versions of the same transmitted signal that arrive at the receiving antenna ,displaced with respect to one another in time and saptial orientation. The random phases and amplitudes of different multipath components cause fluctuation in signal strength therby inducing small scale fading ,signal distortion and or both. Multipath propagation often lengthens the time required for the baseband portion of the signal to reach the receiver which can cause signal smearing or intersymbol interference. Thus in effect we actually get more than a single path from transmiter to receiver all with different path delays . these waves called multipath waves combine at the receiver antennato give a resultant signalwhich can vary in amplitude and phase ,depending upon the distribution of intensity and relative propagation time of each individual wave.

Below we simulate a multi path channel ,showing the congruence between a single multipath channel and several single channels with different propagation delays. The simulation is done with the help of communication module in Vissim.The essential block here is the Multipath channel block. This block implements a multipath channel, in which multiple time and phase shifted versions of a signal are modeled as arriving simultaneously at a receiver. Multipath channels are commonly used to model the interaction between a direct signal and multiple reflected path signals. The reflected signals affect both the amplitude and phase of the received signal. Block parameters include the number of total paths, and the individual path’s delay, relative gain, and phase rotation. This block takes a complex signal as its input, and outputs a complex signal.

x = Complex input signal [Re, Im]

y = Complex output signal [Re, Im]

Multipath propagation

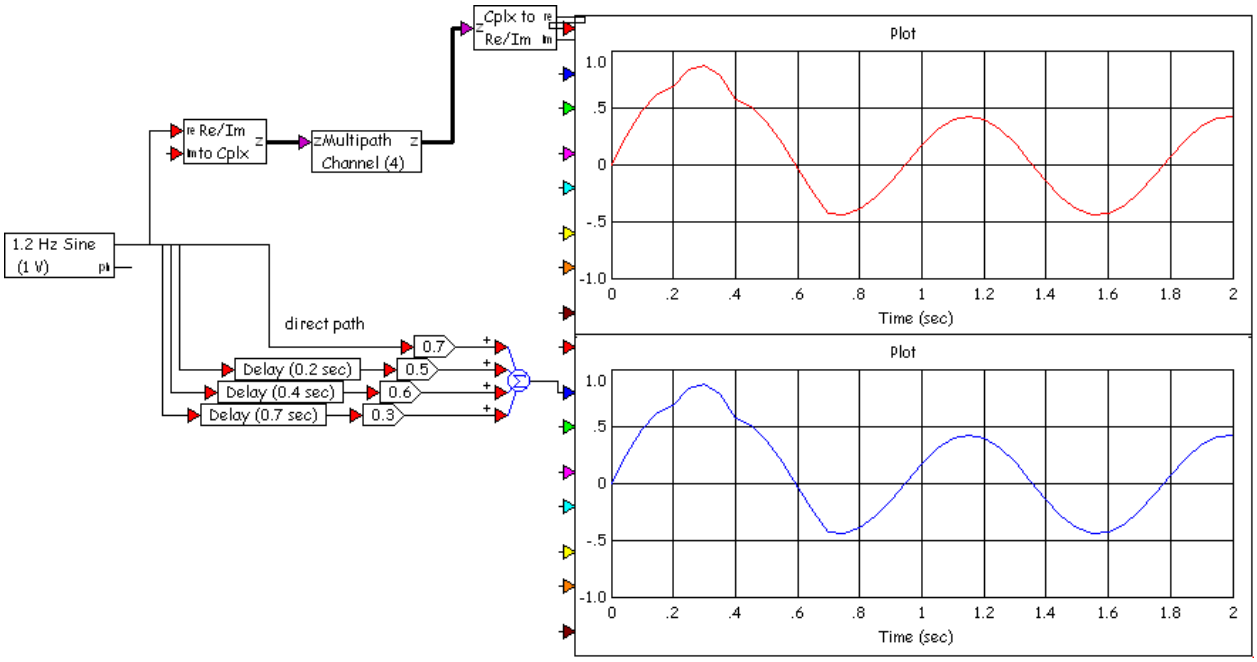


Fig 5.1

5.1.2 Path Loss

Path loss (or path attenuation) is the reduction in power density (attenuation) of an electromagnetic wave as it propagates through space. Path loss is a major component in the analysis and design of the link budget of a telecommunication system.This term is commonly used in wireless communications and signal propagation. Path loss may be due to many effects, such as free-space loss, refraction, diffraction, reflection, aperture-medium coupling loss, and absorption. Path loss is also influenced by terrain contours, environment (urban or rural, vegetation and foliage), propagation medium (dry or moist air), the distance between the transmitter and the receiver, and the height and location of antennas.

Path loss normally includes propagation losses caused by the natural expansion of the radio wave front in free space (which usually takes the shape of an ever-increasing sphere), absorption losses (sometimes called penetration losses), when the signal passes through media not transparent to electromagnetic waves, diffraction losses when part of the radiowave front is obstructed by an

opaque obstacle, and losses caused by other phenomena.The signal radiated by a transmitter may also travel along many and different paths to a receiver simultaneously; this effect is called multipath. Multipath can either increase or decrease received signal strength, depending on whether the individual multipath wavefronts interfere constructively or destructively. The total power of interfering waves in a Rayleigh fading scenario vary quickly as a function of space resulting in fast fades which are very sensitive to receiver position.

