REAL TIME IMPLEMENTATION OF SOFTWARE RADIO ON DSP

SYNDICATE MEMBERS

NC AMMAN DURRANI (LDR)
CAPT OMER ABDUL RASHEED
CAPT ASSAD ULLAH AMIR SADIQ

PROJECT SUPERVISOR

LEC IRTIZA ALI

Dissertation submitted for partial fulfillment of requirement of MCS/NUST for the award of the B.E. degree in Telecommunication Engineering

DEPARTMENT OF ELECTRICAL ENGINEERING
MILITARY COLLEGE OF SIGNALS
NATIONAL UNIVERSITY OF SCIENCE & TECHNOLOGY
RAWALPINDI

APRIL 2004

TABLE OF CONTENTS

DEDICATION

Dedicated to our parents and teachers who have been a source of constant encouragement for us throughout our degree……and our lives

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award of qualification either at this institution or elsewhere.

ACKNOWLEDGMENTS

May Allah bless us through our life as he blessed us through this project for He gave us hope in each failure and exalted us with every success after that brief encounter with dismay.

We are extremely thankful to Lec Irtiza Ali, our project supervisor who helped us at each step. We wish to thank Dr. Jamil Ahmad because he helped us in choosing such an excellent project. Although he left the college soon after that but his initial guidance helped us a lot.

We are also grateful to Colonel Dr. Ashraf who never hesitated in helping us and giving us his precious time.

We wish to extend our gratitude to our colleagues especially Lieutenant Suhail and NC Danish who provided help whenever we needed it. We are also thankful to the DSP Lab NCO Hav Iftikhar who was kept on duty late through the night so that we could work in the Lab.

Special thanks to EE Dept. and the staff members of EE Dept. for their assistance and cooperation in the provision of material resources. We also extend our deep gratitude to all those who we might have not named and who encouraged us in achieving our goals.

ABSTRACT

Since early 80s the exponential blowup of the cellular mobile systems has produced a plethora of analog and digital standards. The competition between the telecommunication giants and the strive for excellence has hindered the development of a unique standard worldwide. This obstacle has denied the users many facilities especially global roaming for many years.

In early 90s the Software Radio Concept came to the rescue with the qualities like multistandard, multiband, multiservice, reconfigurable and reprogrammable by software. It has therefore emerged as the hottest technology in the field of communication.

An assay to implement the software for a user terminal which is able to dynamically adapt to the radio environment, is made in our project. Our design uses three techniques namely Binary Phase Shift Keying, Quadrature Phase Shift Keying and Pi / 4 Quadrature Phase Shift Keying. Hence the processing of different modulation techniques is achieved successfully, realizing the property of reconfigurability which is the essence of the Software Radio.

An algorithm for the symbol timing recovery was designed which is capable of synchronizing with two samples per symbols. It was compared with the Gardner algorithm and performance analysis of both the algorithms was given. Source Coding was done using ADPCM technique which

All the blocks were designed and tested in Matlab and implemented on TMS320C6711 Digital Signal Processor (DSP). Architecture of the DSP was fully understood and programming was done using the C Language. C Language programs were converted into the Common Object Oriented Format (COFF) using the Integrated Development Environment, “Code Composer Studio” provided by Texas Instruments.

Future work has been recommended in the implementation chapter of this dissertation.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

CHAPTER 2 SOFTWARE RADIO

CHAPTER 3 Sampling and Quantization

CHAPTER 7 DSP ARCHITETURE

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

1.1 MOTIVATION:

The project was undertaken considering the immense interest provoked by “software radios” in the recent years. Great development has been achieved in this area with applications ranging from home appliances to industrial gadgets. All this progress is apparent in the western world, but work in this area cannot be viewed in our part of the world so we aimed to initiate the project as dissertation submitted for partial fulfillment of requirement of MCS/NUST for the award of the B.E. degree in Telecommunication Engineering.

1.2 DOCUMENTATION SETUP:

The explanation of the techniques used in the project design is given in the later chapters. The setup of the chapters and the topics included are explained below.

1.2.1 Chapter no. 2:

This chapter describes the basic concept of the Software Radio and the types of architecture that have already been developed in the industry. It gives an explanation of the key components of the radio, their working and the applications of the software radio in various fields. The chapter also includes specific details of the use of digital signal processing chip and its comparison with the use of (ASICs) Application-specific Integrated Circuit and FPGA (Field Programmable Gate Array).

1.2.2 Chapter no. 3

Chapter no.3 gives a thorough explanation of the Analog to digital conversion process which is the first step in the digitizing process. The Chapter includes details about the three major steps required in the traversing from the analog to digital domain. The portions of Sampling, Quantizination and Coding have been explained. The topics of Sampling theorem and Nyquist Criteria has been dealt with.

The problems appearing in the digitizing process such as aliasing are depicted and the solutions for these are also given. The functions of various filters are expressed in the follow up.

The various types of Quantization are pointed out along with the issues regarding companding. The problems that prop up in the Quantizing process such as jitter are also detailed in this chapter.

We have also explained Multi rate systems and the use of interpolation and decimation. In the end we have reviewed the adc that we used in the project.

Chapter no. 4 is about source coding. It gives a background of source coding and explains the need and requirements of this procedure. The chapter also provides an anlysis of the different techniques used worldwide for this purpose.

We have added a complete narrative of adaptive differential pulse code modulation (ADPCM), which is the technique, used to reduce the data rate of the signal. An introduction to the Code Excited Linear Predictive Coding (CELP) which is another technique of source coding is also provided.

A comparison of the two techniques along with their benefits and desired performance is depicted. This is followed with a description of the discrepancies visible in these techniques and an analysis of another technique namely, Vector Sum Excited Linear Predictive Coding is also provided.

After this the theory of operation of ADPCM and its application in the compression and decompression process is given, as this is technique which we have used in our project.

1.2.3 Chapter no. 5

Chapter 5 explains the problems faced during the transmission and the effects that these problems impose on the signal. As an example of this problem intersymbol interference has been described. To solve these harms its countermeasures are depicted and pulse shaping and matched filter are especially explained as the main techniques used by us to resolve these problems. The difference between ideal and practical pulse shaping is also reviewed.

A complete description of the raised cosine pulses, including the design of the root raised cosine filter is given with analysis of their representations in both the time domain and the frequency domain. The design of Matched filter has been eloquently dealt with in the prospect of its role in improving the Signal to Noise ratio of the signal.

An in depth explanation of the Eye diagram is also provided, including its importance in depicting the effect of the root raised cosine filters. The chapter is conclude with a brief description of the Finite Impulse Response filter because of its use in the filtering techniques used in the modulation techniques used by us.

1.2.4 Chapter No.6

The chapter commences with a review of the Encoding and the various line coding techniques. We have included the goals of the line codes, their power spectral densities and the problems faced with in the various codes. The chapter also encompasses the modulation process and it is viewed with respect to the baseband and passband signals.

Advantages of the modulation process are given. Although the different characteristics of the signal that can be modified are reviewed but the main emphasis is on the Phase shift Keying and more detail has been given about the binary phase shift keying (BPSK), the quardrature phase shift keying (QPSK) and the pi/4 quardrature phase shift keying (π / 4 QPSK).

The chapter includes an explanation of the various forms of representation of the signal e.g polar display, I/ Q formats and constellation diagram. The latter two are explained with respect to their use in the modulation techniques used by us and the formation of the I and Q, orthogonal signals in the transmitter and receiver.

We have also explained the power spectrum and the power spectral density in this chapter as they have been referred to in both chapter no.5 and chapter no. 6. We have also detailed about the constant envelope modulation and the differences in the various modulation techniques

1.2.5 Chapter no. 7

Our project has used the Texas Instrument’s digital Signal Processor TMSC3206711. This chapter reviews the requirements of using this board for the processing required in our project. An extensive appraisal of the properties of this board along with the history o fits family, its development process and the advantages of using this board is made.

The architecture of the central processing unit is assessed and a complete evaluation of the addressing modes and the memory of the TMS20C6000 is made.

The various peripherals of the board which assist in interfacing the platform of the processor with other devices are reviewed such as the direct memory access (DMA),
and the host processor interface. A description of the code composer studio, which is used as an interface to implement the software on the processor board, is also added.

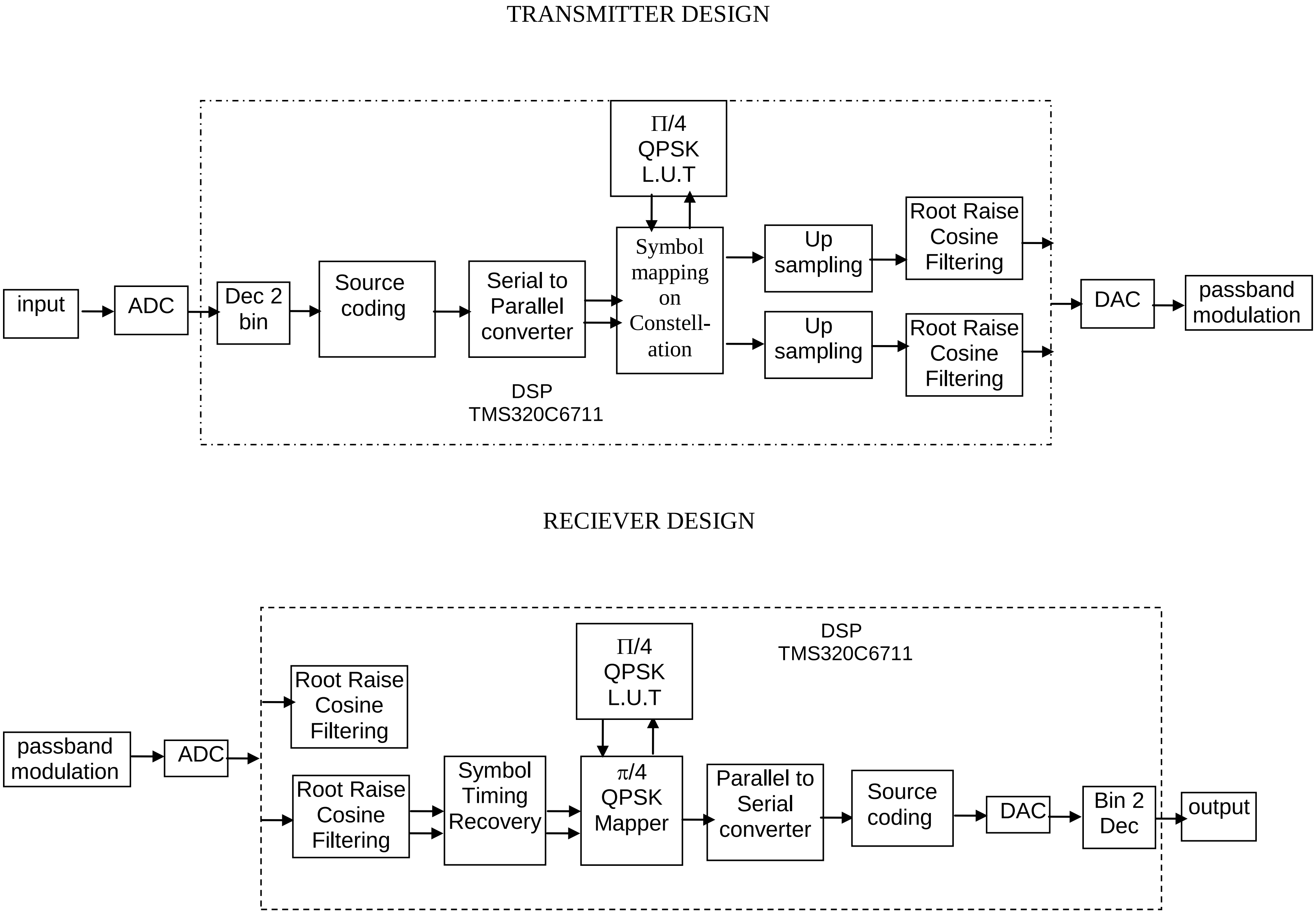
1.2.6 Chapter 8:

This chapter is the essence of all the hard work that went into the project. The analysis and results of each module of the project are projected. Graphs and figures representing the simulation results of the different modules are added to enhance the understanding of the various steps of the project.

The results achieved at the transmitter and receiver side seen at the oscilloscope after the processing in the digital signal processor are also added.

1.2.7 OUR DESIGN

The figures on next page show our transmitter and receiver design for pi/4 QPSK



TRANSMITTER DESIGN

input
ADC
Dec 2 bin
Source coding
Serial to Parallel converter
Π/4 QPSK L.U.T
Symbol mapping on Constell- ation
Up sampling
Up sampling
Root Raise Cosine Filtering
Root Raise Cosine Filtering
DSP TMS320C6711
DAC
passband modulation

RECIEVER DESIGN

Figure 1.1 Transmitter and Receiver Architecture

SOFTWARE RADIO CONCEPT

2.1 INTRODUCTION.

The term 'software radio' means software implementation of the user terminal to enable it to dynamically adapt to the radio environment in which it is, time by time, located. It means that the radio interface functionalities of Tx and Rx usually implemented by dedicated hardware should be software defined. Dedicated hardware would be replaced with DSP (Digital Signal processing engines), which will execute the necessary software.

For example, digital signal processing for demodulation of CDMA and FDMA (frequency hopping) modulated signals can be executed within the same reprogrammable device with simple changing of software. For executing same operations in conventional manner, two completely different hardware architectures are required. Reprogrammable digital signal processing engines (DSP) are suitable for performing processing task.

Software radio is an emerging technology intended to build flexible radio systems, multiservice, multistandard, multiband, reconfigurable and software - reprogrammable.

[1]

2.2 ARCHITECTURES OF SOFTWARE RADIOS

1. SDR IF processing architecture
2. Direct conversion receiver (DCR) architecture

Figure 1. presents SDR IF processing architecture. It has traditional analog conversion of the signal of interest from RF to IF in down conversion side, and from IF to RF in the up conversion side. A digital down converter typically consists of a digital local oscillator, a digital mixer and decimation filter. When operation on multiple channels is needed, these operations can be performed in distributed manner by using transmultiplexer. On the other hand, digital up converter consists of a digital local oscillator, a digital mixer and interpolation filter.

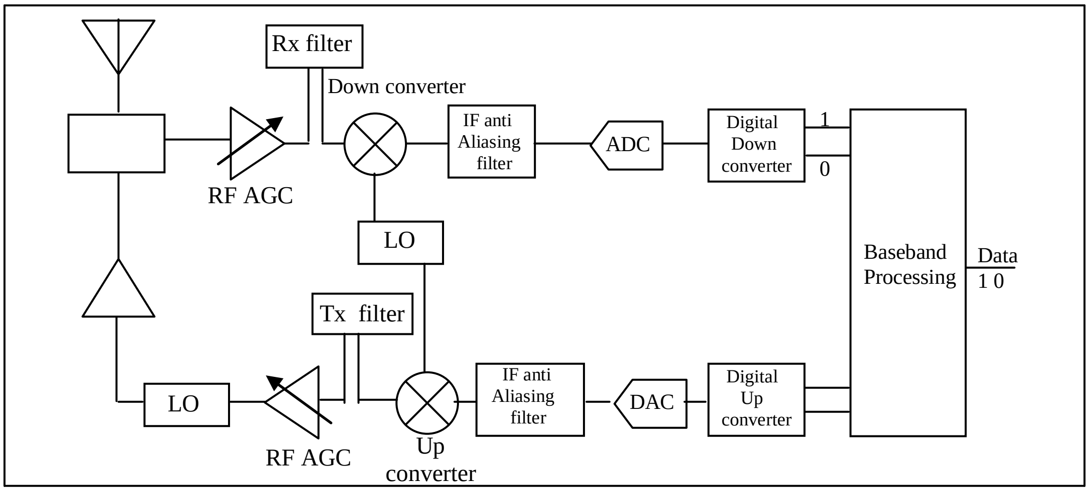


Fig 2.1 . Digital IF Configuration

Transmultiplexer can be used in up conversion of multiple channels. For IF it is important, whether the down converter is operating on a narrowband, wideband or a multiple narrowband signal. Narrowband signal has bandwidth significantly narrower than input signal's bandwidth, whereas bandwidth of wideband signal is greater than 10 percent of a whole input signal's bandwidth. Multiple narrow- band signals have similar characteristics as narrow- band signals. This characterization defines constrains for signal processing components. Narrowband digital IF processing is suitable for applications in which the large dynamic range and accurate selectivity are required, and where multiple signals of interest are with different bandwidth.

Principle of narrowband IF processing is to sample a real IF signal and to mix it digitally with a quadrature local oscillator set to the signal of interested frequency. After this central frequency of the signal of interest is translated to DC. Signal derived as a mixing result is than low pass filtered (LPF) and decimated to achieve a narrowband quadrature baseband signal with a baseband sampling rate commensurate with its bandwidth. Advantage of digital LO is that their phase offset and center frequency can be changed to any desired value within the one clock cycle. Narrowband multichannel digital IF is suitable for situations where multiple channels of the same bandwidth and equal spacing are of interest (i.e. cellular BS). It is realized through a transmultiplexer based on a discrete Fourier transform (DFT) filter bank. Tmux receiver translates one wideband IF sampled signal stream into multiple quadrature baseband streams, one per subscriber channel.

The advantages of wideband single-channel digital IF include: large dynamic range, accurate frequency selection, accurate filter characteristics and flexibility in the capture of multiple signal with different bandwidths. Implementation is same as for the narrowband signal. Only difference is in decimation factor, which is much lower for a wideband processing.

Direct conversion receiver (DCR) architecture is another possible solution for SDR. Figure 2. presents a basic DCR configuration where the received signal is directly

downconverted to baseband by a quadrature mixer. The downconverted in-phase and quadrature (IQ) signals are filtered with anti-aliasing LPFs, which have variable cut-off frequencies. After that, signals are converted from analog to a digital form by IQ ADC.

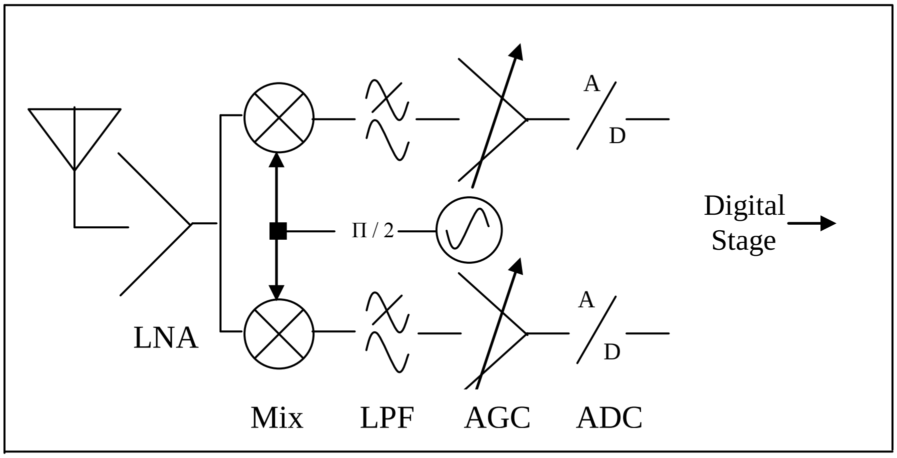


Fig 2.2 - Basic DCR configuration

Output of ADC is fed to digital stage where selection of desirable channels is performed by software defined filter with programmable cutoff frequency. At the end, an original data sequence is recovered. Advantage of DCR configuration over classic heterodyne is that DCR architecture has no image response, so the image rejection filters are not needed. On the other hand imperfect components of DCR cause residual or latent image at zero frequency (DC). Second thing is that in-band linearly modulated large signals cause non-linear distortion around DC in downconverter mixer (linearly modulated signals have constant envelope). Therefore, system level compensation method for a suppression of the nonlinear distortions is required.

In a figure 3, another DCR configuration is shown. The received signal is first directly downconverted to baseband, where the entire input band is roughly selected by antialiasing LPF. After filtering, a whole signal bandwidth is reconverted to programmable IF. Resultant IF signal has limited bandwidth. From that point, signal can be directly digitalized and than digitally downconverted to baseband for the selection of desired channels. Suitable characteristic of this approach is the ability to select a desirable frequency for the IF in order to employ ADC with suitable power consumption for portable terminals. Direct conversion architecture is attractive for handset application because of its suitability for achieving a miniature low-cost transceiver, since bulky and high cost passive filters for receiver and transmitter chains are not necessary, and it is compatible with multiple standards. This approach is effective for single channel applications, but multiple simultaneous channels have fewer spurs and better S/N ratio in near-far conditions through the digital IF architecture.