In the study of wireless communications, path loss can be represented by the path loss exponent, whose value is normally in the range of 2 to 4 (where 2 is for propagation in free space, 4 is for relatively lossy environments and for the case of full specular reflection from the earth surface -- the so-called flat-earth model). In some environments, such as buildings, stadiums and other indoor environments, the path loss exponent can reach values in the range of 4 to 6. On the other hand, a tunnel may act as a waveguide, resulting in a path loss exponent less than 2.Path loss is usually expressed in dB. In its simplest form, the path loss can be calculated using the formula

$$L = 10\ n\ log_{10}(d) + C$$

where L is the path loss in decibels, n is the path loss exponent, d is the distance between the transmitter and the receiver, usually measured in meters, and C is a constant which accounts for system losses.

A path loss model has been simulated as shown below, which allows us to clearly see the signal strength attenuation with distance.

Propagation loss

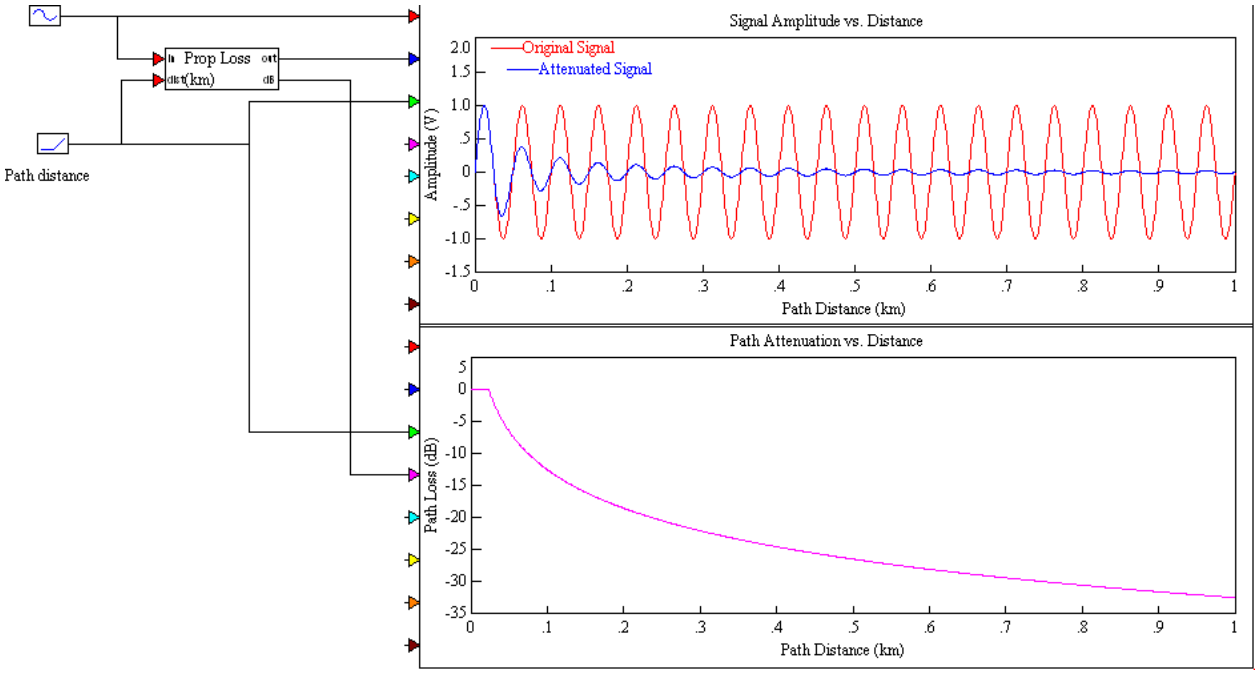


Fig 5.2

5.1.3 Fading

In wireless communications, the presence of reflectors in the environment surrounding a transmitter and receiver create multiple paths that a transmitted signal can traverse. As a result, the receiver sees the superposition of multiple copies of the transmitted signal, each traversing a different path. Each signal copy will experience differences in attenuation, delay and phase shift while travelling from the source to the receiver. This can result in either constructive or destructive interference, amplifying or attenuating the signal power seen at the receiver. Strong destructive interference is frequently referred to as a deep fade and may result in temporary failure of communication due to a severe drop in the channel signal-to-noise ratio.

A common example of multipath fading is the experience of stopping at a traffic light and hearing an FM broadcast degenerate into static, while the signal is re-acquired if the vehicle moves only a fraction of a meter. The loss of the broadcast is caused by the vehicle stopping at a point where the signal experienced severe destructive interference. Cellular phones can also exhibit similar momentary fades.

Fading channel models are often used to model the effects of electromagnetic transmission of information over the air in cellular networks and broadcast communication. Fading channel models are also used in underwater acoustic communications to model the distortion caused by the water. Mathematically, fading is usually modeled as a time-varying random change in the amplitude and phase of the transmitted signal.

5.1.3.1 Slow vs. Fast Fading

The terms slow and fast fading refer to the rate at which the magnitude and phase change imposed by the channel on the signal changes. The coherence time is a measure of the minimum time required for the magnitude change of the channel to become decorrelated from its previous value.

- Slow fading arises when the coherence time of the channel is large relative to the delay constraint of the channel. In this regime, the amplitude and phase change imposed by the channel can be considered roughly constant over the period of use. Slow fading can be caused by events such as shadowing, where a large obstruction such as a hill or large building obscures the main signal path between the transmitter and the receiver. The amplitude change caused by shadowing is often modeled using a log-normal distribution with a standard deviation according to the Log Distance Path Loss Model.

- Fast Fading occurs when the coherence time of the channel is small relative to the delay constraint of the channel. In this regime, the amplitude and phase change imposed by the channel varies considerably over the period of use.