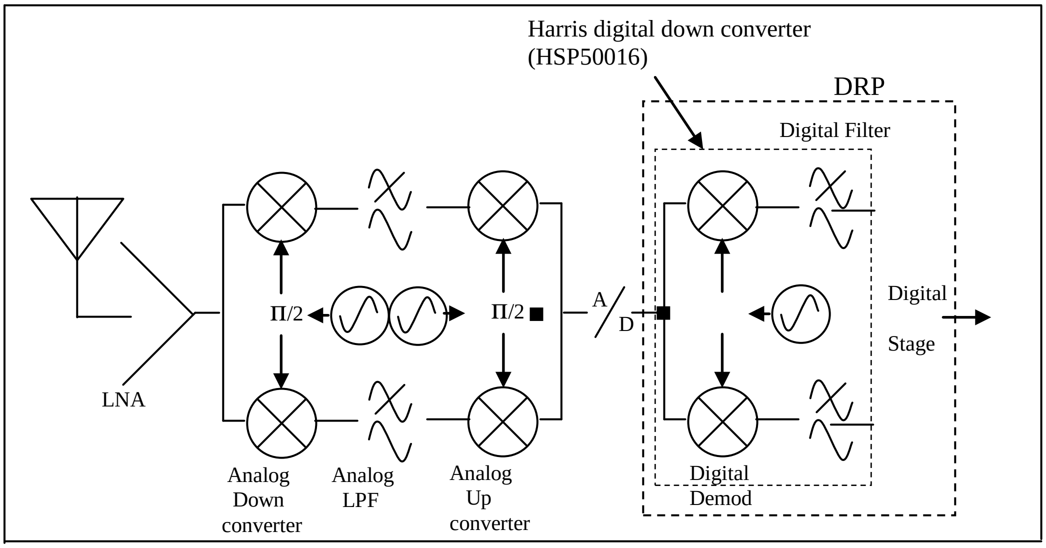


Fig 2.3 – An Up conversion DCR configuration

2.3 KEY COMPONENTS OF SOFTWARE-DEFINED RADIO

1. MMIC RF component
2. Analog to digital converter
3. Digital signal processing circuit.

2.3.1 Integrated RF Component

MMIC (monolithic microwave integrated circuit) technology is used for integrating RF components on one chip. RF components include active components such as transistors and passive components such as resistors, capacitors, and inductors. There are two major materials used for MMIC: GaAs and Si. GaAs is used for frequency ranging from 1 GHz up to 100 GHz, and Si is used for frequency below 10 GHz. The CMOS technology is advancing so that CMOS integrated circuit will be able to handle frequencies of several GHz in a few years. If CMOS analog RF components become available, it will be possible to process not only RF analog signals but also baseband digital signals all on the same chip.

2.3.2 Analog-to-Digital Converter

Key parameters that define the performance of analog-to-digital converters are sampling rate and the resolution (number of bits per sample). A detailed survey of analog-to-digital converters can be found in [2]. Figure 6.6 shows the relationship between the sampling frequency and the number of bits per sample.

One of the key parameters of an analog-to-digital converter is sampling rate; Software-defined radios sometimes use undersampling. When undersampling is done, the sampling rate must be larger than twice the band pass filtered signal bandwidth.

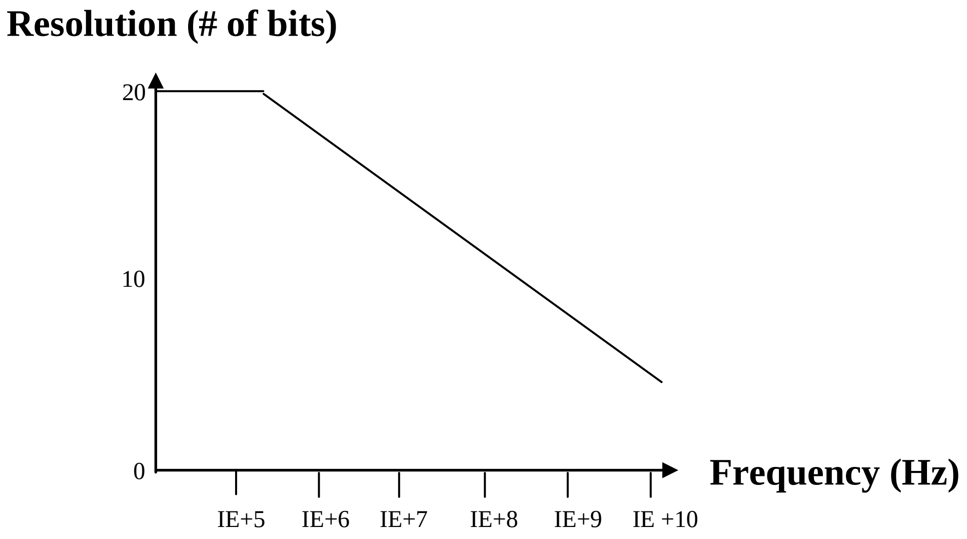


Figure 2.4:Relation between sampling frequency and the number of resolution bits

Another key parameter is dynamic range. In the conventional approach, each radio only deals with a narrow band; by filtering out interfering signals, the receiver can concentrate on the desired one, adjusting gain to optimize signal-to-noise performance and extracting a weak signal from a noisy background. However, with a wideband receiver, none of the signals should be filtered out, because they are all required. There will be a wide range of signals: very strong ones from powerful transmitters nearby, and very weak ones almost buried in noise. As a result, the receiver must have an extremely wide dynamic range for enough sensitivity to accurately recover the weak signals, without their being swamped by the strong ones. It must also be extremely linear; any distortion or harmonics will generate images of strong signals, indistinguishable from true signals.

Performance measure of ADC's can be expressed as $2^m F_s$, where m is the number of bits of a sample and F_s is the sampling range. When the sampling rate is between a few Msps and a few Gsps, which usually covers the software defined radio applications, this performance measure is usually limited by an aperture jitter. The aperture jitter is the variation of the time difference between the sample command time and the actual time the analog input signal is sampled. The jitter originates from thermal Gaussian distributed noise [3].

The improvement of ADC performance will be done mainly by reducing the aperture jitter, but the progress of sampling bits for a given sampling rate has been fairly slow: only 1.5 bits over the last couple of years [2].

There is also an attempt to make a very high speed analog-to-digital converter using superconductor technology [4]. It may be possible to sample analog signals faster than semiconductor analog-to-digital converters. However, there is a problem of the size of a cooling equipment, which is much larger than the ADC device.

2.3.3 Digital Signal Processing Circuit

When an intermediate frequency signal is sampled by an ADC, signals below IF frequency must be processed digitally. The digitized intermediate frequency signal from ADC is down-converted, filtered, and decimated, before the slower speed signal processing is performed by a DSP.

The slower speed signal processing include channel decoding including error correction, and source decoding such as data decompression, description etc. In the transmitter side, the slower signal processing is performed first: source coding such as data compression and encryption, and channel coding including error correction. The data is then filtered for each application, interpolated, and upconverted before its signal is sent to a DAC. Signal processing of high speed signals, such as intermediate frequency signals, requires a very high-speed signal processing circuit. The speed may be as high as several thousand MIPS (million instructions per second). Suitable integrated circuits are DSP's (digital signal processors), FPGA (Field Programmable Gate Array), or software-radio-specific ASIC.

A DSP chip does signal processing by fetching instructions and data from memory, does operations, and stores the results back to memory, just like a regular CPU. The difference between a DSP chip and a CPU chip is that a DSP chip usually has a block that does high-speed signal processing, especially a block called MAC (Multiply and

Accumulate). By calling different routines in memory, a DSP chip can be reconfigured to perform various functions. Some of commercially available high-speed DSP chips are Texas Instruments' TMS320C6202 and Analog Devices' ADSP-21160M SHARC with the speed of 2000 MIPS and 600 MFLOPS respectively.

ASIC (Application-specific Integrated Circuit) is an integrated circuit that is designed to perform a fixed specific task. Examples of signal-processing specific ASIC's are DDC (digital down converter) chip, and digital filter chips. The disadvantage of ASIC is that a user cannot change the function of the chip.

FPGA (Field Programmable Gate Array) is able to perform any task by mapping the task to the hardware. On the other hand, FPGA has a reconfigurability capability that ASIC does not have. Reconfigurability is a feature, which enables FPGA to realize any user hardware by changing the configuration data on a chip as many times as needed. Even though the number of gates realizable on one FPGA chip such as Xilinx's Virtex is in the range of 100,000 gates to 1,000,000 gates which is smaller than several million gates of an ASIC, this reconfigurability capability will be very useful in software-defined radio in the future [6, 7].

Typical FPGA's consist of an array of reconfigurable look-up table logic block to implement combinational and/or sequential logic, and a reconfigurable routing resource that interconnect logic blocks. Some special signal processing algorithms suitable for FPGA architectures have been developed such as distributed arithmetic algorithm [6, 8, 9]. The distributed arithmetic method uses look-up tables for fast signal processing, which makes LUT-based FPGA's very suitable. The FIR filtering using distributed algorithm, for example, has the same speed whether the number of filter taps is 1 or 100. This makes it suitable for implementing a high-speed filter with large number of taps. Many other applications taking advantage of FPGA architectures will appear in the future. A new FPGA feature that some companies are developing is dynamic reconfiguration. For example, Jbits tool from Xilinx enables users to change configuration of portion of FPGA's while FPGA is operating. This is still a new technology, but this will be a very useful tool when, for example, a receiver needs to reconfigure reception algorithms in order to receive signals that come through a dynamically changing channel.

Software-radio-specific ASIC is a new type of chip that has a fixed portion for common signal processing and a reconfigurable portion that needs to be changed depending on different wireless standards such as different cellular phone standards. Since this is targeted to more specific application than a general-purpose FPGA chip, it is more cost-effective and has a higher performance and consumes less power than FPGA. Some software-radio-specific ASIC’s also have dynamic reconfiguration capability. Among the chips mentioned above, chips that have general-purpose reconfigurability features are DSP’s and FPGA’s. Table 2.1 shows a table detailing the difference of features between DSP’s and FPGA’s.

Table 2.1 Comparison of FPGA and DSP

	FPGA chip	**DSP chip**
Programming Language	VHDL, Verilog	C, Assembly language
Ease of Software Progamming	Fairly easy. However, a programmer needs to understand the hardware architecture before programming.	Easy
Performance	Can be very fast if an appropriate architecture is designed.	Speed is limited by the clock speed of DSP chip.
Reconfigurability	SRAM type FPGAs can be reconfigurable for infinite times.	Can be reconfigurable by changing a program memory content.
Reconfiguration method	Reconfiguration is done by downloading configuration data to a chip electronically	Reconfiguration is done by simply reading program at different memory address.
Areas where FPGAs can outperform DSPs, or vice versa	FIR filter, IIR filter, correlator, convolver, FFT, etc	Signal processing program of sequential nature
Power consumption	Can be minimized if the circuit is designed to save power, or if the power is dynamically controlled	Even if a program A is larger than a program B, power consumption does not change as long as the number of memory chips is the same
Implementation method of MAC	Parallel multiplier/adder or a distributed arithmetic.	Repeated operation of MAC function
Speed of MAC	Can be fast if a parallel algorithm is used. If a filter is implemented using distributed arithmetic, the speed does not depend on the number of taps.	Limited by the speed of MAC operation of a DSP chip. If a filter is implemented, the speed becomes slower if the number of taps increases.
Parallelism	Can be parallelized to achieve high performance.	DSP chip programming is usually sequential and cannot be parallelized

Sampling and Quantization

3.1 INTRODUCTION

Analog to Digital Conversion is the first step in moving from the analog to the digital domain thus ADCs are essential parts in signal processing systems. Since these areas are developing rapidly, there is a need for high-performance ADCs. When a system is designed, most important task is to choose the ADC. It is difficult since there is not much knowledge about what type of the ADC is required at the moment and will it suffice for future use. This leads to ADCs with unnecessary high performance being used, and this makes the system unnecessary complex and expensive. One way to find suitable requirements would be to make system simulations using an accurate ADC model which includes all the performance limiting errors. That would give good knowledge about the limits in accuracy, sampling rate, and bandwidth. Such a model would also be a good tool to test error hypotheses. [6]

The following figure depicts the transmission of an analog signal from one end to the other. It is being digitized in the process with the help of analog to digital converter. For the purpose of reducing errors and loss of information the analog signal is first input into the filter before putting it to the analog to digital converter. The processing is done once the signal is converted to digital. At the receiver end the digital to analog converter is used to convert the signal back to analog signal and the reconstruction filter helps in making the signal more intelligible.

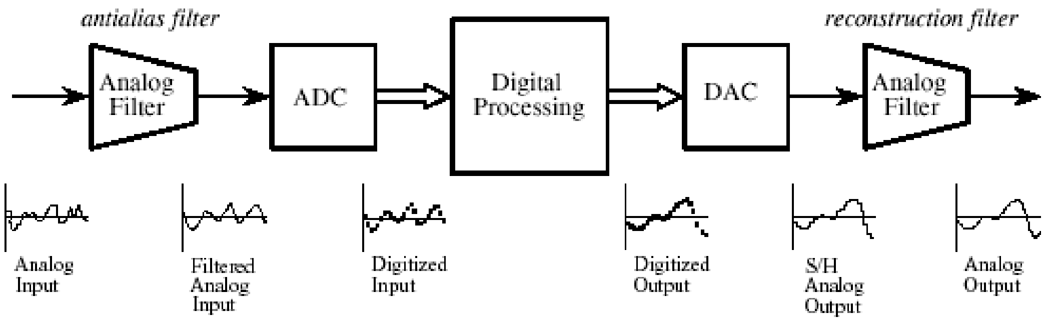


Figure 3.1 (a) ADC's role in the transmission process

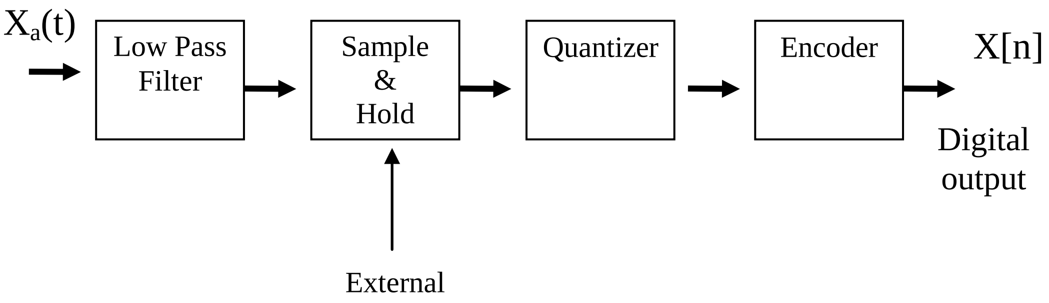


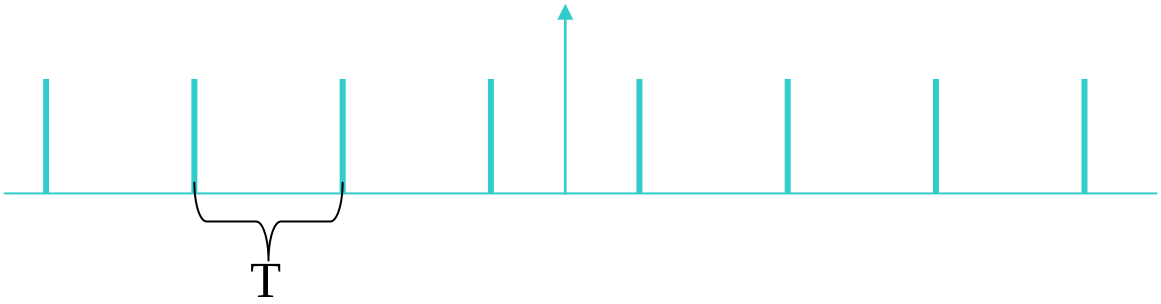
Figure 3.1 (b)

3.2 SAMPLING

3.2.1 **Definition**: Sampling Function or Impulse Train is defined by:

$$S_T(x) = \sum_{k=-\infty}^{\infty} \delta(x - kT)$$

where T is the sample spacing



- **As** $\partial_{T_S}(t)$ **is a periodic signal of Ts, it can be expressed as a Fourier series.**

$$\partial_{T_S}(t) = \frac{1}{T_s}[1 + 2\cos w_s t + 2\cos 2w_s t + 2\cos 3w_s t + ...]$$

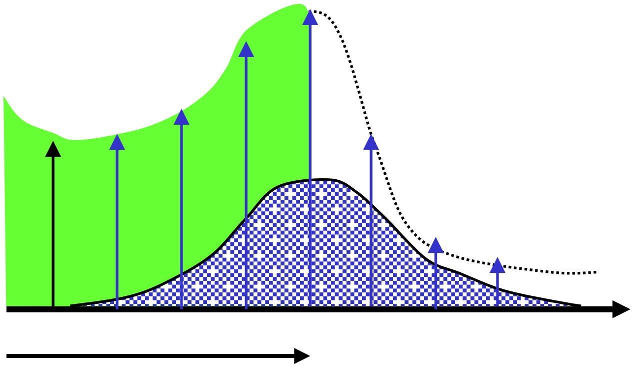
$$G`(t) = g(t)\partial_{T_S}(t)$$

$$G`(t) = \frac{1}{T_s}[g(t) + 2g(t)\cos w_s t + 2g(t)\cos 2w_s t + 2g(t)\cos 3w_s t + ...]$$

3.2.2 **Frequency Domain Representation**. To find G`(w), the Fourier transform of g`(t), we would take the Fourier transform term by term

$$G`(w) = \frac{1}{T_s}\sum_{n=-\infty}^{\infty} G(w - nw_s)$$

3.2.3 **Spatial Domain Representation**



3.2.4 **Mathematical Representation:** g(t)*h(t) is equal to :

Convolution:

$$\int_{-\infty}^{\infty} g(t)\,h(x-t)\,dt$$

Evaluated at discrete points (sum)

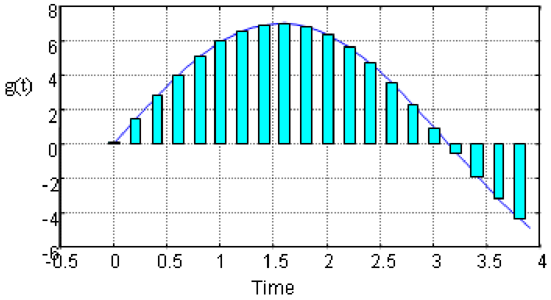
Multiplication:

$$G(\omega)\cdot H(\omega)$$

3.2.5 Applications

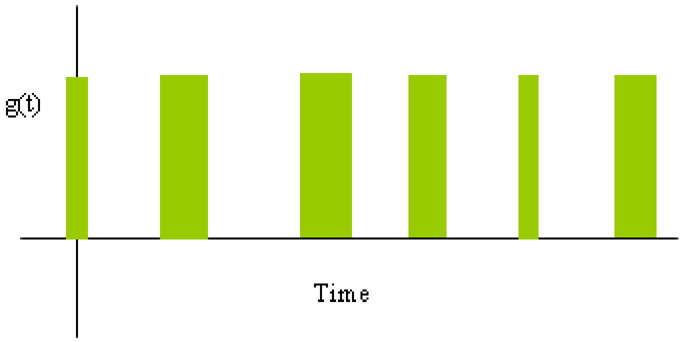
Some Applications of the Sampling Theorem:

3.2.5.1 Pulse Amplitude Modulation (PAM)



PAM

3.2.5.2 **Pulse Width Modulation (PWM)**. Pulse locations are the same but their widths change



PWM

3.2.6 Process of Analog to Digital Conversion

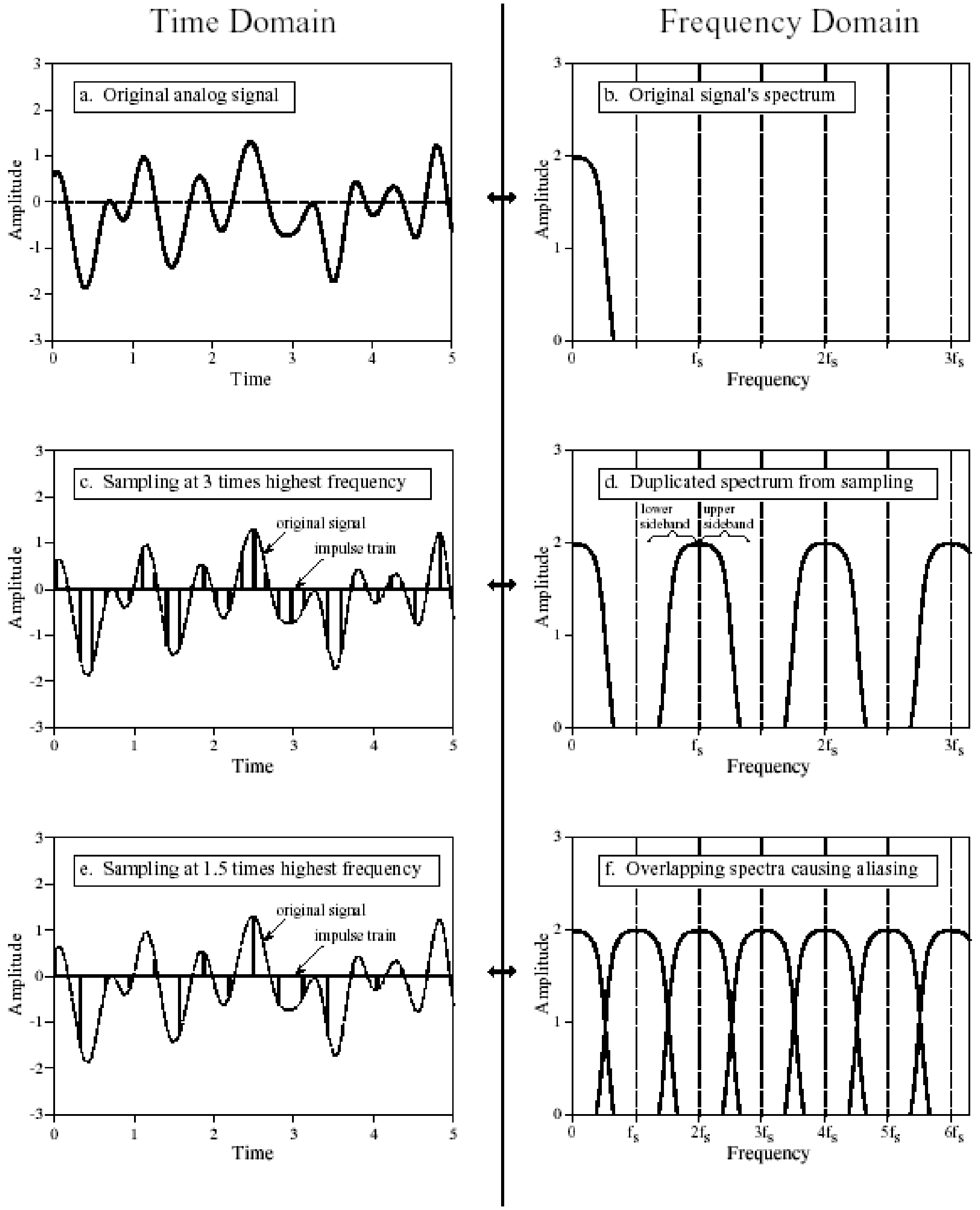


Fig 3.2 [5]

The sampling theorem in the time and frequency domains. Figures (a) and (b) show an analog signal composed of frequency components between zero and 0.33 of the sampling frequency, fs. In (c), the analog signal is sampled by converting it to an impulse train. In the frequency domain, (d), this results in the spectrum being duplicated into an infinite number of upper and lower sidebands. Since the original frequencies in (b) exist undistorted in (d), proper sampling has taken place. In comparison, the analog signal in (e) is sampled at 0.66 of the sampling frequency, a value exceeding the Nyquist rate. This results in aliasing, indicated by the sidebands in (f) overlapping.