In a fast-fading channel, the transmitter may take advantage of the variations in the channel conditions using time diversity to help increase robustness of the communication to a temporary deep fade. Although a deep fade may temporarily erase some of the information transmitted, use of an error-correcting code coupled with successfully transmitted bits during other time instances (interleaving) can allow for the erased bits to be recovered. In a slow-fading channel, it is not possible to use time diversity because the transmitter sees only a single realization of the channel

within its delay constraint. A deep fade therefore lasts the entire duration of transmission and cannot be mitigated using coding.

The coherence time of the channel is related to a quantity known as the Doppler spread of the channel. When a user (or reflectors in its environment) is moving, the user's velocity causes a shift in the frequency of the signal transmitted along each signal path. This phenomenon is known as the Doppler shift. Signals travelling along different paths can have different Doppler shifts, corresponding to different rates of change in phase. The difference in Doppler shifts between different signal components contributing to a single fading channel tap is known as the Doppler spread. Channels with a large Doppler spread have signal components that are each changing independently in phase over time. Since fading depends on whether signal components add constructively or destructively, such channels have a very short coherence time.

5.1.3.2 Flat vs. Frequency-selective Fading

As the carrier frequency of a signal is varied, the magnitude of the change in amplitude will vary. The coherence bandwidth measures the minimum separation in frequency after which two signals will experience uncorrelated fading.

- In flat fading, the coherence bandwidth of the channel is larger than the bandwidth of the signal. Therefore, all frequency components of the signal will experience the same magnitude of fading.

- In frequency-selective fading, the coherence bandwidth of the channel is smaller than the bandwidth of the signal. Different frequency components of the signal therefore experience decorrelated fading.

In a frequency-selective fading channel, since different frequency components of the signal are affected independently, it is highly unlikely that all parts of the signal will be simultaneously affected by a deep fade. Certain modulation schemes such as OFDM and CDMA are well-suited to employing frequency diversity to provide robustness to fading. OFDM divides the wideband signal into many slowly modulated narrowband subcarriers, each exposed to flat fading rather

than frequency selective fading. This can be combated by means of error coding, simple equalization or adaptive bit loading. Inter-symbol interference is avoided by introducing a guard interval between the symbols. CDMA uses the Rake receiver to deal with each echo separately.

Frequency-selective fading channels are also dispersive, in that the signal energy associated with each symbol is spread out in time. This causes transmitted symbols that are adjacent in time to interfere with each other. Equalizers are often deployed in such channels to compensate for the effects of the intersymbol interference

A simulation of a mobile fading channel has been shown below. Here we derive the signal fading as it passes through a mobile channel and also generate the doppler fading spectrum of the channel.The essential blocks used were

- MOBILE FADING :This block implements a mobile Raleigh fading channel suitable for modeling mobile communications systems. This block is similar to the Jakes Mobile block, but uses a different approach for shaping the spectrum of the fading process. While the Jakes Mobile block approximates a Rayleigh fading process via the summation of multiple complex sinusoids, the Mobile Fading block does so by passing a uniform fading spectrum through an appropriate FIR shaping filter. Block parameters include the Doppler shift frequency and the desired number of taps for the FIR fading filter.
- SPECTRUM : This block outputs the complex power spectrum of the input signal. The spectrum can be continuously updated (once started by the external trigger) or produced at user-defined intervals (again, using the external trigger). Results are viewed using a plot block configured in XY mode with an external trigger. An output trigger line and x-axis output are provided for driving the plot block. Block parameters are FFT window, trigger mode ,spectral output and power spectrum units.

Mobile channel fading

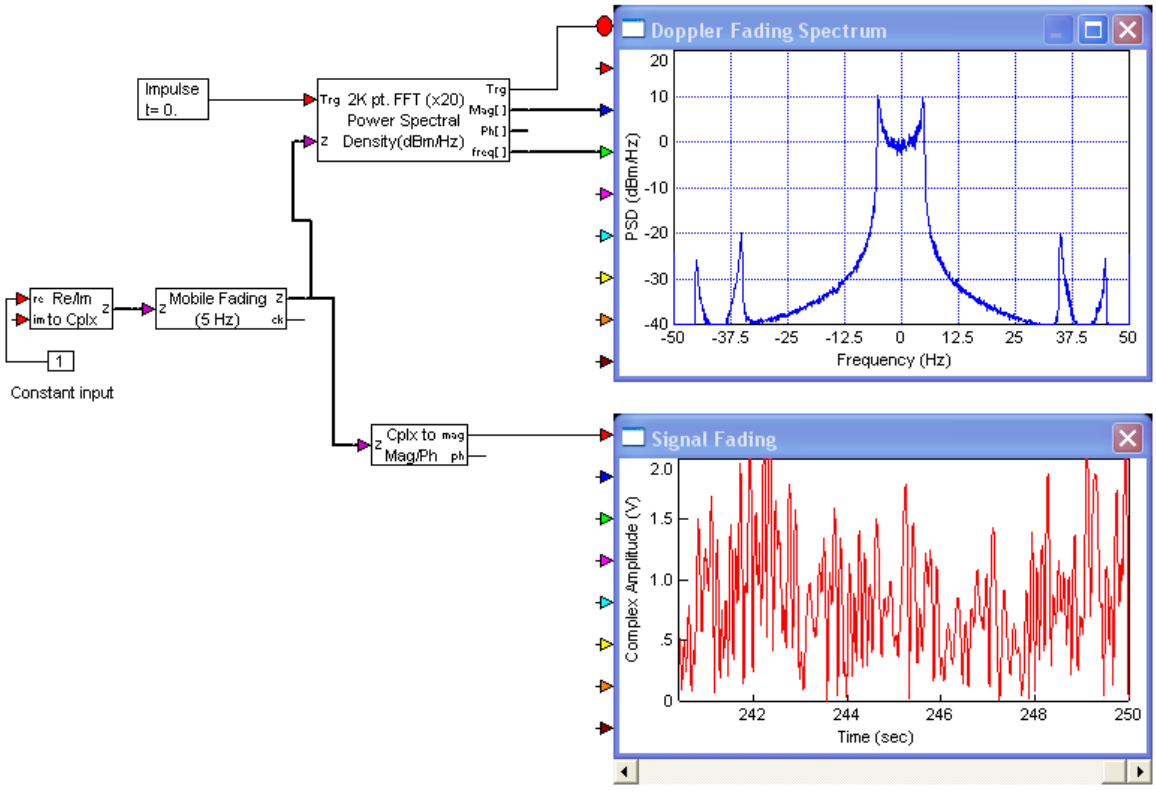


Fig 5.3

5.2 TRANSMISSION TECHNIQUES

5.2.1 Interleaving

Interleaving is used in digital data transmission technology to protect the transmission against burst errors. The

se errors overwrite a lot of bits in a row, so a typical error correction scheme that expects errors to be more uniformly distributed can be overwhelmed. Interleaving is used to help stop this from happening.

Data is often transmitted with error control bits that enable the receiver to correct a certain number of errors that occur during transmission. If a burst error occurs, too many errors can be made in one code word, and that codeword cannot be correctly decoded. To reduce the effect of such burst errors, the bits of a number of codewords are interleaved before being transmitted. This way, a burst error affects only a correctable number of bits in each codeword, and the decoder can decode the codewords correctly.

This method is popular because it is a less complex and cheaper way to handle burst errors than directly increasing the power of the error correction scheme

To understand interleaving we take up an example as below

We apply an error correcting code so that the channel codeword has four bits and one-bit errors can be corrected. The channel codewords are put into a block like this: aaaabbbbccccddddeeeeffffgggg.

Consider transmission without interleaving:

Error-free message: aaaabbbbccccddddeeeeffffgggg
Transmission with a burst error: aaaabbbbccc____deeeeffffgggg

The codeword dddd is altered in three bits, so either it cannot be decoded at all (decoding failure) or it might be decoded into the wrong codeword (false decoding). Which of the two happens depends on the error correcting code applied.

Now, with interleaving:

Error-free code words:	aaaabbbbccccddddeeeeffffgggg
Interleaved:	abcdefgabcdefgabcdefgabcdefg
Transmission with a burst error:	abcdefgabcd____bcdefgabcdefg
Received code words after deinterleaving:	aa_abbbbccccdddde_eef_ffg_gg

In each of the codewords aaaa, eeee, ffff, gggg, only one bit is altered, so our one-bit-error-correcting-code will decode everything correctly

Block interleaver

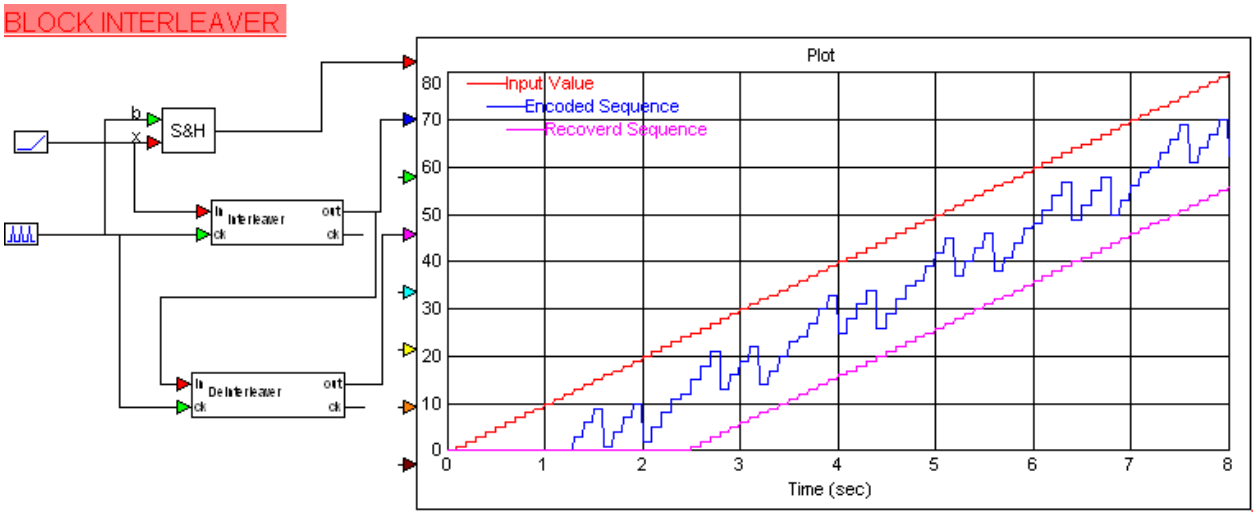


Fig 5.4

5.2.2Encoding and Decoding

5.2.2.1 Gray code

The reflected binary code, also known as Gray code, is a binary numeral system where two successive values differ in only one digit.The reflected binary code was originally designed to prevent spurious output from electromechanical switches. Today, Gray codes are widely used to facilitate error correction in digital communications

Dec	Gray	Binary
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110
7	100	111