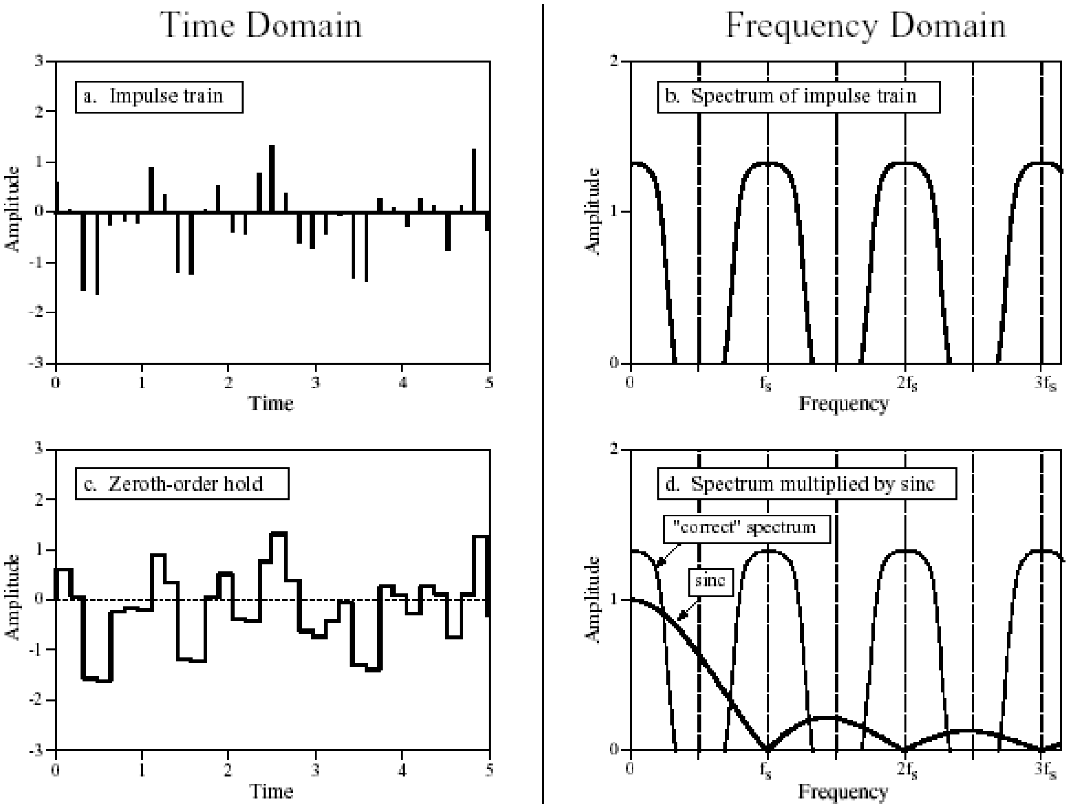


Figure 3.3

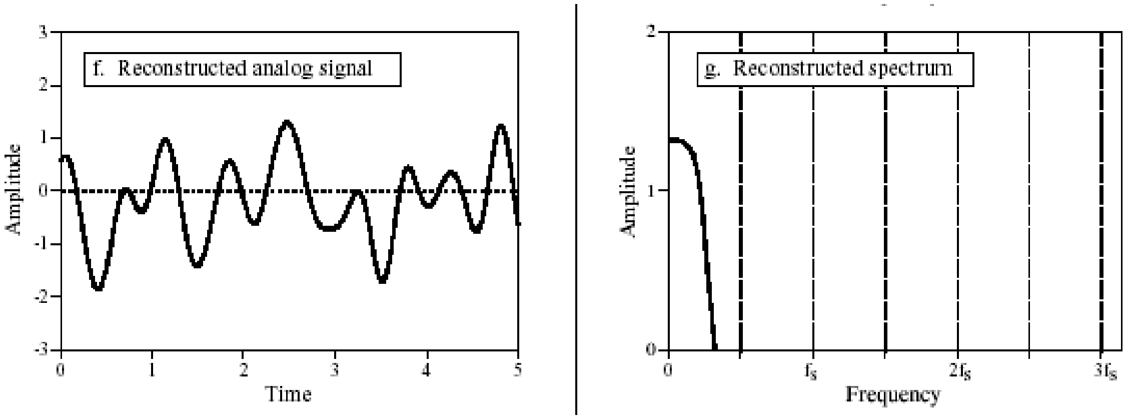


FIGURE 3-4

Analysis of digital-to-analog conversion. In (a), the digital data are converted into an impulse train, with the spectrum in (b). This is changed into the reconstructed signal, (f), by using an electronic low-pass filter to remove frequencies above one-half the sampling rate [compare (b) and (g)]. However, most electronic DACs create a zeroth-order hold waveform, (c), instead of an impulse train. The spectrum of the zeroth-order hold is equal to the spectrum of the impulse train multiplied by the sinc function shown in (d). To convert the zeroth-order hold into the reconstructed signal, the analog filter must remove all frequencies above the Nyquist rate, *and* correct for the sinc, as shown in (e).

3.2.5.3 **Shannon-Hartley Theorem**

The Shannon-Hartley theorem ties together the preceding concepts into an equation providing the maximum data rate limitation for a channel based on baseband bandwidth limitations. Let *C* represent the *maximum channel capacity* in bits/second (bits/s) the maximum channel capacity is given by

$$C = 2B\log_2 M \qquad (3\text{-}1)$$

Strictly speaking, this equation ignores any effects of noise and assumes that all *M* levels are discernible at the receiver. In practice, the presence of noise may be such that the number of levels must be reduced, in which case the channel capacity predicted by (3-1) may be unrealizable. The next section will provide further information on this subject. Note that for basic binary transmission

$$C = 2B \qquad (3\text{-}2)$$

Thus, for basic binary baseband transmission, the channel capacity in bits per second is twice the bandwidth.

3.2.5.4 Shannon Limit

When random noise exists on a communications channel, errors with digital transmission may occur in two ways. (1) A positive spike of noise may add to the signal and cause the transmitted level to be interpreted as a higher level. (2) A negative spike may subtract from the signal and cause the transmitted level to be interpreted as a lower level. The maximum data rate that can theoretically be transmitted without error is based on the signal-to-noise ratio and the bandwidth and is called the *Shannon limit*. Expressed as a channel capacity *C* in bits/second, it is given by

$$C = B \log 2 [1 + (S/N)] \qquad (3\text{-}3)$$

Note that it is the **absolute signal-to-noise ratio** used in (3-3) and **not the decibel form.** Between the Shannon-Hartley law of (3-2) and the Shannon limit of (3-3), there are two ways of characterizing the baseband channel capacity. The Shannon-Hartley law provides a bound based on bandwidth and the number of levels, which would be the limit in a noise-free environment. However, the Shannon limit places an upper bound based on the signal-to-noise ratio. In practice, ***the result providing the smallest value of channel capacity is the one that establishes the limit.***

The Shannon-Hartley law and the Shannon limit should be interpreted more as theoretical or even "philosophical" concepts rather than practical design equations. They have been widely used in establishing operating bounds for information transmission but they are guidelines rather than exact operating constraints.

3.2.6 Filters:

Before encountering the analog-to-digital converter, the input signal is processed with an electronic low-pass filter to remove all frequencies above the Nyquist frequency (one-half the sampling rate). This is done to prevent aliasing during sampling, and is correspondingly called an **antialias filter**. On the other end, the digitized signal is passed through a digital-to-analog converter and another low-pass filter set to the Nyquist frequency. This output filter is called a **reconstruction filter**, and may include the previously described zeroth-order-hold frequency boost. Unfortunately, there is a serious problem with this simple model: the limitations of electronic filters can be as bad as the problems they are trying to prevent.

Three types of analog filters are commonly used: **Chebyshev**, **Butterworth**, and **Bessel** (also called a Thompson filter). Each of these is designed to optimize a different performance parameter. The complexity of each filter can be adjusted by selecting the number of **poles** and **zeros**. The more poles in a filter, the more electronics it requires, and the better it performs.

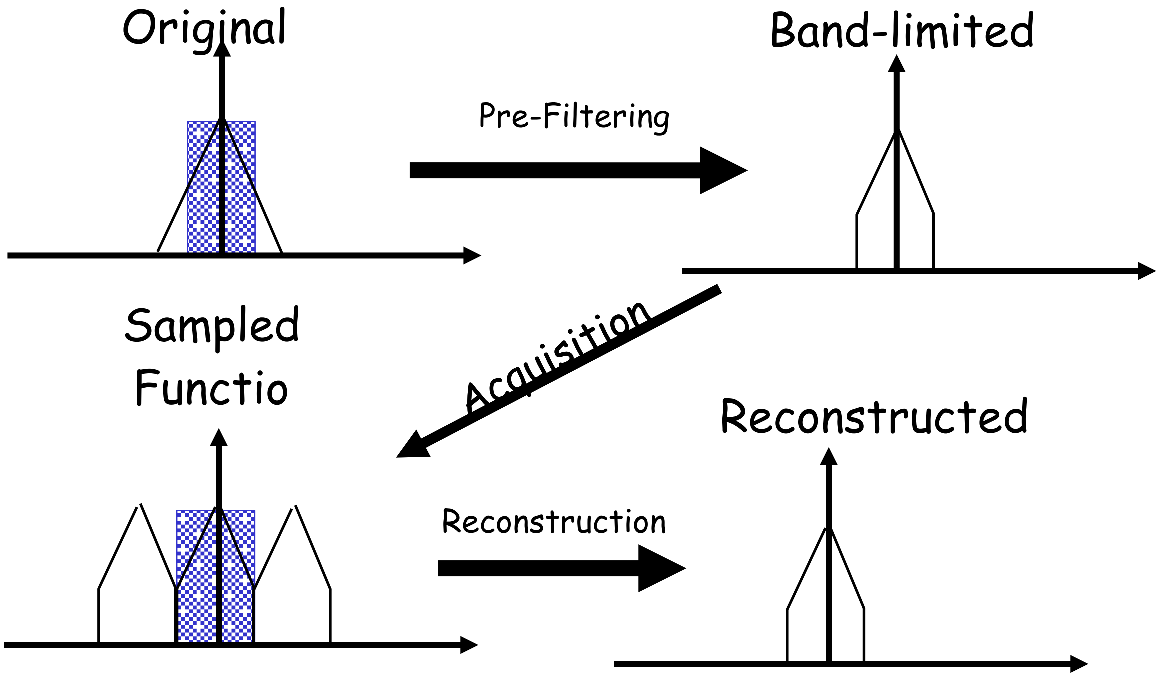


Figure 3.5

3.2.7 Shannon Sampling Theorems

A continuous-time signal x(t) with frequencies no higher than fmax can be reconstructed exactly from its samples x[n] = x(T_sn) if the samples are taken at a rate fs which is greater than 2fmax.

The quantity 2fmax is called the Nyquist rate. The Nyquist frequency is fs/2. The Sampling Theorem makes several ideal assumptions:

1. The continuous-time signal has no frequency content above the frequency fmax.
2. The sampling time is exactly the same between any two samples.
3. The sequence of numbers obtained by sampling is represented in exact precision.

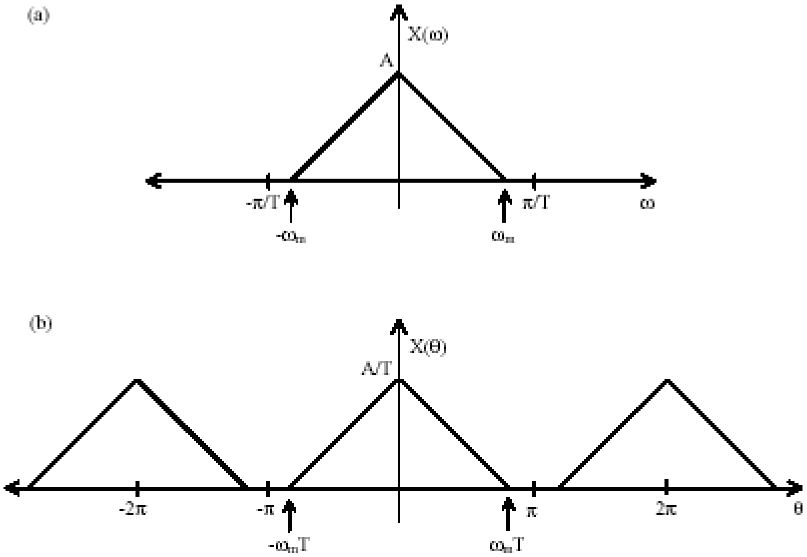


Figure 3.6: Sampling of a band limited signal above the Nyquist rate: (a) Fourier transform of the continuous-time signal; (b) Fourier transform of the sampled signal.

3.2.8 Aliasing

Aliasing is caused by sampling at a rate lower than that of the Nyquist frequency for a given signal. Now let us consider the case of a band limited signal when it is sampled at a lower rate than the Nyquist frequency. Figure 3.7(a) illustrates the case of sampling at rate lower than that of Nyquist. Here, the shape of the Fourier transform of the sampled signal in the range [Pi ±w(n), -Pi ±w(n)] becomes distorted. Distortion occurs in this frequency range because two adjacent replicas overlap and their superposition give rise to the shape illustrated by Figure 3.7(b).

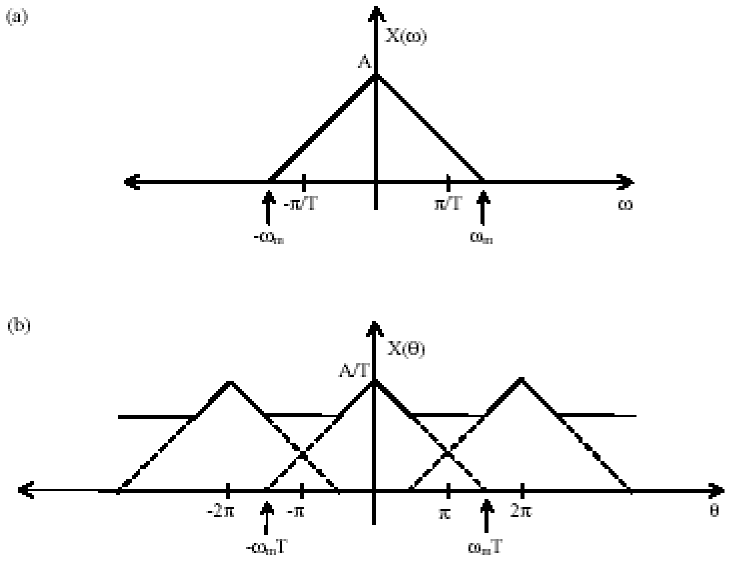


Figure 3.7

3.2.8.1 Anti-alias Filter Requirements

A common practice in the sampling of a continuous–time signal is to filter the signal before it is passed to the sampler. The filter used for this purpose is usually an analogue low-pass filter whose cut-off frequency is not larger than half the sampling rate. Such a filter is called an *anti-aliasing* filter. In addition, a further consideration in the design of the ant aliasing filter is to implement a so-called *Guard Band*.

3.2.8.2 Over sampling and noise shaping

Oversampling enables:

1. Sharp cutoff anti aliasing filtering
2. Reduction in the number of bits in quantizer with the same SNR.

3.3 **Quantization** is a process which maps the continuous range of signal amplitudes to a number of discrete values. As most of the signals are continuous in time and amplitude they signal must be digitized in both time and amplitude to be represented in binary form. It is done by

1. Discrete in time by sampling at Nyquist rate
2. Discrete in amplitude by quantization

Once samples have been captured, they must be made discrete in amplitude. So the Quantization :-

1. Converts actual sample values (usually voltage measurements) into an integer approximation
2. Rounds off a continuous value so that it can be represented by a fixed number of binary digits

3.3.1 **Drawbacks**

By quantization we are introducing an error. This error which is introduced in the beginning of the digital processing may have devastating effects. So there is a tradeoff between number of bits required and error. More the number of bits lesser is the error. We can tolerate some amount of error because

1. Human perception limitations affect allowable error.
2. Specific application affects allowable error.

3.3.2 **Quantization Process**

There are two approaches generally followed :-

1. Rounding the sample to the closest integer.(e.g. round 3.14 to 3)
2. Create a Quantizer table that generates a staircase pattern of values based on a step size.

Figure 3.9 illustrates the process. We are considering an audio signal with a voltage range between -10 and +10. Assume that the audio waveform has already been time sampled, as shown

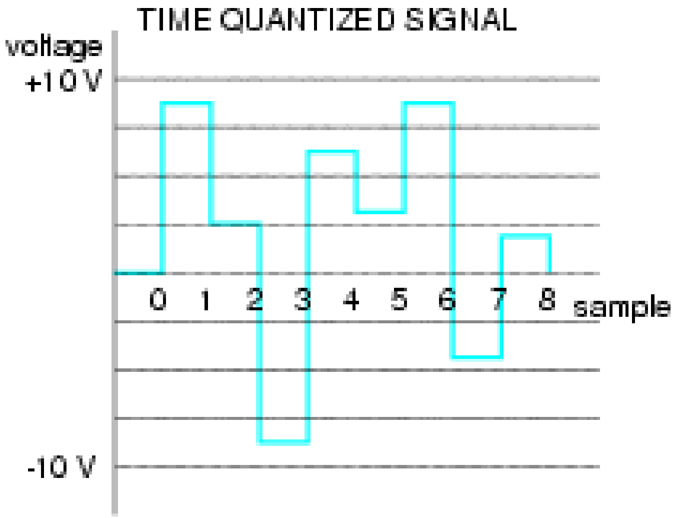


Figure 3.8

We have chosen to represent each sample by 4 bits. There are an infinite number of voltages between -10 and 10. We will have to assign a range of voltages to each 4-bit codeword. As we can represent 16 levels with the help of 4 bits so we will have 16 quantization levels.

3.3.3 <u>Types of Quantizers</u>

1. Uniform Quantizer
2. Non Uniform Quantizer
3. A-Law
4. U - Law

3.4 **Signal Reconstruction** So far we have discussed Analog-to-Digital Converter (ADC) which provides us the sampled and quantized binary code. When we move from digital to analog domain we need another interface Digital-to-Analog Converter (DAC) which converts the quantized binary code back into an ***approximation*** of the analog signal by clocking the code to the same sample rate as the ADC conversion.

If a sampled signal $x[n]$ has been obtained from a band-limited signal $x(t)$ by sampling at the Nyquist rate (or higher), the signal $x(t)$ can be perfectly reconstructed using the formula:

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT)\operatorname{sinc}\left(\frac{t-nT}{T}\right)$$

However this represents a filter which has an infinite response and is therefore non-causal.

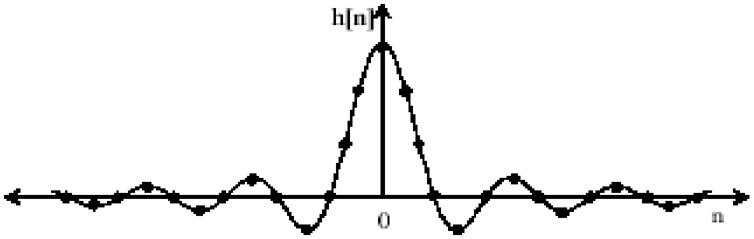


Figure 3.9

In practice, therefore, it is usual to use the “zero-order hold” reconstruction filter which has the formula:

$$h_{zoh}(t) = \begin{cases} 1, & 0 \le t < T \\ 0, & otherwise \end{cases}$$

which has an impulse response and overall response as shown:

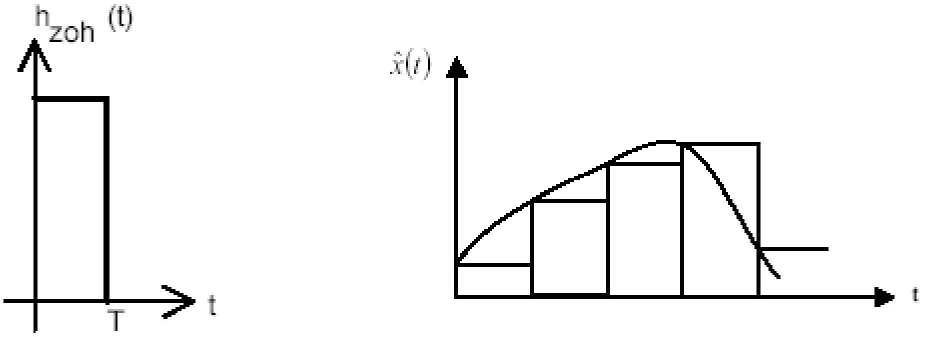


Figure 3.10

So we can completely reconstruct or determine *g(t)* from G*(w)* using the Inverse Fourier Transform. In order to do this, we need to remove all of the shifted copies of *G(w)*t. This is done by simply multiplying *G(w)* by a box function of width 2fs.