Gray encoding and decoding

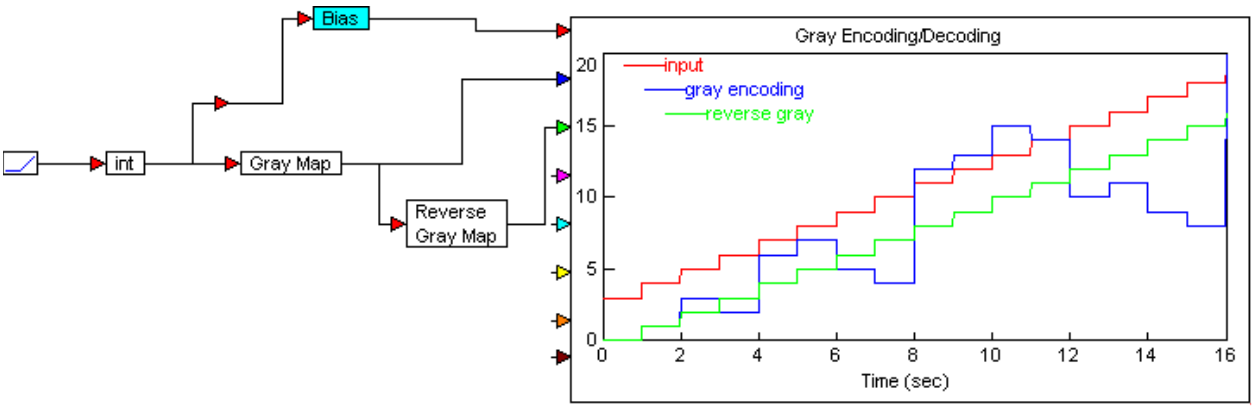


Fig 5.5

5.2.2.2 Convolution codes

Convolutional code is a type of error-correcting code in which (a) each m-bit information symbol (each m-bit string) to be encoded is transformed into an n-bit symbol, where m/n is the code rate ($n \geq m$) and (b) the transformation is a function of the last k information symbols, where k is the constraint length of the code.

To convolutionally encode data, start with k memory registers, each holding 1 input bit. Unless otherwise specified, all memory registers start with a value of 0. The encoder has n modulo-2 adders, and n generator polynomials — one for each adder (see figure below). An input bit m_1 is fed into the leftmost register. Using the generator polynomials and the existing values in the remaining registers, the encoder outputs n bits. Now bit shift all register values to the right (m_1 moves to m_0, m_0 moves to m_{-1}) and wait for the next input bit. If there are no remaining input bits, the encoder continues output until all registers have returned to the zero state. One can see that the input being encoded is included in the output sequence too (look at the output 2). Such codes are referred to as systematic; otherwise the code is called non-systematic

A convolutional encoder is called so because it performs a convolution of the input stream with encoder's impulse responses:

$$y_i^j = \sum_{k=0}^{\infty} h_k^j x_{i-k},$$

where x is an input sequence, y^j is a sequence from output j and h^j is an impulse response for output j.

A convolutional encoder is a discrete linear time-invariant system. Every output of an encoder can be described by its own transfer function, which is closely related to a generator polynomial. An impulse response is connected with a transfer function through Z-transform.

Convolution coding is simulated below to show how coding can enhance the efficiency of a communication channel.The essential blocks used in the simulation are

- P N Sequence generator:. This block generates a maximal length pseudo noise (PN) sequence, also known as a pseudo random binary sequence (PRBS). The generator's shift register size, coefficients, initial state, and bit rate can be specified. The block can also accept an external clock. A clock level greater than 0.5 is considered high
- Convolution coder: This block implements a convolutional encoder. Block parameters include the numbers of input bits (k) and coded bits (n), the encoder constraint length (L), the input bit rate, and the generator coefficients
- Viterbi decoder : This block implements a hard decision Viterbi decoder to decode convolutionally encoded data. Block parameters include the numbers of information bits (k), coded bits (n), the encoder constraint length (L), the trellis truncation length M, the output bit rate, and the generator coefficients. This block takes as input a binary stream {0, 1} and outputs a binary stream {0, 1}. Hamming distance is used as the internal metric. Assuming that no uncorrected channel errors have occurred, the best metric output (y2) is an indication of the number of channel bit errors. The decoding delay through this block, from when the first coded bit is received to when the first information bit is produced, is equal to (kMT - kT/n), where k and n are the number of information bits and coded bits respectively, M is the Truncation Length, T = 1/R and R is the information bit rate.