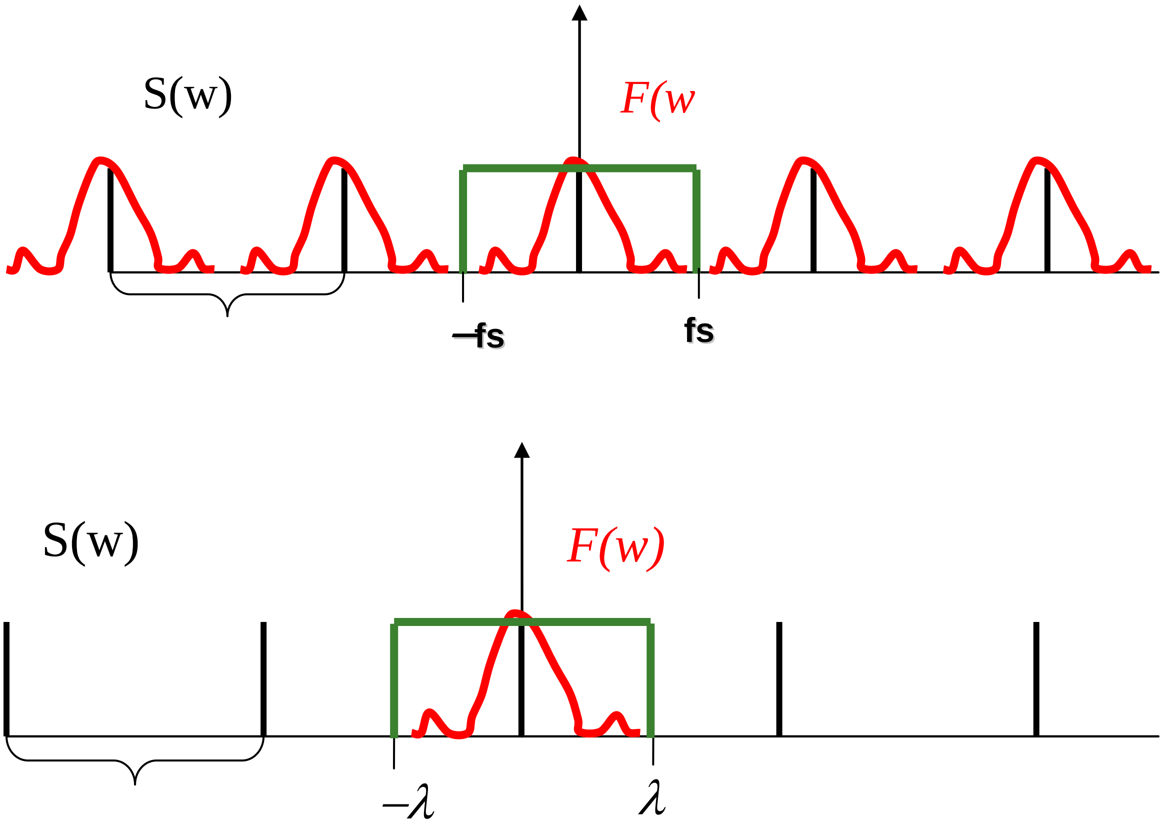


Figure 3.11

Mathematically:

g(t)*h(t) = (Σg[i])*h(t)

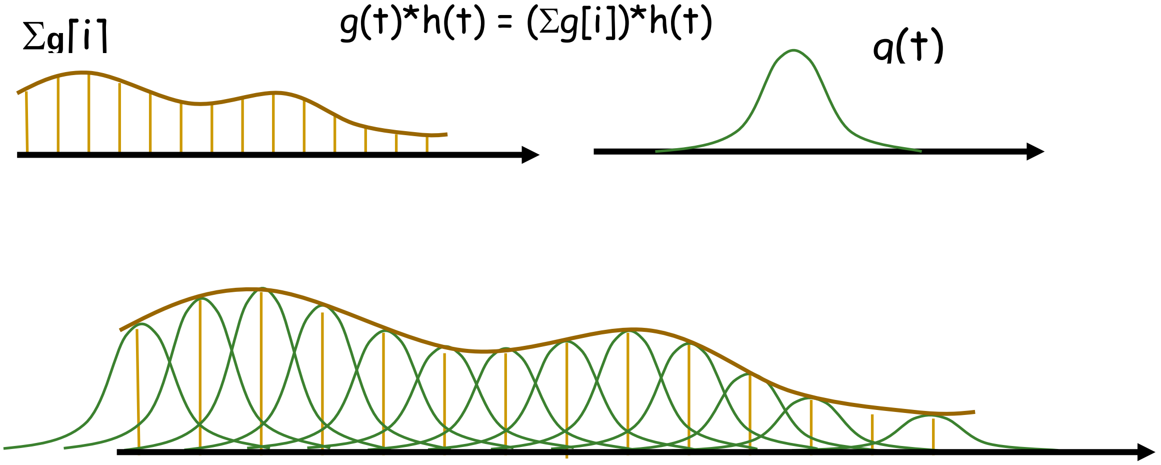


Figure 3.12

3.5 **Problems in ADCs / DACs_** There are many problems which we face while moving from analog to digital world and back. Few are illustrated below :-

3.5.1 **Errors in ADCs** Quantization errors occur in all ADCs, even in so called ideal ones. It arises because the analog input signal, which can take any value, is rounded to a finite number of output levels. The only way to decrease the quantization error is to increase the resolution. Quantization is deterministic, but since the input normally consists of complicated signals and noise, the error is approximately random and has a noise-like behavior.

3.5.2 **Jitter**

Another important error is jitter, i.e. uncertainty in sample time. This can be considered a random error and consequently it will lead to an increase in noise level. This is related to frequency. If the input frequency is doubled, the error is doubled. That is equivalent to a 6 dB decrease in SNR. This is illustrated in figure 3.13.

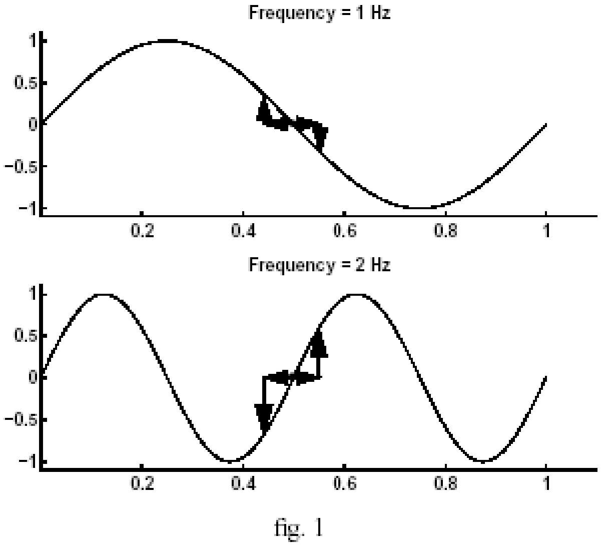


Figure 3.13

3.5.3 **Mismatch** The accuracy of a system is proportional to the matching accuracy. Mismatch, i.e. gradients on a chip, will of course decrease performance. Accuracy is improved with deeper submicron technologies and it is also possible to reduce mismatch to a certain degree by using design methods that split up devices and place the parts in such a way that it minimizes the effect of gradients.

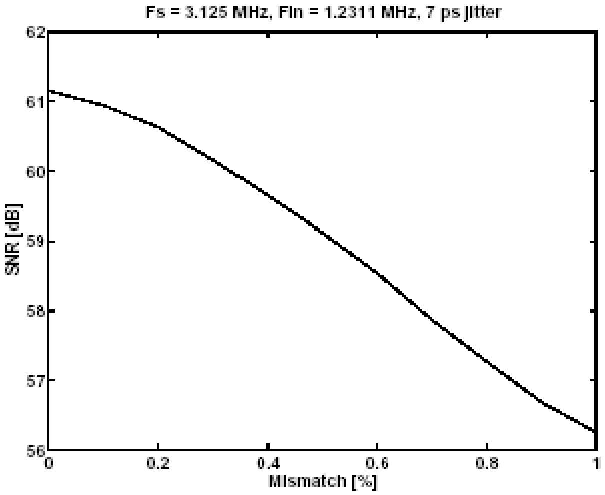


Figure 3.14

3.6 **Multirate Systems.** The next level of sophistication involves multirate techniques, using more than one sampling rate in the same system. It works like this for the digital voice recorder example. First, pass the voice signal through a simple RC low-pass filter and sample the data at 64 kHz. The resulting digital data contains the desired voice band between 100 and 3000 hertz, but also has an unusable band between 3 kHz and 32 kHz. Second, remove these unusable frequencies in *software*, by using a *digital* low-pass filter at 3 kHz. Third, resample the digital signal from 64 kHz to 8 kHz by simply discarding every seven out of eight samples, a procedure called **decimation.** The resulting digital data is equivalent to that produced by aggressive analog filtering and direct 8 kHz sampling.

Multirate techniques can also be used in the output portion of our example. The 8 kHz data is pulled from memory and converted to a 64 kHz sampling rate, a procedure called **interpolation**. This involves placing seven samples, with a value of zero, between each of the samples obtained from memory. The resulting signal is a digital *impulse train*, containing the desired voice band between 100 and 3000 hertz, plus spectral duplications between 3kHz and 32 kHz. Refer back to Figs. 3-6 a&b to understand why this it true. Everything above 3 kHz is then removed with a *digital* low-pass filter. After conversion to an analog signal through a DAC, a simple RC network is all that is required to produce the final voice signal.

Multirate data conversion is valuable for two reasons:

1. It replaces analog components with software, a clear economic advantage in massproduced products.

2. It can achieve higher levels of performance in critical applications. For example, compact disc audio systems use techniques of this type to achieve the best possible sound quality. This increased performance is a result of replacing analog components (1%precision), with digital algorithms (0.0001% precision from round-off error). Digital filters outperform analog filters by *hundreds of times* in key areas.

3.6 **ADC Used in our Design_**(Oversampled Sigma-Delta A/D Converter)

The Oversampled Sigma-Delta A/D Converter is a noise-shaping quantizer. The main purpose of noise-shaping is to reshape the spectrum of quantization noise so that most of the noise is filtered out of the relevant frequency band (say the audio band for speech applications). The main objective here is to trade off bits for samples - that is, increasing the sampling rate but reducing the number of bits per sample. The resulting increase in quantization noise is compensated by a noise shaping quantizer. This quantizer pushes the added quantization noise out of the relevant frequency band and thereby preserves a desired level of signal quality. This reduction in the number of bits simplifies the structure of A/D and D/A converters.

The analog input is prefiltered by an antialiasing prefilter whose structure is simplified (order 5) because of oversampling. The input signal is oversampled by a factor of 64 and a two-level ADC (1-bit quantizer) is used. The Zero-Order Hold converts it back to analog form, which is then subtracted from the analog input. The difference signal, which is the "delta" part, is accumulated into the integrator, which is the "sigma" part, and provides a local average of the input. The feedback loop (approximation loop) causes the quantization noise generated by the ADC to be highpass filtered, pushing its energy towards the higher frequencies (64*fs/2) and away from the relevant signal band.

The multistage decimation stage (4*4*4) reduces the sampling rate back to 8 KHz. During this process, it removes the high frequency quantization noise that was introduced by the feedback loop and removes any undesired frequency components beyond fs/2 (4 KHz) that were not removed by the simple analog prefilter.

SOURCE CODING

4.1 INTRODUCTION

Speech coding has been and still is a major issue in the area of digital speech processing. Speech coding is the act of transforming the speech signal at hand, to a more compact form, which can then be transmitted with a considerably smaller memory. The motivation behind this is the fact that access to unlimited amount of bandwidth is not possible. Therefore, there is a need to code and compress speech signals. Speech compression is required in long-distance communication, high-quality speech storage, and message encryption. For example, in digital cellular technology many users need to share the same frequency bandwidth. Utilizing speech compression makes it possible for more users to share the available system. Another example where speech compression is needed is in digital voice storage. For a fixed amount of available memory, compression makes it possible to store longer messages [7].

Speech coding is a lossy type of coding, which means that the output signal does not exactly sound like the input. The input and the output signal could be distinguished to be different. Coding of audio however, is a different kind of problem than speech coding. Audio coding tries to code the audio in a perceptually lossless way. This means that even though the input and output signals are not mathematically equivalent, the sound at the output is the same as the input. This type of coding is used in applications for audio storage, broadcasting, and Internet streaming [8].

The recent exponential growth of wireless telecommunications today drives all aspects of wireless tech-neology to higher degrees than ever before. As the Information Age transitions from the medium of physical wires to the airwaves, economical, physical, and practical barriers drive many areas of research which are all ultimately intertwined in a single objective: to transfer maximal amount of information consuming minimal relevant resources. Due to the impact on parameters such as bandwidth requirements and conversation quality, it can be argued that the most important component of any telephony system is that which generates the digital representation of the speech [9].

In mobile communication systems, since service providers are continuously met with the challenge of accommodating more users within a limited allocated bandwidth, speech coders that provide toll quality speech at low bit rates are needed. The goal of all speech coding systems is to transmit speech with the highest possible quality using the least possible channel capacity. In general, there is a positive correlation between coder bit-rate efficiency and the algorithmic complexity required to achieve it. The more complex an algorithm is, the more is processing delay and cost of implementation.

4.2 BACKGROUND

In the most basic form, speech coding is simply the digital representation of the analog speech signal. The brute force method to preserve quality of speech typically requires a 12 or 13 bit linear quantization. The two coding techniques which form the basis for most speech coding algorithms today, A-law and u-law pulse code modulation (PCM), retain the same quality represented with only 8 bits using logarithmic quantization. These techniques typically form the parametric baseline used for all other comparisons, especially the data rate. At a sampling rate of 8 kHz, the data rate for both A-law and u-law is 64 kbps, and many algorithms today assume the input for the algorithm is already in this format. Compression is improved by exploiting properties of speech signals, which are bandwidth limited and relatively coherent from one sample to the next. Both ADPCM and CELP leverage from these characteristics. There are essentially two approaches to speech coding: waveform coding and vo-coding. ADPCM falls into the category of waveform coding, which attempts to preserve the waveform of the original speech signal. Speech coders based on CELP are vo-coders, a technique that attempts to model the generation of speech. The quality of waveform coders can be compared by direct comparison between the decoded waveform and the original waveform. Since vo-coders do not attempt to preserve the waveform, this approach for quality comparison is not applicable. For this reason, the quality of most speech coding techniques is compared by a measure of user perception. The mean opinion score, MOS, is a number between 0 and 5 which represents the average quality as judged by listeners, 0 being the worst. A score of 4 is labelled ‘toll quality’ and is the quality of signal perceived by users in a typical long-distance call. Two other characteristics often used to compare coding schemes are complexity and delay. Complexity, commonly measured in MIPS often relates to system implementation requirements. Delay is the time required for an input signal to be coded and reconstructed at the receiver and is important due to its impact on conversation quality.

4.2.1 METHODOLOGY ANALYSED

There are several different methods to successfully accomplish speech coding. Some main categories of speech coders are LPC Vo-coders, Waveform and Sub band coders. Linear Predictive Coding is one possible technique of analyzing and synthesizing human speech. LPC method has been used for a long time. Texas Instruments had developed a monolithic PMOS speech synthesizer integrated circuit as early as 1978. This marked the first time the human vocal tract had been electronically duplicated on a single chip of silicon [11]. This one of the first speech synthesizer used LPC to accomplish successful synthesis. LPC makes coding at low bit rates possible. For LPC-10, the bit rate is about 2.4 kbps. Even though this method results in an artificial sounding speech, it is intelligible. This method has found extensive use in military applications, where a high quality speech is not as important as a low bit rate to allow for heavy encryption of secret data. However, since a high quality sounding speech is required in the commercial market, engineers are faced with using other techniques that normally use higher bit rates and result in higher quality output. In LPC-10 vocal tract is represented as a time-varying filter and speech is windowed about every 20 ms, for each frame, the gain and only 10 of the coefficients of a linear prediction filter are coded for analysis and decoded for synthesis. In 1996, LPC-10 was replaced by mixed-excitation linear prediction (MELP) coder to be the United States Federal Standard for coding at 2.4 kbps. this MELP coder is an improvement to the LPC method, with some additional features that have mixed excitation, aperiodic pulses, adaptive spectral enhancement and pulse dispersion filtering as mentioned above [10].

Waveform coders on the other hand, are concerned with the production of a reconstructed signal whose waveform is as close as possible to the original signal, without any information about how the signal to be coded was generated. Therefore, in theory, this type of coders should be input signal independent and work for both speech and non-speech input signals [10]. Waveform coders produce a good quality of speech signals above bit rates of 16 kbps. However, if the bit rate is decreased below 16 kpbs, the quality deteriorates quickly. One form of waveform coding is Pulse Code Modulation (PCM). This type of waveform coding involves sampling and quantizing the input signal. PCM is a memory less coding algorithm as mentioned above [10]. Another type of PCM is Differential Pulse Code Modulation (DPCM). This method quantizes the difference between the original and the predicted signals. This method involves prediction of the next sample from the previous samples. This is possible since there is a correlation in speech samples because of the effects of the vocal tract and the vibrations in the vocal cords [12]. It is possible to improve the predictor as well as the quantizer in DPCM if they are made adaptive, in order to match the characteristics of the speech that is to be coded. This type of coders is called Adaptive Differential Pulse Code Modulation (ADPCM).

One other type of speech coders is called the Subband coders. This type of coding involves filter bank analysis to be undertaken in order to filter the input signal into several frequency bands. Bit allocation is done to each band by a certain criterion [10]. Presently however, Subband coders are not widely used for speech coding. It is very difficult to create high quality speech by using low bit rates with this technique. As suggested in [10], Subband coding is mostly utilized in the medium to high bit rate applications of speech coding.

4.2.1.1 ADPCM ADPCM, adaptive differential pulse code modulation, is one of the most widely used speech compression techniques used today due to its high speech quality, low delay, and moderate complexity [7]. ADPCM is a waveform coder, and achieves its compression improvements by taking advantage of the high correlation exhibited by successive speech samples. Referring to Figure 1 below, the ADPCM encoder first calculates the difference between the input signal, typically A-law or *u*-law, and a signal generated by a linear adaptive predictor. Due to the high correlation between samples, the difference signal has a much lower dynamic range than the input signal [7]. The reduced dynamic range means fewer bits are required to accurately represent the difference signal as compared to the input. Transmitting only the quantized difference signal reduces the transmitted data rate. A four-bit quantization translates to a 32 kbps data rate for a compression ratio of 2:1. Higher quality is achieved through the adaptive nature of the quantization. Analyzing the time varying characteristics of the difference signal, the size of the quantization steps and the rate at which quantization steps change facilitate higher accuracy for a wider dynamic range. As mentioned, signal received at the decoder is the quantized difference signal. The ADPCM decoder is essentially the reverse process of the encoder. More than one standard exists for ADPCM. The International Telegraph and Telephone Consultative Committee, CCITT first adopted recommendation G.721 that defines standard ADPCM operating at 32 kbps using 4 bit quantization. This standard was eventually encompassed by recommendation G.726, which expanded the original definition to include data rates of 40, 32, 24, and 16 kbps. This relates to 5, 4, 3, and 2-bit quantization of the difference signal [8]. A third recommendation was also adopted, G.727, addressing 5, 4, 3, and 2 bit sample embedded ADPCM which is effectively a variable rate definition ofG.726 [9].

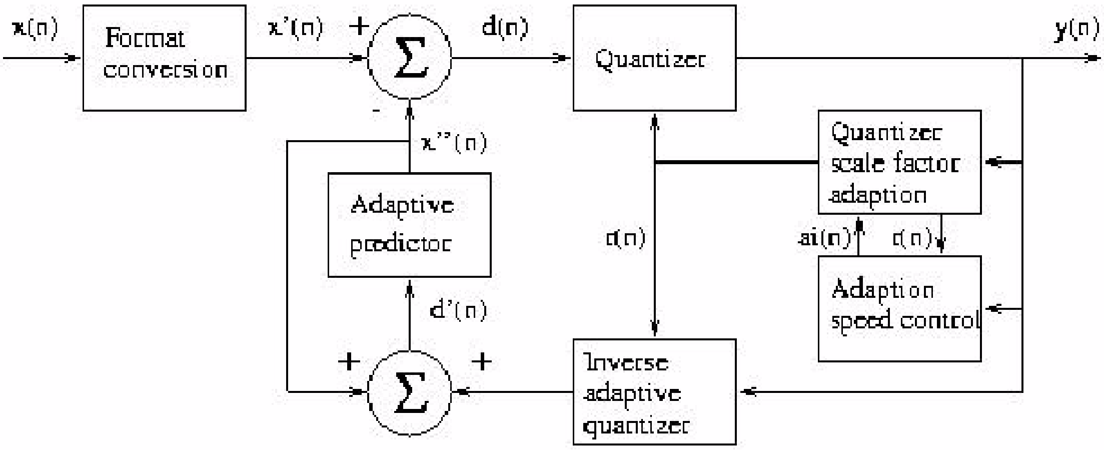


Figure 4.1 ADPCM ENCODER

As most cellular applications demand a speech coder that can operate at under 8 kbps, even at 2-bit quantization ADPCM is considered to have high bandwidth demands. The relatively low complexity of ADPCM makes it attractive for systems, which require waveform preservation or have relaxed bandwidth restrictions. The latter of these fits cordless telephony and microcell applications well. The G.721 ADPCM has been specified in a few popular cordless standards. Great Britain introduced the CT2 system to address second generation cordless applications [10]. The DECT, Digital European Cordless Telephony, system also covers second generation cordless, facilitating compatible systems across Europe [10]. The third generation Personal Access Communication System, PACS, also specifies voice coding using 32 kbps ADPCM [10].