Convolution coding

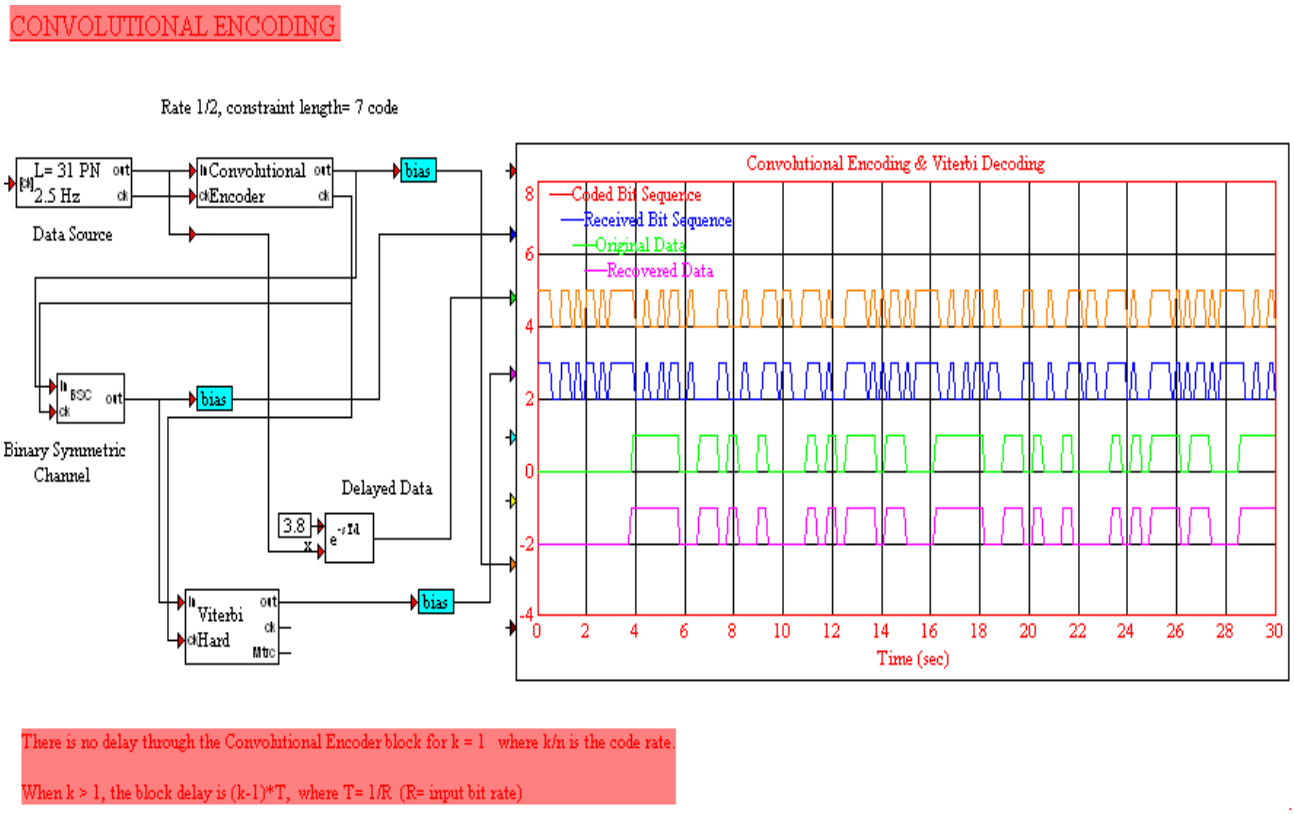


Fig 5.6

5.2.2.3 Block codes

A block code is a type of channel coding. It adds redundancy to a message so that, at the receiver, one can decode with minimal (theoretically zero) errors, provided that the information rate (amount of transported information in bits per sec) would not exceed the channel capacity.

The main characterization of a block code is that it is a fixed length channel code (unlike source coding schemes such as Huffman coding, and unlike channel coding methods like convolutional encoding). Typically, a block code takes a k-digit information word, and transforms this into an n-digit codeword.Block coding was the primary type of channel coding used in earlier mobile communication systems.

Reed-Solomon codes are block codes. This means that a fixed block of input data is processed into a fixed block of output data. In the case of the most commonly used R-S code (255, 223) –

223 Reed-Solomon input symbols (each eight bits long) are encoded into 255 output symbols.Reed-Solomon error correction is an error-correcting code that works by oversampling a polynomial constructed from the data. The polynomial is evaluated at several points, and these values are sent or recorded. Sampling the polynomial more often than is necessary makes the polynomial over-determined. As long as it receives "many" of the points correctly, the receiver can recover the original polynomial even in the presence of a "few" bad points. The Reed-Solomon code, like the convolutional code, is a transparent code. This means that if the channel symbols have been inverted somewhere along the line, the decoders will still operate. The result will be the complement of the original data. However, the Reed-Solomon code loses its transparency if virtual zero fill is used. For this reason it is mandatory that the sense of the data (i.e., true or complemented) be resolved before Reed-Solomon decoding.

The error-correcting ability of any Reed-Solomon code is determined by n − k, the measure of redundancy in the block. If the locations of the errored symbols are not known in advance, then a Reed–Solomon code can correct up to (n − k) / 2 erroneous symbols, i.e., it can correct half as many errors as there are redundant symbols added to the block. Sometimes error locations are known in advance (e.g., “side information” in demodulator signal-to-noise ratios)—these are called erasures. A Reed–Solomon code (like any linear code) is able to correct twice as many erasures as errors, and any combination of errors and erasures can be corrected as long as the inequality $2E + S < n - k$ is satisfied, where E is the number of errors and S is the number of erasures in the block.

The properties of Reed-Solomon codes make them especially well-suited to applications where errors occur in bursts. This is because it does not matter to the code how many bits in a symbol are in error—if multiple bits in a symbol are corrupted it only counts as a single error. Conversely, if a data stream is not characterized by error bursts or drop-outs but by random single bit errors, a Reed-Solomon code is usually a poor choice.

A Reed Solomon implementation has been simulated using Vissim environment. The blocks of significance are the two encoding and decoding blocks

- Encoder: Here a circular buffer is used to feed the block encoder ,ie Reed Solomon encoder. The Reed Solomon encoder -Solomon (RS) encoder. RS codes are a type of

block code and are systematic. In an RS(n, k) code, n - k parity symbols are added at the end of the k input symbols, resulting in a total of n output symbols. An RS(n, k) code is capable of correcting up to (n - k)/2 error

- Decoder: This block implements a Reed-Solomon (RS) decoder. RS codes are a type of block code and are a subclass of BCH codes. RS codes use non-binary symbols from a Galois Field (GF) and are systematic. In an RS(n, k) code, n - k parity symbols are added at the end of the k input symbols, resulting in a total of n output symbols. An RS(n, k) code is capable of correcting up to (n - k)/2 errors.

Reed Solomon coding

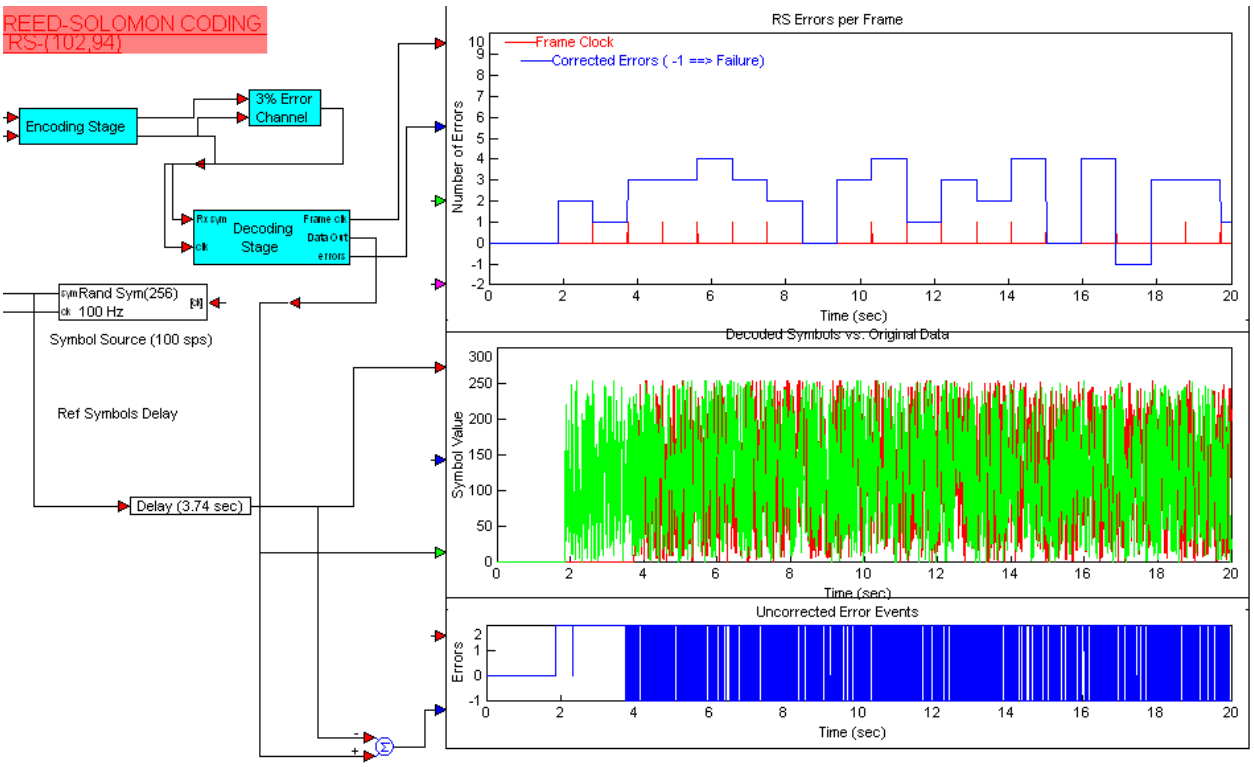


Fig 5.7

5.3 TURBO CODES

Turbo codes are a class of high-performance error correction codes which are finding use in deep space satellite communications and other applications where designers seek to achieve maximal information transfer over a limited-bandwidth communication link in the presence of data-corrupting noise. Of all practical error correction methods known to date, turbo codes and low-density parity-check codes (LDPCs) come closest to approaching the Shannon limit, the theoretical limit of maximum information transfer rate over a noisy channel.

Turbo codes make it possible to increase data rate without increasing the power of a transmission, or they can be used to decrease the amount of power used to transmit at a certain data rate. Its main drawbacks are the relative high decoding complexity and a relatively high latency, which makes it unsuitable for some applications..

5.3.1 The encoder

The encoder sends three sub-blocks of bits. The first sub-block is the m-bit block of payload data. The second sub-block is n/2 parity bits for the payload data, computed using a recursive systematic convolutional code (RSC code). The third sub-block is n/2 parity bits for a known permutation of the payload data, again computed using an RSC convolutional code. That is, two redundant but different sub-blocks of parity bits for the sent payload. The complete block has m+n bits of data with a code rate of m/(m+n). The permutation of the payload data is carried out by a device called interleaver.

Hardware-wise, turbo-code encoder consists of two identical RSC coders, C_1 and C_2, as depicted on the figure, which are connected to each other using a concatenation scheme, called parallel concatenation:

Turbo Encoder

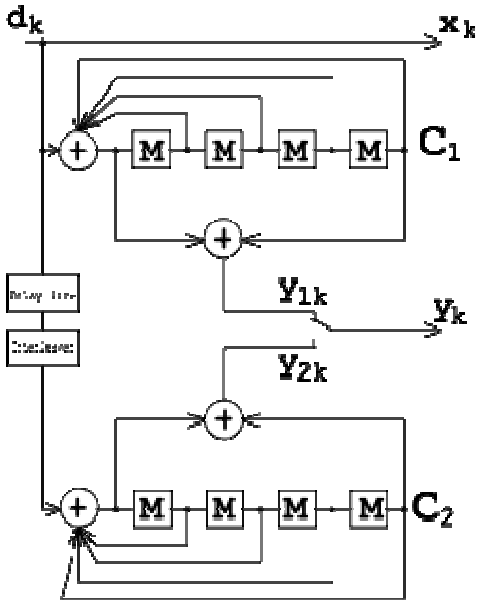


Fig 5.8

On the figure, M is a memory register. Delay line and interleaver force input bits d_k to appear in different sequences. At first iteration, the input sequence d_k appears at both outputs of the encoder, x_k and y_{1k} or y_{2k} due to the encoder's systematic character. If the encoders C_1 and C_2 are used respectively in n_1 and n_2 iterations, their rates are respectively equal to

$$R_1 = \frac{n_1 + n_2}{2n_1 + n_2,} \; R_2 = \frac{n_1 + n_2}{2n_2 + n_1.}$$

5.3.2 The decoder

The decoder is built in the similar way as the above encoder - two elementary decoders are interconnected to each other, but in serial way, not parallel. The DEC_1 decoder operates on lower speed (i.e. R_1), thus, it is intended for the C_1 encoder, and DEC_2 is for C_2 correspondingly. DEC_1

yields a soft decision which causes L_1 delay. The same delay is caused by the delay line in the encoder. The DEC_2's operation causes L_2 delay.