4.2.1.2 **CELP** In order to maintain acceptable quality below the 8 kbps data rate, a fundamentally different approach to speech coding and sizeable jump in complexity is encountered. The most widely documented scheme for speech coding operating at under 8 kbps is CELP, code excited linear prediction. Cited as early as 1982, CELP actually represents a class of vocoders, all computationally expensive, which are based on the analysis and synthesis of incoming speech. The scope of this project limits the focus on one variation of CELP, and for the purpose of example and demonstration, this project focuses on the U.S. Federal Standard 1016 that defines a CELP algorithm operating at 4.8 kbps. Opposed to waveform encoding, the goal of CELP algorithms is to code and transmit the minimal amount of information necessary to *synthesize* speech which is audibly *perceived* to be accurate. This is achieved by modeling aspects of the speech generation process, effectively modeling the vocal tract. Two components are generally found in these models, one which models the long term speech structure, the pitch, and another, which models the short-term speech structure, the formant. Relying on human perception as the measure of quality makes measurements of SNR irrelevant, and the overall rating of a specific implementation is in terms of mean opinion score, or MOS. The technique of incorporating synthesis as a fundamental part of signal analysis is common to CELP algorithms and is why they are often referred to as analysis-by-synthesis techniques. Another departure from ADPCM, CELP algorithms work on frames of sampled input speech. The length and duration of the frame depends on the specific CELP algorithm.

USFS-1016 operates on 30 ms frames which are furthure split into four 7.5 ms sub frames. Termed vector quantization, the frames of sampled input speech, input vectors, are compared to an existing set of excitation vectors called the code book. The search for the code book entry which produces the closest match is often an exhaustive search, making this class of algorithms computationally expensive. Each code book entry is applied to an LPC filter to generate a synthesized speech vector.

The synthesized sample is compared to with the input vector and rated with a perceptual weighing filter. The index of the code book with the vector producing the closest perceptual match to the input vector is selected and transmitted.

Low bit rates are achieved because only the code book indices, LPC filter coefficients, and gain information are transmitted. Figure 2 below, shows the block diagram for the USFS-1016 CELP encoder. Two code books and a 10th order LPC filter are used in USFS-1016. The 256 entry adaptive code book models the long-term speech structure, and the 10th order LPC filter models the short term speech structure. The residual signal remaining after adaptive code book quantization and LPC synthesis is modeled by the static 512 entry stochastic code book. The encoder transmits the index and gain for both stochastic and adaptive code books each subframe, or four times per 30 ms frame. The LPC filter coefficients are transmitted only once per frame. Modeling the long-term speech by the 256 entry adaptive code book improves quality at the cost of additional computational load of the same complexity as the 512 entry stochastic code book look-up. At the expense of quality, bit rates below 4.8 kbps can be achieved by limiting the search of the stochastic code book.

Disadvantages associated with USFS-1016 CELP include the less than toll quality speech (3.2 MOS), and high processing delays (90 ms). The frame-by-frame analysis, coupled with the high processing demands introduces delay which can degrade quality of conversation and introduce difficulties in related speech processing components, such as echo canceling. Other standards have been introduced which addressed both of these issues. Important standards often referenced in current literature include CCITT recommendation G.728 for LD-CELP, or low delay CELP, and G.729 for CS-ACELP, conjugate structure algebraic CELP. Both offer quality improvements but have higher bit rates (16 kbps and 8 kbps respectively). Less complex than LD-CELP, CS-ACELP operates at half the bit rate and offers toll quality speech. Yet another algorithm, VSELP, is defined in the IS-54 standard [10].

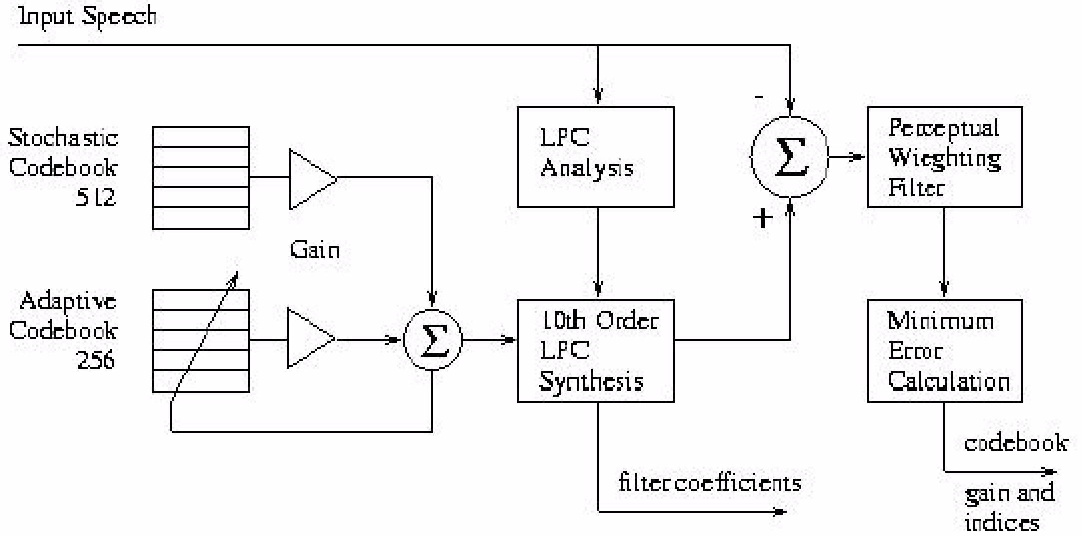
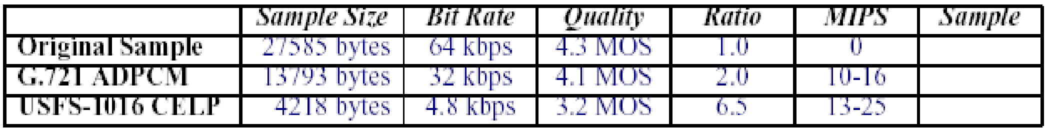


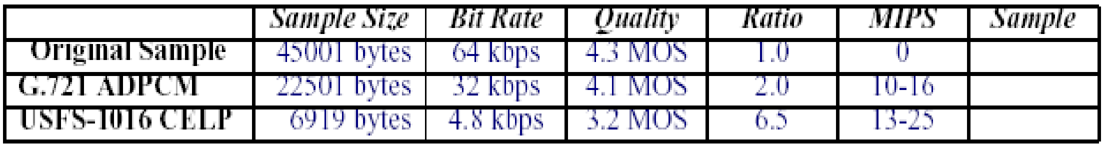
Figure 4.2 USFS-1016 CELP ENCODER

4.2.1.3 **RESULT** The results presented here are the products of samples processed by software implementations of G.721 [11] and USFS-1016 [12]. Since speech widely varies between age, gender, and ethnicity, etc., speech encoders must adequately support a wide range of speech characteristics. Tables 1 and 2 below present data and processed samples for typical male and female speech respectively. The quantitative data is extracted from the processed sample sizes and data presented in. Figure 3 below demonstrates the difference between waveform coding and vocoding. The samples are fragments from the male speech sample.



	Sample Size	*Bit Rate*	*Quality*	*Ratio*	*MIPS*	*Sample*
Original Sample	27585 bytes	64 kbps	4.3 MOS	1.0	0	
G.721 ADPCM	13793 bytes	32 kbps	4.1 MOS	2.0	10-16	
USFS-1016 CELP	4218 bytes	4.8 kbps	3.2 MOS	6.5	13-25	

Table 1. Demonstration of Male Speech.



	Sample Size	*Bit Rate*	*Quality*	*Ratio*	*MIPS*	*Sample*
Original Sample	45001 bytes	64 kbps	4.3 MOS	1.0	0	
G.721 ADPCM	22501 bytes	32 kbps	4.1 MOS	2.0	10-16	
USFS-1016 CELP	6919 bytes	4.8 kbps	3.2 MOS	6.5	13-25	

Table 2. Demonstration of Female Speech

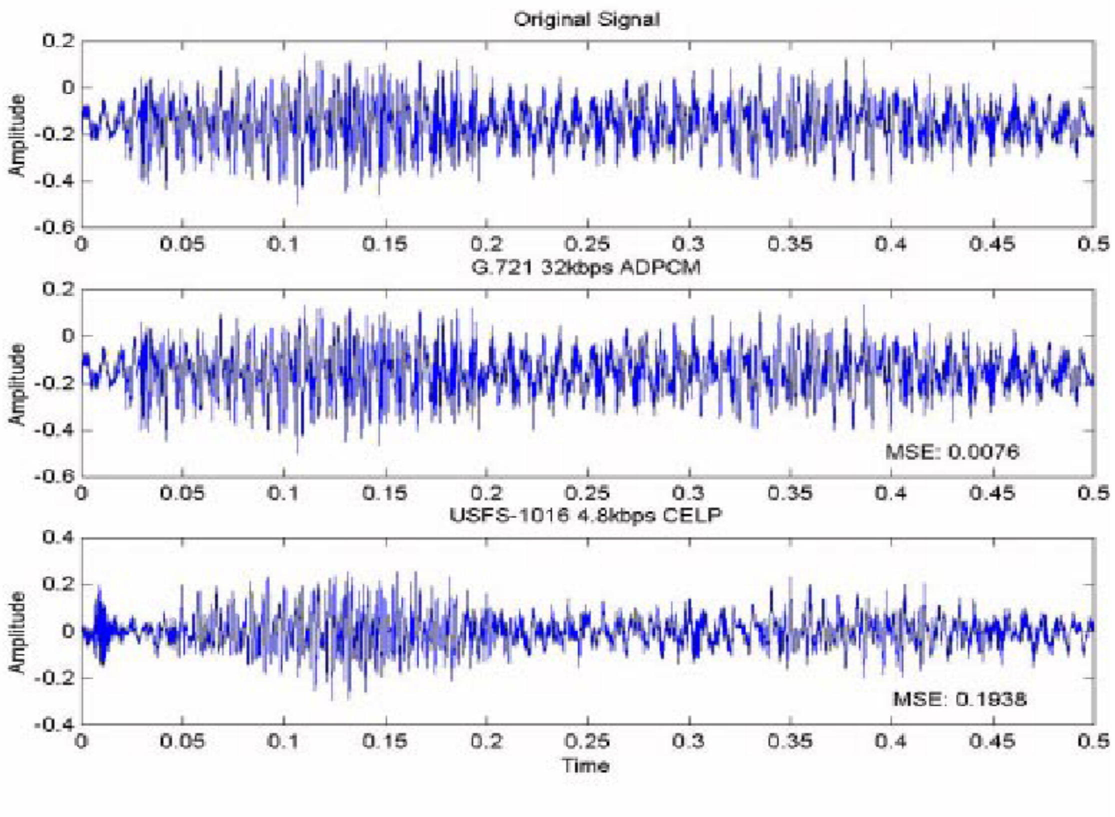


Figure 3. Waveforms resulting from ADPCM and CELP encode & decode.

It is evident, from visual comparison, the difference between waveform coding and vocoding. The mean square error is calculated for each algorithm against the original signal. Note that the standard deviation if the original speech sample is 0.0747.

4.2.2 COMPARISON OF SPEECH CODING ALGORITHMS

4.2.2.1 <u>ADAPTIVE DIFFERENTIAL PUSLE CODE MODULATION (ADPCM)</u> Differential coding refers to coding the difference between two signals rather than the signals themselves. In differential coding, the short-term redundancy of the speech waveform is removed as much as possible. This is accomplished by forming an *error (difference) signal* by subtracting an estimate of the signal from the original signal. The estimate is generally obtained by a linear predictor that estimates the current samples from a linear combination of one or more past samples. The main source of performance improvement for DPCM coders is the reduced dynamic range of the quantizer input signal. Since the quantization noise is proportional to the step size, a signal with a smaller dynamic range can be coded more accurately with a given number of quantization levels.

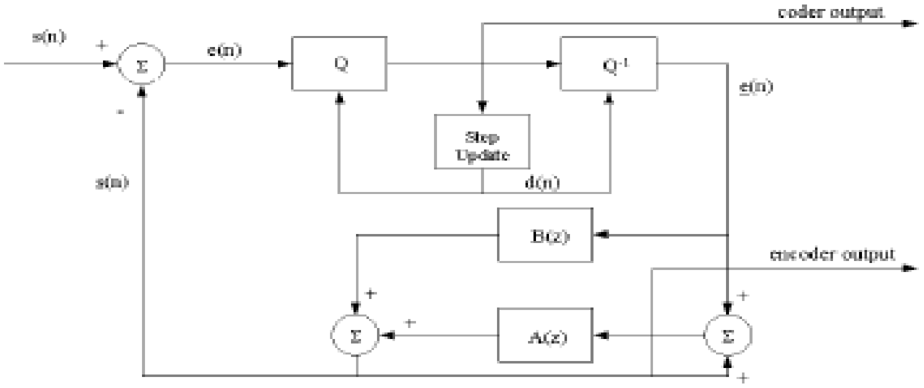


Figure 4. 2 ADPCM ENCODER

The basic issues in differential coding are the predictor properties and the quantization of the prediction error signal. DPCM with adaptation of the quantizer is known as Adaptive DPCM (ADPCM) as shown in Figure 1. The ADPCM algorithm is conceptually similar to DPCM but more sophisticated in that it uses an eight-order predictor, adaptive quantization, and adaptive prediction. Furthermore, the algorithm is designed to recognize the difference between voice or data signals and use a fast or slow quantizer adaptation mode, respectively. ADPCM provides greater levels of prediction gain than simple DPCM depending on the sophistication of the adaptation logic and the number of past samples used to predict the next sample. The prediction gain of ADPCM is ultimately limited by the fact that only a few past samples are used to predict the input and the adaptation logic only adapts the quantizer – not the prediction weighting coefficients.

4.2.2.2ANALYSIS-BY-SYNTHESIS LINEAR PREDICTIVE CODERS

Analysis-by-synthesis speech coders, which include such important classes of coders as *code excited linear predictive* (CELP) coders, are among the newest and most effective of modern speech coders. Note that the analysis-by-synthesis LPC is essentially a hybrid coder in the sense that it combines the features of model-based vocoders, by representing the formant and the pitch structure of speech and the properties of waveform coders by providing for the matching of the input speech waveform. In the following subsections, we describe CELP and *vector sum excited linear predictive* (VSELP) coders.

4.2.2.3 CODE EXICITED LINEAR PREDICTIVE CODER (CELP) Medium or low bit-rate speech coders have been researched for application to mobile radio communications. Code excited linear prediction (CELP) coding is one of the most effective coding methods at low bit-rates, which was proposed in the mid-eighties by and Schroeder and Atal [15]. CELP algorithm can produce low-rate coded speech comparable to that of medium-rate waveform coders thereby bridging the gap between waveform coders and vocoders. CELP (as shown in Figure 2) is an efficient closed loop analysis-synthesis method for narrow and medium band speech coding systems [4-16 kbps]. In CELP coders, speech is segmented into frames (typically 10-30 ms long) and for each frame an optimum set of linear prediction and pitch filter parameters are determined and quantized. Each speech frame is further divided into a number of subframes (typically 5 ms) and for each subframe an excitation codebook is searched to find the input vector to the quantized predictor system that gives the best reproduction of the speech signal [7]. For simplicity of design and performance robustness, the encoding of the LPC parameters is commonly achieved by the scalar quantization of the equivalent set of Reflection Coefficients, Log Area Ratios or Line Spectrum Pairs.

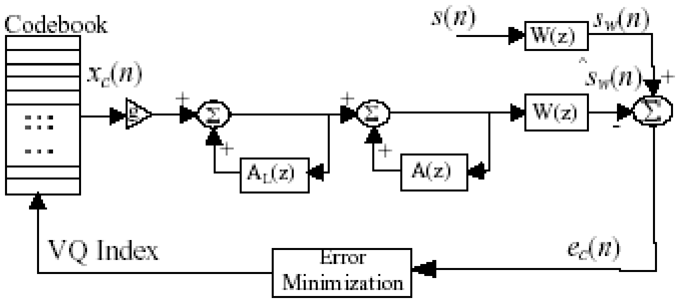


Figure 3.4 ANALYSIS-BY-SYNTHESIS CELP ENCODER

CELP coders operating at low bit rates (less than 1 bit/sample) are not able to adequately reproduce the high frequency components or the transients in speech signal. The reconstructed signal suffers from a degradation that is more noticeable for high-pitched voices with strong glottal excitation pulses such as female voice. This limitation is partly due to the size of the excitation code book and its population density and partly due to fixed frame length LPC analysis where one set of LPC parameters are used to describe spectral information about a frame of speech that may contain widely different speech events. In general the reproduction accuracy improves with increasing excitation codebook size and when the code book population is a better approximation to the speech excitation distribution.

4.2.2.4 VECTOR SUM EXICITED LINEAR PREDICITIVE CODER (VSELP)

CELP speech coders exhibit good performance at data rates as low as 4800 bps. The major drawback to CELP type coders is their large computational requirements. This problem motivated a great deal of work focussed upon developing structured codebooks and fast search procedures. Gerson and Jasiuk [14] proposed a vector sum excited linear predictive (VSELP) coder (as shown in Figure 3), which is associated with fast codebook search and robustness to channel errors, for use in digital cellular and mobile communications. An 8 kbps VSELP coder was selected by the Telecommunications Industry Association (TIA) as the standard for use in North American digital cellular telephone systems.

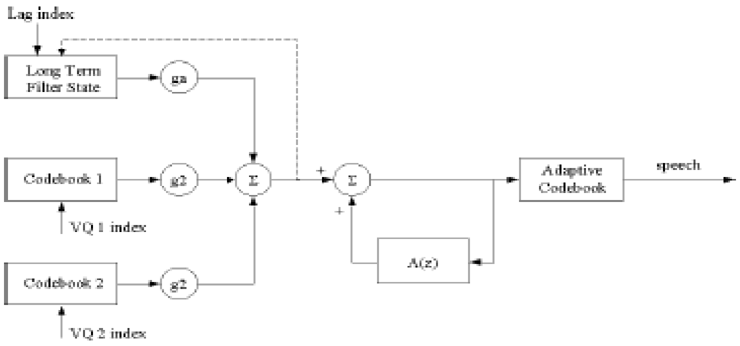


Figure 4. 4 ANALYSIS-BY-SYNTHESIS VSELP ENCODER

The code books in the VSELP encoders are organized with a predefined structure which significantly reduces the time required for the optimum code word search. The VSELP codec utilizes three excitation sources. The first is from the long term (“pitch”) predictor state (adaptive codebook). The second and third sources are from two VSELP excitation codebooks. Each of these VSELP codebooks contains the equivalent of 128 vectors. These three excitation sources are multiplied by their corresponding gain terms and summed. This becomes the combined excitation sequence ex(n). The synthesis filter is a direct form 10_{th} order LPC all-pole filter. The LPC coefficients are coded once per 20 ms frame and updated in each 5 ms subframe through interpolation. The excitation parameters are also updated in each 5 ms subframe. The number of samples in a subframe is 40 at an 8 kHz sampling rate. The implementation details of the algorithm, including codebook information, post filtering, etc. are presented in the IS-54 standard [13].

4.3 THEORY OF OPERATION

The ADPCM algorithm takes advantage of the high correlation between consecutive speech samples, which enables future sample values to be predicted. Instead of encoding the speech sample, ADPCM encodes the difference between a predicted sample and the speech sample. This method provides more efficient compression with a reduction in the number of bits per sample, yet preserves the overall quality of the speech signal. The implementation of the ADPCM algorithm provided in this application note is based on the Interactive Multimedia Association’s (IMA) Recommended Practices for Enhancing Digital Audio Compatibility in Multimedia Systems revision 3.00. The ITU (formerly CCITT) G.721 ADPCM algorithm is well known and has been implemented on many digital signal processors such as the TMS320 family from Texas Instruments and the ADSP-2100 family fromAnalog Devices. ITU G.721 uses floating point arithmetic and logarithmic functions which are not easily implemented in the 8-bit microcontroller world. The IMA Reference algorithm significantly reduces the mathematical complexity of ITU G.721 by simplifying many of the operations and using table lookups where appropriate.

4.3.1 COMPRESSION

The input, si , to the encoder routine must be 16-bit two’s complement speech data. The range of allowable values for si is 32767 to -32768. Figure 1 shows a block diagram for ADPCM compression and Appendix A has a listing of the function

ADPCM Encoder() . The predicted sample, sp, and the quantizer step size index are saved in a structure for the next iteration of the encoder. Initially, the quantizer step size index and the predicted sample (s p) are set to zero. The encoder function takes a 16-bit two's complement speech sample and returns an 8-bit number containing the 4-bit sign-magnitude ADPCM code.

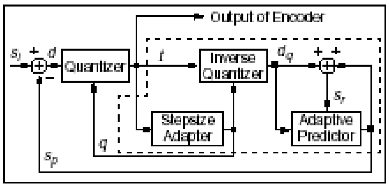


Figure 4. 5 ADPCM ENCODER BLOCK DIAGRAME

The predicted sample, sp, is subtracted from the linear input sample, si, to produce a difference, d. Adaptive quantization is performed on the difference, resulting in the 4-bit ADPCM value, t. The encoder and decoder both update their internal variables based on this ADPCM value. A full decoder is actually embedded within the encoder. This ensures that the encoder and decoder are synchronized without the need to send any additional data. The embedded decoder is shown within the dotted lines of Figure 1. The embedded decoder uses the ADPCM value to update the inverse quantizer, which produces a dequantized version, dq, of the difference, d. The implementation of the ADPCM algorithm presented in this application note uses a fixed predictor instead of an adaptive predictor which reduces the amount of data memory and instruction cycles required. The adaptive predictor of ITU G.721 adjusts according to the value of each input sample using a weighted average of the last six dequantized difference values and the last two predicted values. So at this point, the dequantized difference, dq, is added to the predicted sample, sp, to produce a new predicted sample, sr. Finally the new predicted sample, sr, is saved into sp. This function requires five 16-bit variables and two 8-bit variables. Some optimizations can be made to the code in Appendix A such as combining steps 7 and 8.