Turbo Decoder

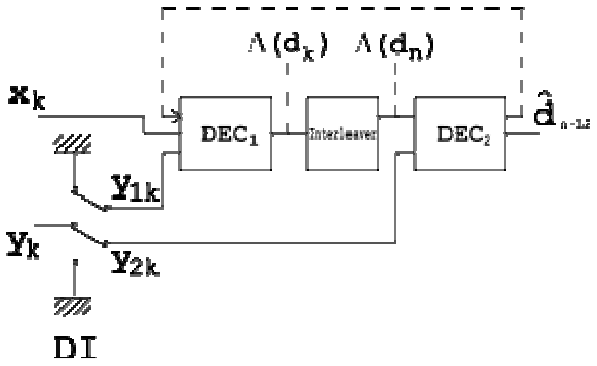


Fig 5.9

An interleaver installed between two decoders is used here to scatter error bursts coming from DEC_1 output. DI block is a demultiplexing and insertion module. It works as a switch, redirecting input bits to DEC_1 at one moment and to DEC_2 at another. In OFF state, it feeds both y_{1k} and y_{2k} inputs with padding bits (zeros).

The inbuilt Turbo Codes functionality in Vissim can be used to simulate a communication environment using turbo codes. Though actual understanding og the complexity of the turbo codes is a little difficult ,yet the simulation certainly allows us to visualize the efficiency of the coding in a noisy environment. Shown below is a Turbo coding and decoding simulation.

Turbo codes

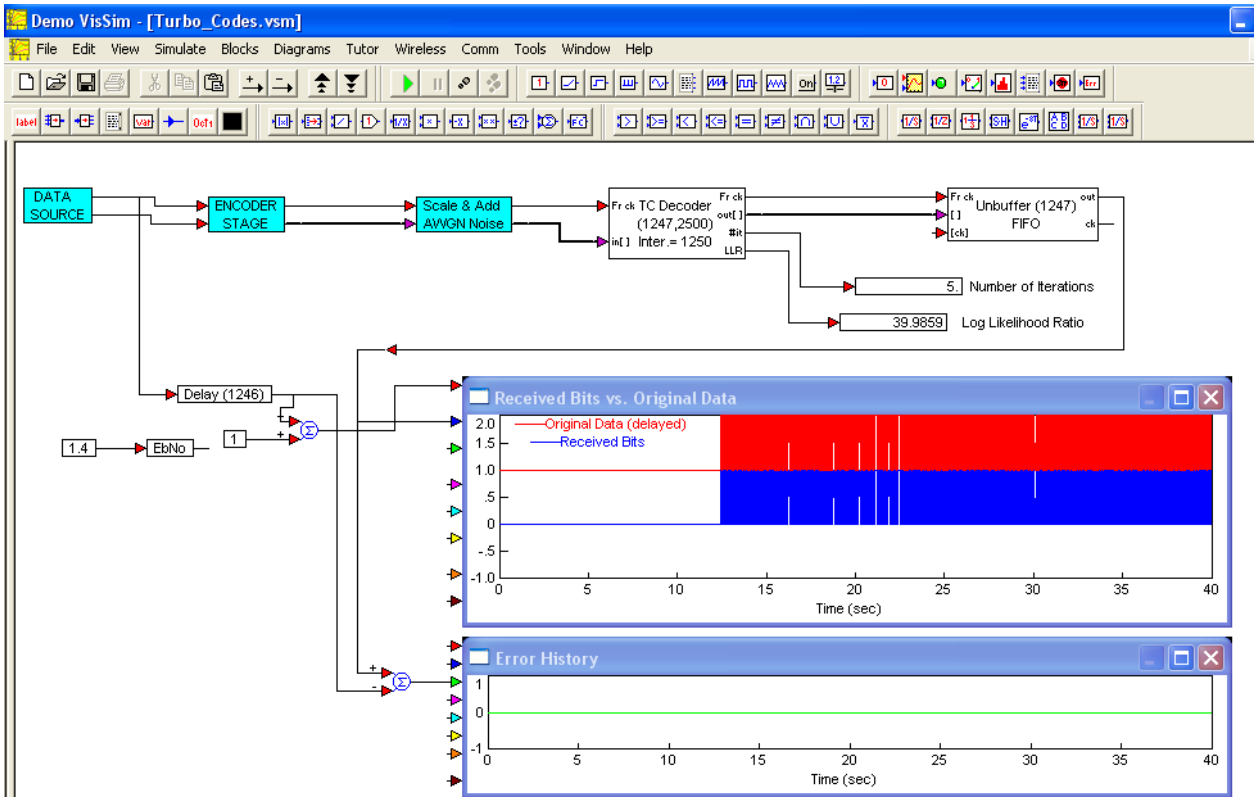


Fig 5.10

REFERENCES

1. Wireless Communications - T .S.Rappaport
2. Principles of Communication System – Taub Schilling
3. Communication Systems – Simon Haykin
4. Visual Simulator User Guide
5. Internet References-

http://www.wikipedia.com/

http:// www.thensys.com/index.php?title=dataflow_programming

http://www.informit.com