4.3.2 DECOMPRESSION

The input into the decoder, t, must be an 8-bit number containing the 4-bit ADPCM data in sign-magnitude format. The range of allowable values for t is 0 to 15, where 7 = 0x07 and -7 = 0x0F. Figure 2 shows a block diagram for ADPCM decompression and Appendix B has a listing of the function **ADPCMDecoder()**. The predicted sample, sp, and the quantizer step size index are saved in a structure for the next iteration of the decoder. Initially, the quantizer step size index and the predicted sample (sp) are set to zero. This function takes a 4-bit sign-magnitude ADPCM code and returns the 16-bit two’s complement speech sample.

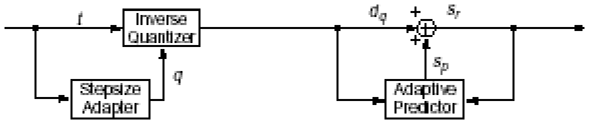


Figure 4. 6 ADPCM DECODER BLOCK DIAGRAME

This decoder is the same as the one used in the encoder routine. It uses the ADPCM value to update the inverse quantizer, which produces a difference, dq. The difference, dq, is added to the predicted sample, sp, to produce the output sample, sr. The output sample, sr, is then saved into the predicted sample, sp, for the next iteration of the decoder.

4.4 DIGITAL AUDIO DATA

The digital representation of audio data offers many advantages: high noise immunity, stability, and reproducibility. Audio in digital form also allows the efficient implementation of many audio processing functions (e.g., mixing, filtering, and equalization) through the digital computer. The conversion from the analog to the digital domain begins by sampling the audio input in regular, discrete intervals of time and quantizing the sampled values into a discrete number of evenly spaced levels. The digital audio data consists of a sequence of binary values representing the number of quantizer levels for each audio sample. The method of representing each sample with an independent code word is called pulse code modulation (PCM). Figure 1 shows the digital audio process.

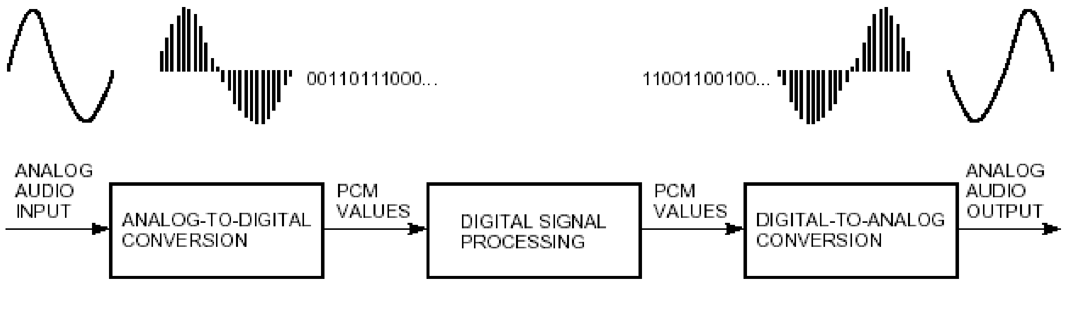


Figure 4. 7 DIGITAL AUDIO PROCESS

According to the Nyquist theory, a time-sampled signal can faithfully represent signals up to half the sampling rate [16]. Typical sampling rates range from 8 kilohertz (kHz) to 48 kHz. The 8-kHz rate covers a frequency range up to 4 kHz and so covers most of the frequencies produced by the human voice. The 48-kHz rate covers a frequency range up to 24 kHz and more than adequately covers the entire audible frequency range, which for humans typically extends to only 20 kHz. In practice, the frequency range is somewhat less than half the sampling rate because of the practical system limitations.The number of quantizer levels is typically a power of 2 to make full use of a fixed number of bits per audio sample to represent the quantized values. With uniform quantizer step spacing, each additional bit has the potential of increasing the signal-to-noise ratio, or equivalently the dynamic range, of the quantized amplitude by roughly 6 decibels (dB).The typical number of bits per sample used for digital audio ranges from 8 to 16. The dynamic range capability of these representations thus ranges from 48 to 96 dB, respectively. To put these ranges into perspective, if 0 dB represents the weakest audible sound pressure level, then 25 dB is the minimum noise level in a typical recording studio, 35 dB is the noise level inside a quiet home, and 120 dB is the loudest level before discomfort begins [17].

In terms of audio perception, 1 dB is the minimum audible change in sound pressure level under the best conditions, and doubling the sound pressure level amounts to one perceptual step in loudness. Compared to most digital data types (digital video excluded), the data rates associated with uncompressed digital audio are substantial. For example, the audio data on a compact disc (2 channels of audio sampled at 44.1 kHz with 16 bits per sample) requires a data rate of about 1.4 megabits per second.

There is a clear need for some form of compression to enable the more efficient storage and transmission of this data. The many forms of audio compression techniques differ in the trade-offs between encoder and decoder complexity, the compressed audio quality, and the amount of data compression. The techniques presented in the following sections of this paper cover the full range from the _-law, a low-complexity, low-compression, and medium audio quality algorithm, to MPEG/audio, a high-complexity, high compression, and high audio quality algorithm.

These techniques apply to general audio signals and are not specifically tuned for speech signals. This paper does not cover audio compression algorithms designed specifically for speech signals. These algorithms are generally based on a modeling of the vocal tract and do not work well for nonspeech audio signals.The federal standards 1015 LPC (linear predictive coding) and 1016 CELP (coded excited linear prediction) fall into this category of audio compression.

4.4.1 U-LAW COMPRESSION The U-law transformation is a basic audio compression technique specified by the Comité Consultatif Internationale de Télégraphique et Téléphonique CCITT) Recommendation G.711 [18]. The transformation is essentially logarithmic in nature and allows the 8 bits per sample output codes to cover dynamic range equivalent to 14 bits of linearly quantized values. This transformation offers a compression ratio of (number of bits per source sample) 8 to 1. Unlike linear quantization, the logarithmic step spacings represent low-amplitude audio samples with greater accuracy than higher-amplitude values. Thus the signal-to-noise ratio of the transformed output is more uniform over the range of amplitudes of the input signal.

PULSE SHAPING

5.1 INTERSYMBOL INTERFERENCE

In high speed communication system, where the data symbols closely follow each other, time dispersion results in overlap of successive symbols, in effect known as intersymbol interference [23] as illustrated.

The following figures depict the transmission of a signal from one point to other. There is considerable interference in the symbols in fig 5.2 when we compare the this waveform with the one shown for the transmitted signal shown in Figure 5.1.

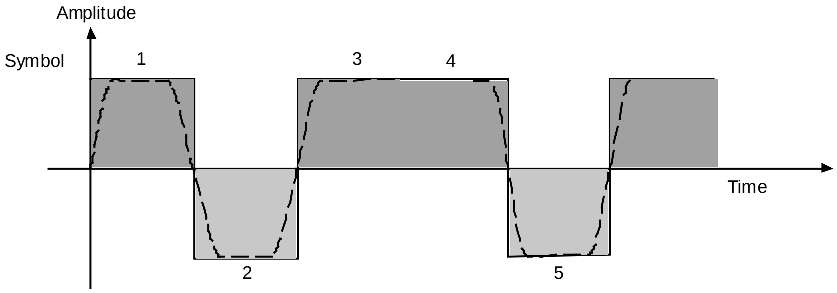


Figure 5.1 Transmitted Signal

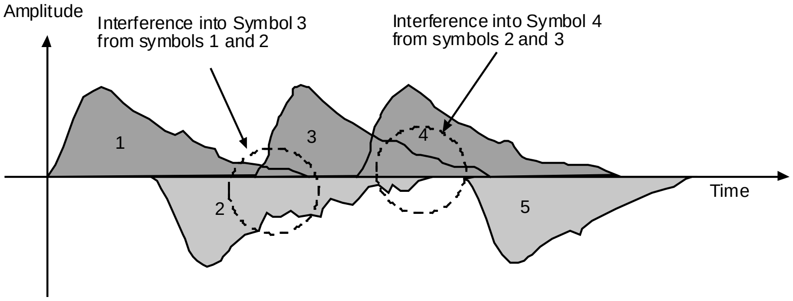

Amplitude
Interference into Symbol 3 from symbols 1 and 2
Interference into Symbol 4 from symbols 2 and 3
1
2
3
4
5
Time

Figure 5.2 - Each symbol is spread by the medium

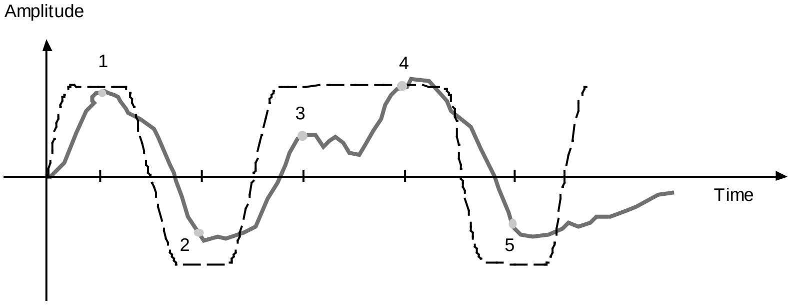


Fig 5.3 - Received signal vs. the transmitted signal

Compared to the dashed line that was the transmitted signal, the received signal looks quite bad. The dots show the value of the amplitude at the timing instant. For symbol 3, this value is approximately half of the transmitted value, which makes this particular symbol is more susceptible to noise and incorrect interpretation.

5.2 REASONS OF OCCURRENCE OF ISI :

ISI can be caused by many different phenomenon's. It can be caused by filtering effects from hardware or frequency selective fading, from non-linearities and from charging effects. Very few systems are immune from it and it is nearly always present in wireless communications.

5.3 COUNTER MEASURES AGAINST ISI:

1) One of the simplest things we can do to reduce ISI is to just slow down the signal.
2) The other method is pulse shaping to keep the symbols from interfering in such a way that they do not affect the amplitude at the sampling instant

5.3.1 Ideal pulse shaping

There is a problem with using a rectangular pulse shape. We say this because recalling from Fourier theory, a rectangular pulse in the time domain is equivalent to a sinc pulse in the frequency domain. [20]

The frequency response of a square pulse (its Fourier Transform) is given by the equation

$$h(t) = \frac{\sin(2\pi t / T_s)}{2\pi t} \qquad (5.1)$$

where T_s = symbol time

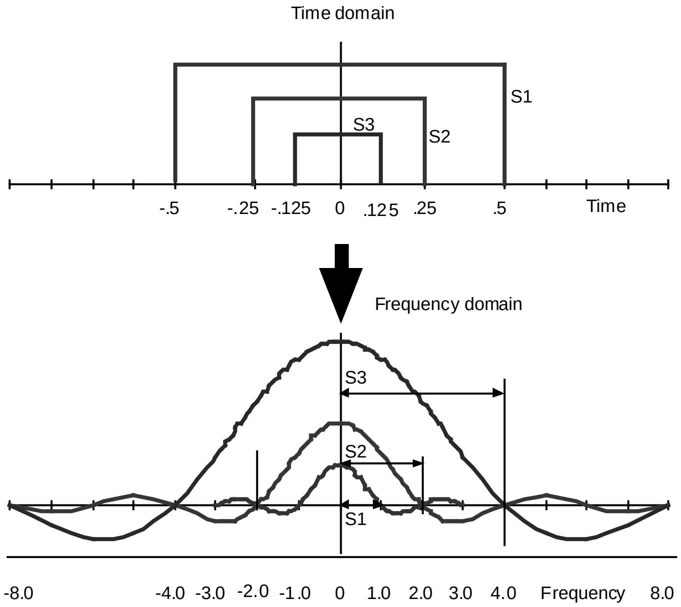


Figure 5.4 Effect of square pulse symbol times and their frequency response

A narrow pulse (which is approaching a delta function) has a wide frequency response because it has a lot of frequency content. A wide one (similar to a flat line if you squint or have a good imagination) has lesser frequency content and hence its bandwidth is smaller. For each pulse, the bandwidth which we measure is only on the positive half and is equal and its symbol rate in Hz.

Hence the square pulse has the following disadvantages:

1. The square pulse is difficult to create in time domain because of rise time and a decay time.
2. Its frequency response goes on forever and decays slowly.
   The second lobe is only 13 dB lower than the first one.
3. It is very sensitive to ISI.

The ideal pulse shape should have two properties.

Property 1:

It should have a limited bandwidth to allow transmission on practical band-limited systems.

Property 2:

The pulse shape should also have zero intersymbol interference if sampled at the correct time interval. That is, when a pulse train is sampled every *T* seconds, the value of the sample at time *T* should only be due to the current pulse.And there should be no interference from the other transmitted pulses. In other words, ideally,

$$h(t) = 1 \text{ for } t = 0 \tag{5.3}$$

and

$$h(t) = 0 \text{ for } t = \pm kT \tag{5.4}$$

where *k* is a non-zero integer.

An ideal pulse shape that meets these requirements is a time-domain sinc pulse. Property one is satisfied as shown in the following figure which compares a sinc pulse with a square one.

As opposed to the square pulse, we see that using the sinc pulse cuts the bandwidth requirement to one-half! In the figure below, several pulses are shown of symbol rates 1, 2 and 4. In each case, we see that the required lowpass bandwidth is one-half of the symbol rate.

The bandwidth achieved by the sinc pulse is called the Nyquist bandwidth and requires only 1/2 Hz per symbol or bandwith W is given by

$$W = 1/2T_s \qquad (5.5)$$

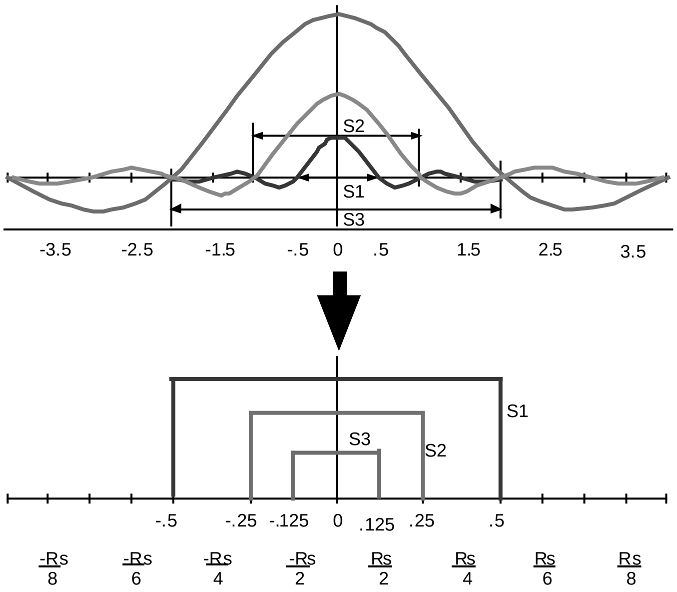


Figure 5.5:Pulse width vs. frequency response

An example of a sinc pulse for which *T* = 10 is shown in Figure 3. Note that the pulse takes on a value of 1 at its peak and its zerocrossings occur at intervals of integer multiples of ±10 samples away from the peak. The lower half of the figure shows a pulse train of four pulses. This example illustrates how the peak of any given pulse lines up with the zerocrossings of the remaining pulses.Therefore, there is no ISI.

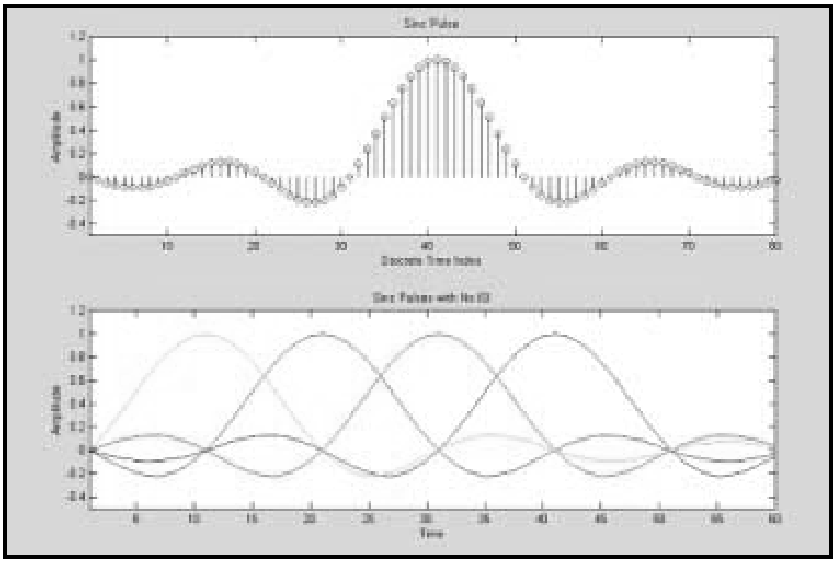


Figure 5.6:Sinc Pulses

5.3.2 Practical pulse shaping

Although the sinc pulse represents the ideal pulse shape, it cannot be implemented in practice because the pulse extends in time for infinite duration. The infinite signal duration is due to the discontinuities in the sinc pulse's rectangular-shaped spectrum. Signals with discontinuities in their spectrum are physically unrealizable. However, practical pulse shapes can be formed by smoothing the roll-off of the spectrum and allowing it to occupy excess bandwidth beyond that which is needed for the spectrum of the ideal sinc pulse.One pulse shape that has properties similar to the sinc pulse, but without the frequency-domain discontinuities, is the raised cosine pulse.

The raised cosine pulses have an adjustable bandwidth which can be varied from W to 2W. We want to get as close to W, which is called the Nyquist bandwidth, as possible with a reasonable amount of power. The factor α related the achieved bandwidth to the ideal bandwidth W as

$$\alpha = 1 - \frac{W}{W_0} \qquad (5.5)$$

where W is Nyquist bandwidth, and W_0 is the utilized bandwidth.

α □□ is the roll off factor. The value of the rolloff factor determines how rapidly the frequency-domain spectrum of the pulse rolls off. α indicates how much bandwidth is being used over the ideal bandwidth. The smaller this factor, the more efficient the scheme. The percentage over the minimum required W is called the excess bandwidth. It is 100% for roll-off of 1.0 and 50% for roll-off of 50%. The alternate way to express the utilized bandwidth is

$$W_0 = (1 + \alpha) R_s$$

(5.6)

5.4 RAISED COSINE PULSE IN TIME DOMAIN:

Here is how the class of raised cosine pulse is defined in time domain.

$$h(t) = \frac{\sin(2\pi t / T_s)}{2\pi t} \frac{\cos(2\pi \alpha t)}{1 - (2\alpha t / \pi)^2} \qquad (5.7)$$

The first part of equation 5.7 is the sinc pulse. The second part is a cosine correction applied to the sinc pulse to make it behave better. The sinc pulse insures that the function transitions at integer multiples of symbol rate which makes it easy to extract timing information of the signal. The cosine part works to reduce the excursion in between the sampling instants. The bandwidth is now adjustable. It can be any where from 1/2 R_s to

R_s. It is greater than the Nyquist bandwidth by a factor (1+ α). For α = 0, the above equation reduces to the sinc pulse, and for α = 1, the equation becomes that of a pure square pulse.

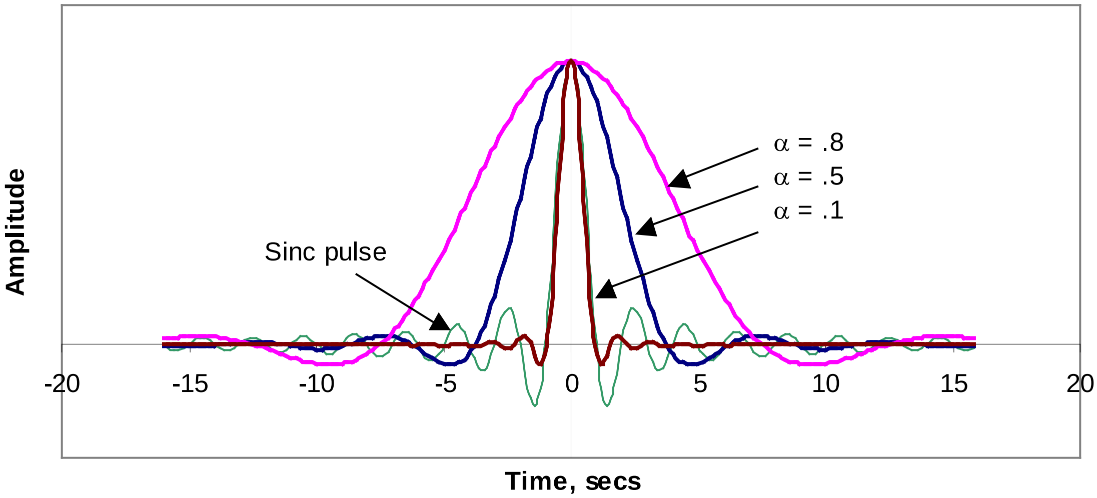


Fig 5.7: Raised cosine impulse response

In frequency domain, the relationship is given by

$$H(f) = \begin{cases} 1 & for\ |f| \le \dfrac{(1-\alpha)}{2T_s} \\ \left\{\cos^2 \dfrac{\pi T_s}{2\alpha}\left(|f| - \dfrac{(1-\alpha)}{2T_s}\right)\right\} & for\ \dfrac{(1-\alpha)}{2T_s} \le |f| \le \dfrac{(1+\alpha)}{2T_s} \\ 0 & for\ |f| > \dfrac{(1+\alpha)}{2T_s} \end{cases}$$

As the rolloff factor is increased, the spectrum begins to decay more gradually and this increased rolloff causes the pulse to occupy more bandwidth. When the rolloff reaches its maximum value of one, the spectrum requires twice as much bandwidth as the pulse with a rolloff of zero. Practical digital communications systems often use a rolloff factor of between 0.10 and 0.35. A pulse with a rolloff factor of 0.35 occupies 35% more bandwidth than the ideal sinc pulse. Figure 5.8 shows the effect of the rolloff factor on the pulse's spectrum. Figure 5.9 contains time-domain raised cosine pulses for the same rolloff factors used in Figure 5.8. The pulses in Figure 5 exhibit zerocrossings at integer multiples of the symbol period. Thus, even with nonzero roll-off factors, the raised cosine pulse maintains this desirable (from the standpoint of no ISI) property of the sinc pulse. The choice of the roll-off factor is a trade-off between required bandwidth and the duration of the time-domain pulse. Tthe tails of the time-domain pulse are reduced for higher values of the roll-off factor. The smaller tails are desirable from a timing recovery standpoint because, in the presence of a timing offset, they will contribute less to ISI compared to the larger tails of the sinc pulse.

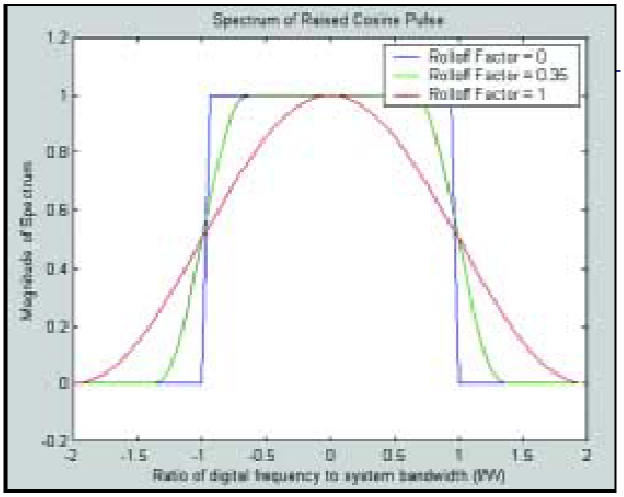


Figure 5.8: Spectrum of raised cosine pulse for different values of the roll-off factor.

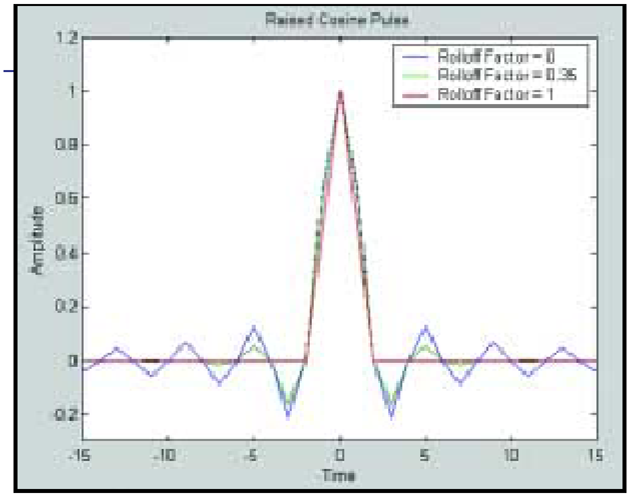


Figure 5.9: Raised cosine pulse for different values of the rolloff factor.

The most popular pulse shape used in practical communications systems is the root-raised cosine pulse. The rootraised cosine pulse is formed by taking the square root of a raised cosine pulse. This pulse shape is used to split the spectral characteristics of the raised cosine pulse equally between the transmitter and receiver.

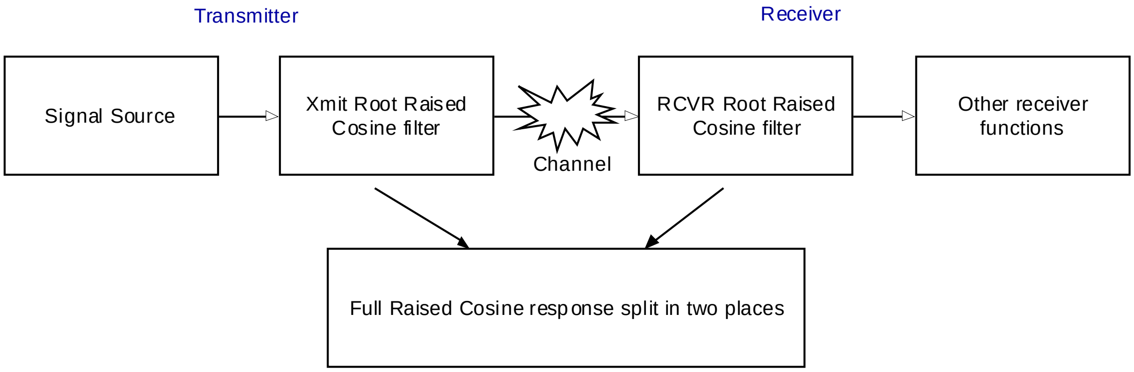

Transmitter
Receiver
Signal Source
Xmit Root Raised Cosine filter
Channel
RCVR Root Raised Cosine filter
Other receiver functions
Full Raised Cosine response split in two places

Figure 5.10: Split filtering of raised cosine response, a root-raised cosine filter at the transmitter and one at the receiver, giving a total response of a raised cosine.

The root raised cosine shaping of pulses is also called baseband filtering. The frequency response of the root raised cosine is given by

$$H\ (f)=\begin{cases}1 & for\,|f|\le\frac{(1-\alpha)}{2T_s}\\ \left\{\cos\frac{\pi T_s}{2\alpha}\left(|f|-\frac{(1-\alpha)}{2T_s}\right)\right\} & for\ \frac{(1-\alpha)}{2T_s}\le|f|\le\frac{(1+\alpha)}{2T_s}\\ 0 & for\,|f|>\frac{(1+\alpha)}{2T_s}\end{cases}$$

Comparing this response for the raised cosine we see it is just a square root relation ship.

By matched-filtering the root-raised cosine pulse and then sampling it at the symbol period, the root-raised cosine pulse is essentially squared. Thus, the output of the matched filter has a raised cosine pulse response.

Comparing the impulse response of the root raised filter to that of the raised cosine we do not see much of a difference except that there is a little bit more excursion in the root-raised cosine response. The time domain function is of course NOT the square root. The root part applies to frequency domain.

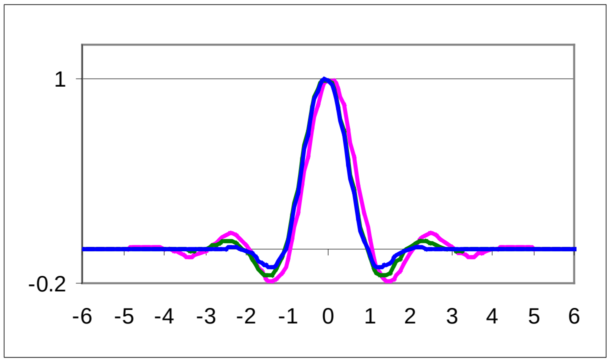


Figure 5.11 – Impulse response of Raised cosine

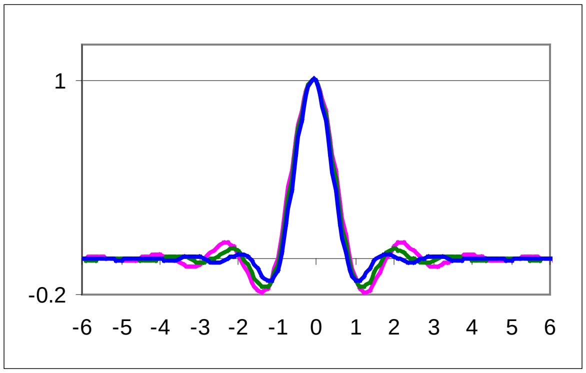


Figure 5.12 – Impulse response of Root Raised cosine

Figure 5.13 shows the spectrum of the root raised cosine and the raised cosine. The bandwidth of the raised cosine filter is specified at the 6 dB point due to the fact that the Nyquist frequency response is reached at 3 dB which is then doubled to give 6 dB for power. The root-raised cosine however is specified at 3 dB point because it is the square root.

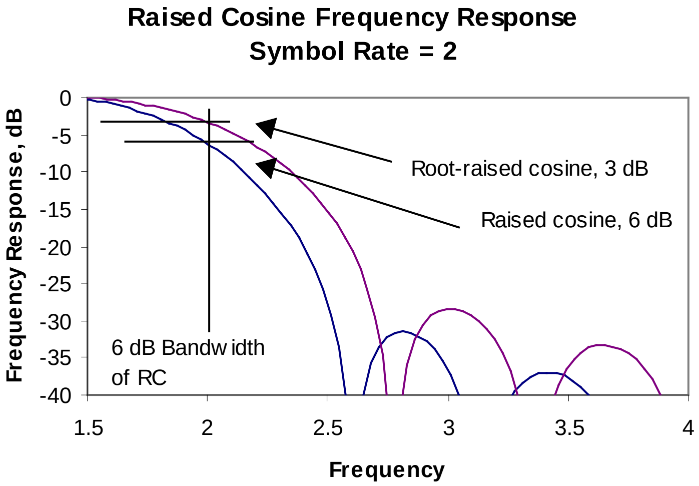


Figure 5.13: Frequency response of Raised cosine and a Root-raised cosine

An example of the matched filter output for a pulse train of root-raised cosine pulses with a rolloff factor of 0.35 is shown in Figure 5.14. Note that the matched filter output exhibits zero ISI because of the locations of the zero crossings for the case of perfect timing.

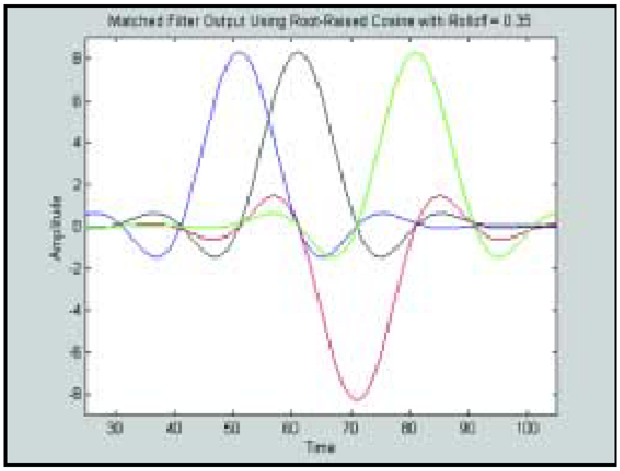


Figure 5.14: Zero ISI at output of matched filter using a root-raised cosine pulse.

Looking at the eye diagram (described later in this chapter) at the transmitter and the receiver gives us some further insight into how the root-raised cosine filtering splits the total filtering.

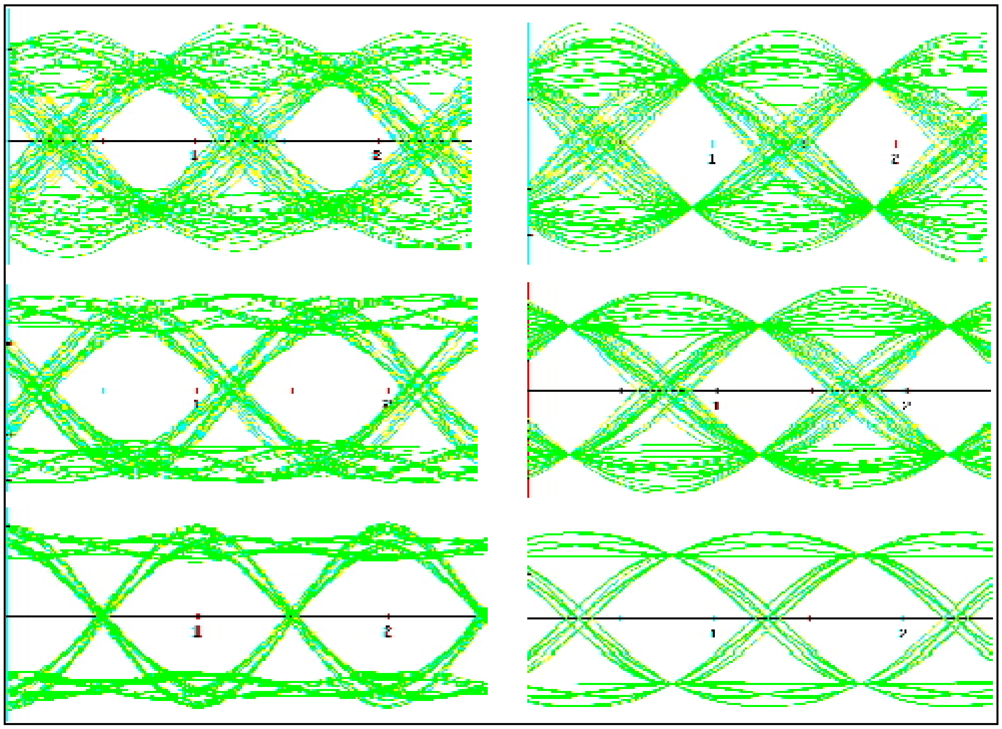


Figure 5.15: Eye diagram of shaped signals, after transmit and receive root raised cosine filters

Root raised cosine pulses out of Transmitter	Root raised cosines pulses at the receiver
α = .2, .4, .6	α = .2, .4, .6

The left side is the signal out of the first root raised cosine filter. The signal does not look that great. But when it is filtered by the second root raised cosine filter (shown on the right side) the output looks really nice, just as we hoped. As the roll-off factor gets larger, the eye opens up. This says us that if there were no bandwidth limitations, it would be much easier on the receiver if we used a large α. But since bandwidth is almost always a limiting resource, the push is on to make α as small as possible.

5.4 DESIGN OF ROOT-RAISED COSINE FILTERS

The raised cosine and the root raised cosine filters are both designed as FIR filters with a specific number of taps. In time domain the fewer taps mean that the impulse response has been truncated. The response of both the raised and the root raised cosine filters is effected by choice of tap length. In general fewer taps give worse rejection. As taps numbers are increased, the length of the time domain sequence is increased and the rejection increases.

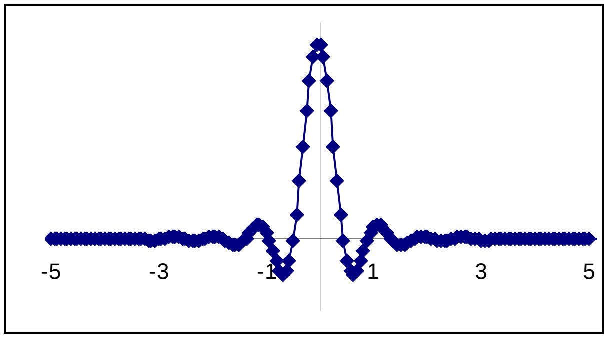


Figure 5.16 – RRC with taps sizes 128

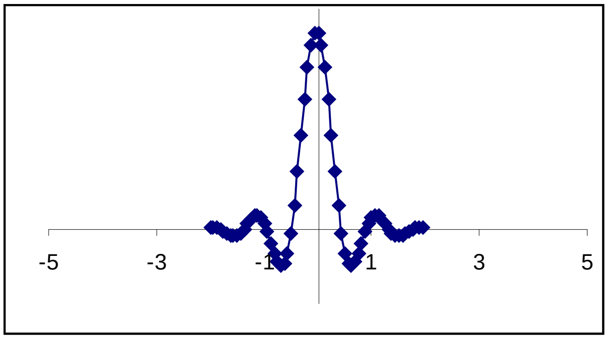


Figure 5.17 – RRC with taps sizes 64.

The two versions shown in figure 5.14 and figure 5.15 are for tap length of 128 and 64. The shape for tap length 64 is exactly the same as the one for 128 taps; the only difference is that it has been truncated. The effect of this truncation is that in frequency response we get some leakage and the spectrum does not drop down as far as one for a longer tap size as shown below in Figure 5.16. This is true for both raised cosine and root raised cosine filters.

As we can see the rejection increases considerably as number of taps is increased. However if 3 dB bandwidths used as the discriminator, there is not a large difference so most RRC filters today use either 48 or 64 tap designs.

Figure 5.16 shows the spectrum of a QPSK signal through the first root-raised cosine filter of α = .3 and various tap lengths.

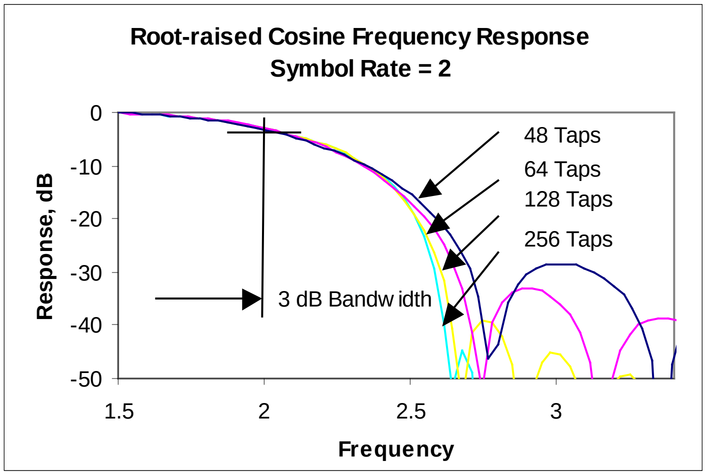


Figure 5.16: Spectrum of a root raised and a raised cosine signal after the first root raised cosine at the transmitter

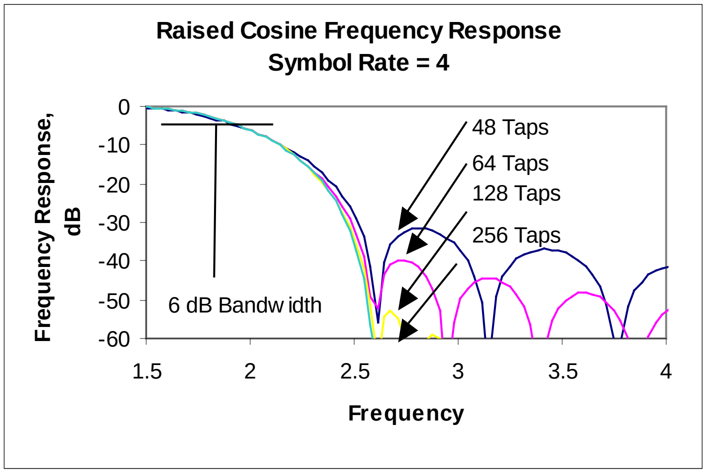


Figure 5.17: Spectrum of a root raised and a raised cosine signal after the second root raised filter at receiver

Figures 5.18 and 5.19 also depict the spectrum of the raised cosine and root raised cosine filters with different values of the taps.

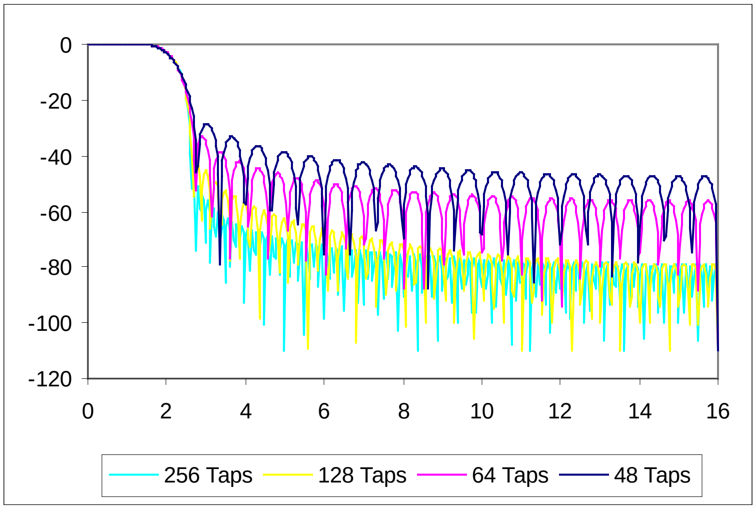


Figure 5.18: Root raised cosine spectrum

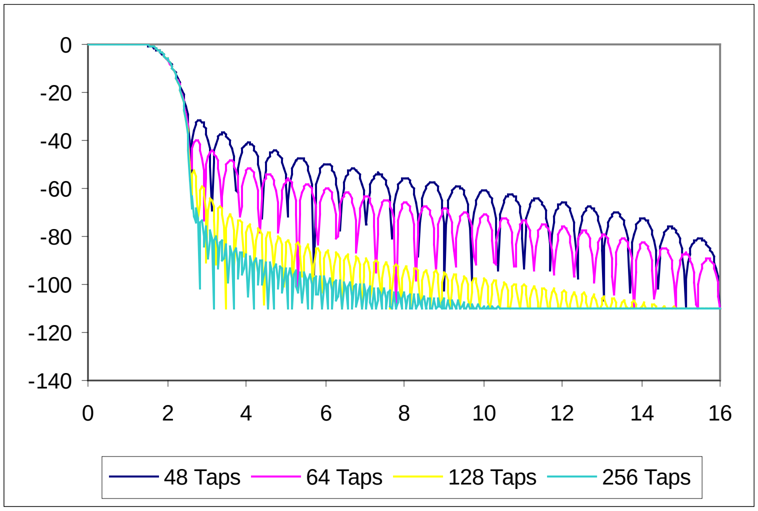


Figure 5.19: Raised cosine spectrum

5.5 MATCHED FILTER:

A matched filter is theoretically the best method to discern whether a particular waveform is present in a noisy environment. If we know the statistics of the noise and the characteristics of the particular signal, then we can build a filter that optimizes the output signal-to-noise ratio (SNR) when the signal of interest is present. [19]

A matched filter is a linear filter designed to provide the maximum signal-to-noise power ratio at its output for a given transmitted symbol waveform. It is called match filter because it impulse response exactly matches with the impulse response of the transmitted signal1 [22]. It will be proved now.

Consider that a known signal *s(t)* plus AWGN *n(t)* is the input to a linear, time-invariant (receiving) filter followed by a sampler. Actually the receive filter is replaced with match filter**.** At time $t = T$, the sampler output *z(T)* consists of a signal component a_i and a noise component n_0 .

$$z\ (T) = a_i + n_0 \qquad (5.7)$$

The variance of the output noise (average noise power) is denoted by $\sigma_0{}^2$ □, so that the ratio of the instantaneous signal power to average noise power. *(S/N)$_T$*. at time t = *T,* out of the sampler is

$$\left(\frac{S}{N}\right)_T = a_i^2 \Big/ \sigma_O^2 \tag{5.8}$$

We wish to find the filter transfer function *H$_o$(f)* that *maximizes e*quation 5.8. We can express the signal *a*$_i$*(t)* at the filter output in terms of the filter transfer function *H(f)* (before optimization) and the Fourier transform of the input signal,

$$a_i\ (t) = \int_{-\infty}^{\infty} H(f)S(f)e^{j2\pi Tf}\,df \tag{5.9}$$

where *S(f)* is the Fourier transform of the input signal, S(t). If the two-sided power spectra} density of the input noise is *No/2* watts/hertz, then, we can express the output noise power as

$$\sigma_o^2 = \frac{N_o}{2}\int_{-\infty}^{\infty} \left|H(f)\right|^2 df \tag{5.10}$$

We then combine equations 5.8 to 5.10 to express *(S/N)$_T$*. as follows

$$\left(\frac{S}{N}\right)_T = \frac{\left|\int_{-\infty}^{\infty} H(f)S(f)e^{j2\pi Tf}\,df\right|^2}{\frac{N_O}{2}\int_{-\infty}^{\infty}\left|H(f)\right|^2 df} \tag{5.11}$$

We next find that value of *H(f)* = *H$_o$(f)* for which the maximum *(S/N)*$_T$ is achieved, by using *Schwarz's inequality.* One form of the inequality can be stated as

$$\left|\int_{-\infty}^{\infty} f_i(x) f_2(x)dx\right|^2 \leq \int_{-\infty}^{\infty}\left|f_1(x)\right|^2 dx \int_{-\infty}^{\infty}\left|f_2(x)\right|^2 dx \tag{5.12}$$

The equality holds if f_1(x) = $kf^*{}_2(x)$ where *k* is an arbitrary constant and * indicates complex conjugate. If we identify *H*(f) with f_1(x) and *S(f)* $e^{j2\Pi fT}$ with f_2(x), then

$$\left|\int_{-\infty}^{\infty} H(f)S(f)e^{j2\pi Tf}\,df\right|^{2} \leq \int_{-\infty}^{\infty}|H(f)|^{2}\,df\int_{-\infty}^{\infty}|S(f)|^{2}\,df \tag{5.13}$$

Substituting into Equation 5.11 yields

$$\left(\frac{S}{N}\right)_{T} \leq \frac{2}{N_{O}}\int_{-\infty}^{\infty}|S(f)|_{2}\,df \tag{5.14}$$

$$\max(\frac{S}{N})_{T} = \frac{2E}{N_{O}} \tag{5.15}$$

Where the energy E of the input signal *S(t)* is

$$E = \int_{-\infty}^{\infty}|S(f)|^{2}\,df \tag{5.16}$$

Thus, the maximum output *(S/N)*$_{\mathrm{T}}$ depends on the input *signal energy* and the power spectral density of the noise, *not on the particular shape* of the waveform that is used. The equality in Equation 5.14 holds only if the optimum filter transfer function $H_{\mathrm{O}}(f)$ is employed, such that

$$H(f) = H_{O}(f) = kS^{*}(f)e^{-j2\pi Tf} \tag{5.17}$$

$$h(t) = \Im^{-1}\left\{kS^{*}(f)e^{-j2\pi Tf}\right\} \tag{5.18}$$

Since S(t) is a real-valued signal, we can write,

$$h(t) = \begin{cases} ks(T-t) & 0 \leq t \leq T \\ 0 & elsewhere \end{cases} \tag{5.19}$$

Thus, the impulse response of a filter that produces the maximum output signal-to-noise ratio is the mirror image of the message signal *s*(t), delayed by the symbol time duration *T*. Note that the delay of *T* seconds makes Equation 5.19 causal; that is, the delay of *T* seconds makes *h*(t) a function of positive time in the interval $0 < t < T$ as shown in figure 4.14. Without the delay of *T* seconds, the response *s*(–*t*) is unrealizable because it describes a response as a function of negative time.

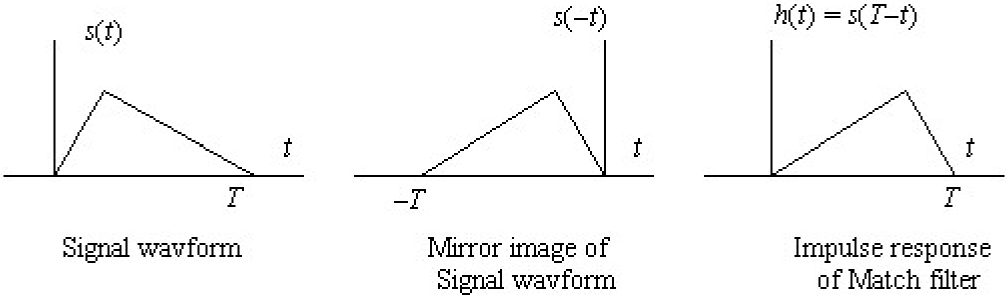

s(t)
t
T
Signal wavform
s(−t)
t
−T
Mirror image of Signal wavform
h(t) = s(T−t)
t
T
Impulse response of Match filter

Figure 5.20: Impulse responses of received signal and match filter

The above mathematical discussion proves that if the impulse responses of the received signal and the match filter are mirror images of each other then at t=T the SNR is maximized. Actually the convolution with itself is a process of integration. By match filtering we are actually integrating the received signal. AWGN is a zero mean random variable. By averaging we are trying to force it to zero. It can be verified that as *T* approaches ∞ the noise averaged to zero.

Such processing has two advantages. One advantage is that typical pulse shapes have a low-pass response. By filtering the received signal with such a filter at the receiver, the frequencies containing the data signal are passed while the remaining frequencies are attenuated. This matched filtering limits the amount of the noise spectrum that is
passed on to subsequent stages in the receiver.

A second advantage is that a matched filter correlates the received signal with the transmit pulse shape over the symbol period *T*. Recall that passing a signal *r*(*t*) through a filter $h_m(t)$ is a convolution operation. The convolution of these two signals can be written as:

$$y(t) = \int_T^o r(t)h_o(T-t)dt \tag{5.20}$$

where *y(T)* represents the output of the matched filter sampled at time *T*. However, the matched filter's response was defined as

$$h_m(t) = h(T - t) \tag{5.21}$$

By substituting this definition into the above equation (equation 5.20), the following integral is obtained:

$$y(t) = \int_T^o r(t)h\ (T-(T-t)dt \tag{5.22}$$

$$= \int_0^T r(t)h(t)dt \tag{5.23}$$

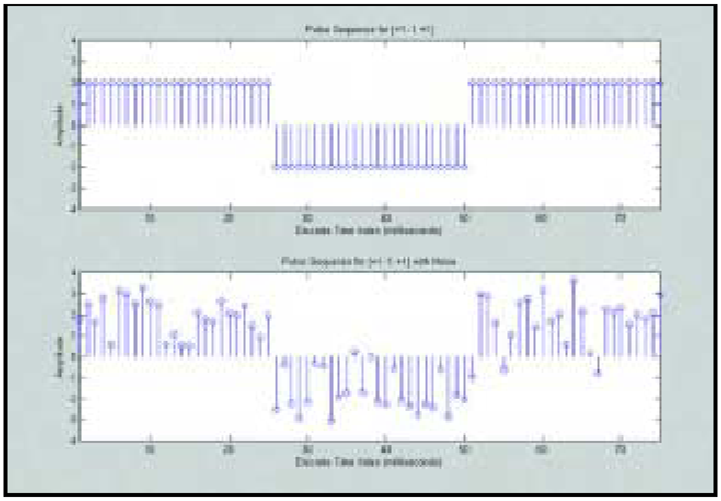


Figure 5.21: Rectangular pulse train shown with and without noise. The SNR for the lower plot is roughly 5 dB.

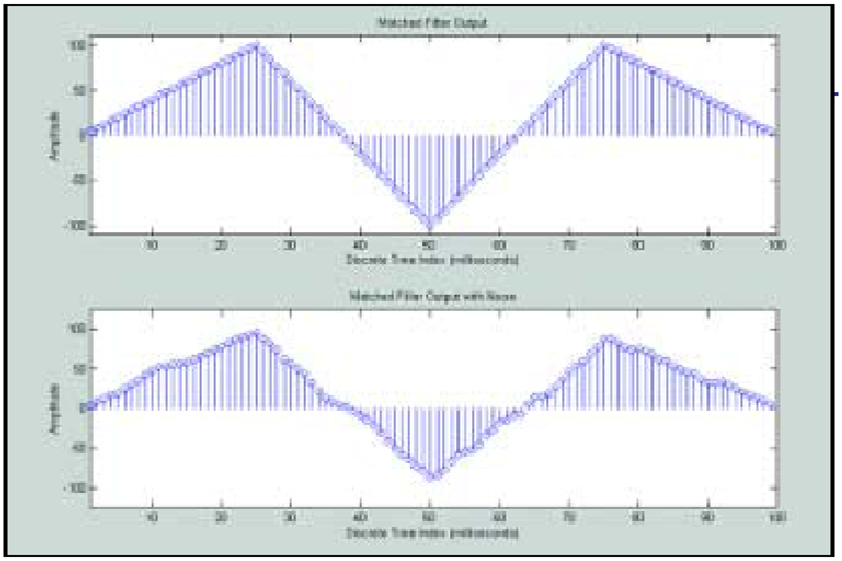


Figure 5.22. Matched filter outputs for signals in Figure 5.21.

The above equation is the cross-correlation (sampled at time *T*) of *r*(*t*) with *h*(*t*) for a lag of 0. Thus, this simple derivation has illustrated how matched filtering effects the correlation of the received signal with the matched filter. Such processing results in a correlation gain by integrating the received signal energy while averaging out the zero-mean AWGN. An example of matched filtering is shown in Figure 5.22. The received signals for the top and bottom halves of the figure are the signals shown in Figure 5.21.

The matched filter used was:

$$h_m(t) = 2 \text{ for } 0 \leq t \leq T$$

Note that sampling the matched filter output at time $T = 25$ ms provides the sample with the highest SNR. The samples from Figure 5.21 had an amplitude of 2, whereas the matched filter output (when sampled properly) has a value of 100. The value of 100 representsthe integral, over the time period

T, of the received signal pulse shape exactly lined up with the matched filter response. The value of this peak can be calculated as follows:

$$y(t) = \int_0^\infty (2.2)dt \qquad (5.24)$$

$$= 25(4)-0(4)=100$$

As can be seen from Figure 2, it is important to sample the matched filter’s output exactly at time *T* to obtain the sample with the highest SNR. Sampling the matched filter’s output at some time $T + \Delta$ will significantly reduce the effective SNR seen by subsequent receiver blocks.

5.6 Eye Diagram:

With suboptimal filtering, it is useful to quantify the degradation of the signal. A useful graphical illustration of the degradation is the *eye diagram,* so called because shape is similar to that of the human eye. An eye diagram is easily generated using an oscilloscope to observe the output of the receive filter, where the symbol timing serves as the trigger. Such displays have historically served as a quick check of the performance of a modem in the field. The eye diagram is also a useful design tool during the analytical and simulation design phase of the system.

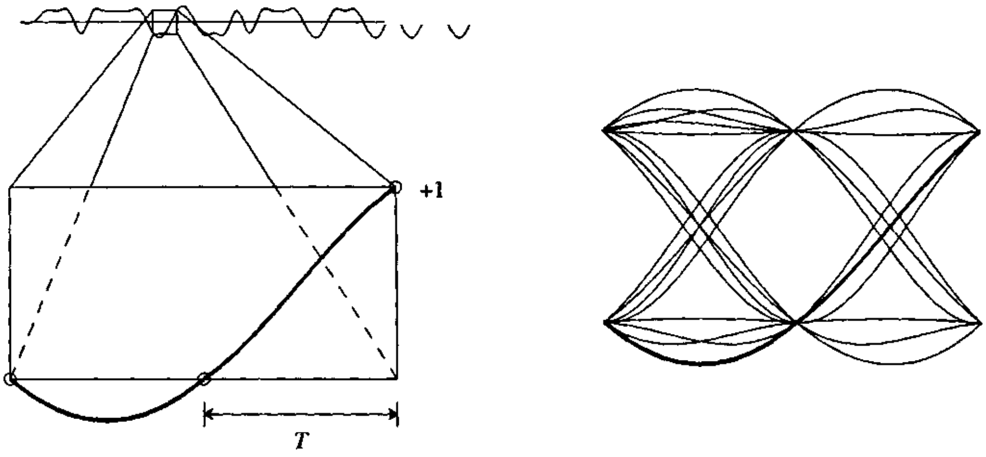


figure 5.21. a binary ram signal made with 50% excess-bandwidth raised-cosine pulses. a segment of length *2t* is shown in detail in (a). the small circles indicate the sample points where symbols are unperturbed by neighboring symbols. in (b), an eye diagram is made by overlaying sections of length *2t*. the component from

part (a) is shown darkened. this display is typical of an oscilloscope display of a signal, where the oscilloscope is triggered at the symbol rate.

An eye diagram consists of many overlaid traces of small sections of a signal, as own in Figure 6-5. If the data symbols are random and independent, it summarizes visually all possible intersymbol interference waveforms. It summarizes several features of the signal, as shown in Figure 5.22. In the presence of intersymbol interference, when the pulse shape does not satisfy the Nyquist criterion, the eye diagram will tend to close vertically. For error-free transmission in the absence of noise, the eye must maintain some vertical opening, since otherwise there are intersymbol interference waveforms that will cause errors

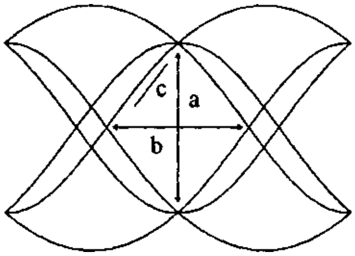


Figure 5.22. A summary of the salient features of an eye diagram. The vertical eye opening (a) indicates the immunity to noise. The horizontal eye opening (b) indicates the immunity to errors in the timing phase. The slope of the inside eye lid (c) indicates the sensitivity to jitter in the timing phase. The ZF criterion is satisfied if all traces pass through the two symbol values in the center of the eye.

When there is incomplete vertical closure, the intersymbol interference will reduce the size of the additive noise required to cause errors. Hence, the wider the vertical opening, the greater the noise immunity. The ideal sampling instant is at the point of maximum (vertical) eye opening, but this can never be achieved precisely by practical timing recovery circuit. Thus the horizontal eye opening is also practically important, since the smaller this opening the greater the sensitivity to errors in timing base (the instant at which the signal is sampled).

The shape of the eye is determined by the pulse shape. In particular, the vertical ye opening is determined by the size of the pulse at multiples of *T*, and the horizon-d eye opening is determined by the size of the tails of the pulse $p(t)$. In Figure 5.23 are shown two eye diagrams for 25% and 100% excess-bandwidth raised-cosine useless. It is important to note the beneficial effect of increasing the excess bandwidth in terms of horizontal eye opening. However, more bandwidth might allow more noise to reach the decision slicer, if we do not carefully design the receive filter. 'bus, there is a basic system tradeoff between excess bandwidth, noise immunity, and the complexity of the timing recovery circuitry. In particular, without special coding it is futile to try to achieve zero excess bandwidth because the horizontal eye opening is zero

An eye diagram for a four-level PAM signal is shown in Figure 5.24.

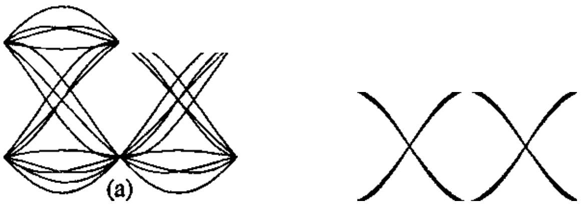


Figure 5.23. Eye diagrams for (a) 25% and (b) 100% excess bandwidth raised-cosine pulses. Note that the pulse with larger tails and less bandwidth (25%) has a smaller eye opening.

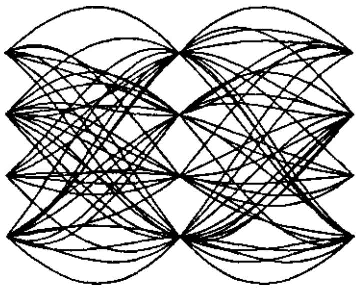


Figure 5.24. An eye diagram for a baseband PAM signal made with 25% excess-bandwidth raised-cosine pulses and an alphabet with four equally spaced symbols.

5.7 FIR Filter Design :

FIR filters have the advantage of being always stable and capable of providing linear phase characterstics, i.e. phase lag is proportional to the frequency. [21] Consider

$$x(t) = \sin(t)$$

With linear phase lag,

$$y(t) = C \sin(t - k) = C$$

Thus, time delay is constant (k) irrespective of frequency. So, various frequency components of x(t) are delayed by the same amount and the wave shape is preserved.

An FIR filter of length M is described by a difference equation of the type:

$$y[n] = b0x[n] + b1x[n] + ... + bM-1x[n]$$

THE MODULATION PROCESS

6.1 INTRODUCTION

The messages emanate from a message source. A vector encoder converts each message into a symbol, which is a real vector x that represents the message. Each possible message corresponds to a distinct value of the symbol vector x. The words, symbol and .message. are often used interchangeably, with the tacit understanding that the symbol actually represents the message via the action of the encoder. A message from the set of M possible messages mi i = 0, ...,M -1 is sent every T seconds, where T is the symbol period for the discrete data transmission system. Thus, one message is sent every T seconds at the symbol rate of 1/T messages per second. The number of messages that can be sent is often measured in bits so that b = $\log_2$(M) bits are sent every symbol period. Thus, the data rate is R = b/T bits per second. The message is often considered to be a real integer equal to the index i, in which case the message is abbreviated m with possible values 0, ...M - 1.

6.2 LINE CODING:

Coding is used to control the statistics of data symbol, thereby introducing a measure of control over the spectrum of the transmitted signals. [24]

Line coding is not a big issue in passband systems as baseband. Baseband systems, particularly those operating over cable, typically have a large variation in attenuation over the Nyquist bandwidth also a large variation in crosstalk coupling loss.

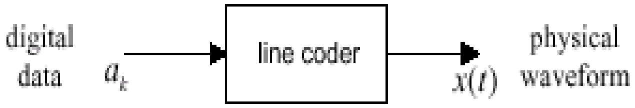


Figure 6.1: Line Coding

Line code is formed by the following method:

- the input to line encoder is a sequence of values a_k, that is a function of data bit and or

  an ADC output bit.

- The output is a waveform

$$x(t)=\sum a_k\ p(t-kTb) \qquad (6.1)$$

- Where p(t) is the **pulse shape** and Tb is the **bit period.**

- Tb=Ts/n for n-bit quantizer. (Pulse Duration).
- Rb(BW)=1/Tb=nfs for n-bit quantizer. (BW requirement).

- The operational details of this function are set by particular type of the line code that is being used.

Various line codes used in the transmission process are shown in figure 6.1

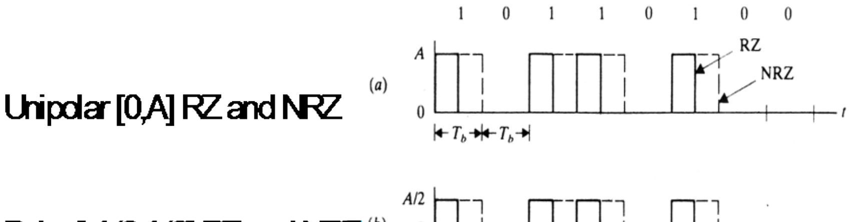


Figure 6.2: Types of Line Codes

6.2.1 Goals of Line codes:

line code is designed to meet several goals, a few of which are explained below [25]

1) Self Synchronization

- This is the ability to recover the timing from the signal itself.
- Long series of Zeros and Ones could cause a problem which has to be tackled.

2) Low Probability of Error.

- The receiver needs to be able to distinguish the waveform associated with a mark from the waveform associated with a space, even if there is a considerable amount of noise and distortion in the channel.

3) Spectrum that is suitable for the channel.

- In some cases DC component should be avoided e.g. if the channel has a DC blocking capacitance.
- The transmission bandwidth should be minimized.

4) Power Efficiency
   - This should be small, for given BW and Specified detection error probability.
   - Favorable Power spectral density (PSD is explained later in this chapter)
     i. Ideally PSD must be zero at ω=0 (dc), due to use of the ac coupling and transformers at the repeaters.
     ii. Adequate Timing Content.
     iii. Ability to extract timing (or clock) information from the signal.

5) Transparency

6) Reliable transmission of data should be ensured

6.2.2 Problems with line codes:

A big problem with the line codes discussed so far is that they are not band-limited: the absolute bandwidth is infinite. the power outside the 1st null bandwidth is non-negligible and the side lobes can be quite high therefore, can cause ***“adjacent channel interference (aci***)”. If the channel is band-limited, then high frequency components will not be passed. high frequency components correspond to sharp transitions in the pulse, thus the pulse will spread out. If the pulse spreads out into the adjacent symbol period, then “***intersymbol interference”*** occurs.

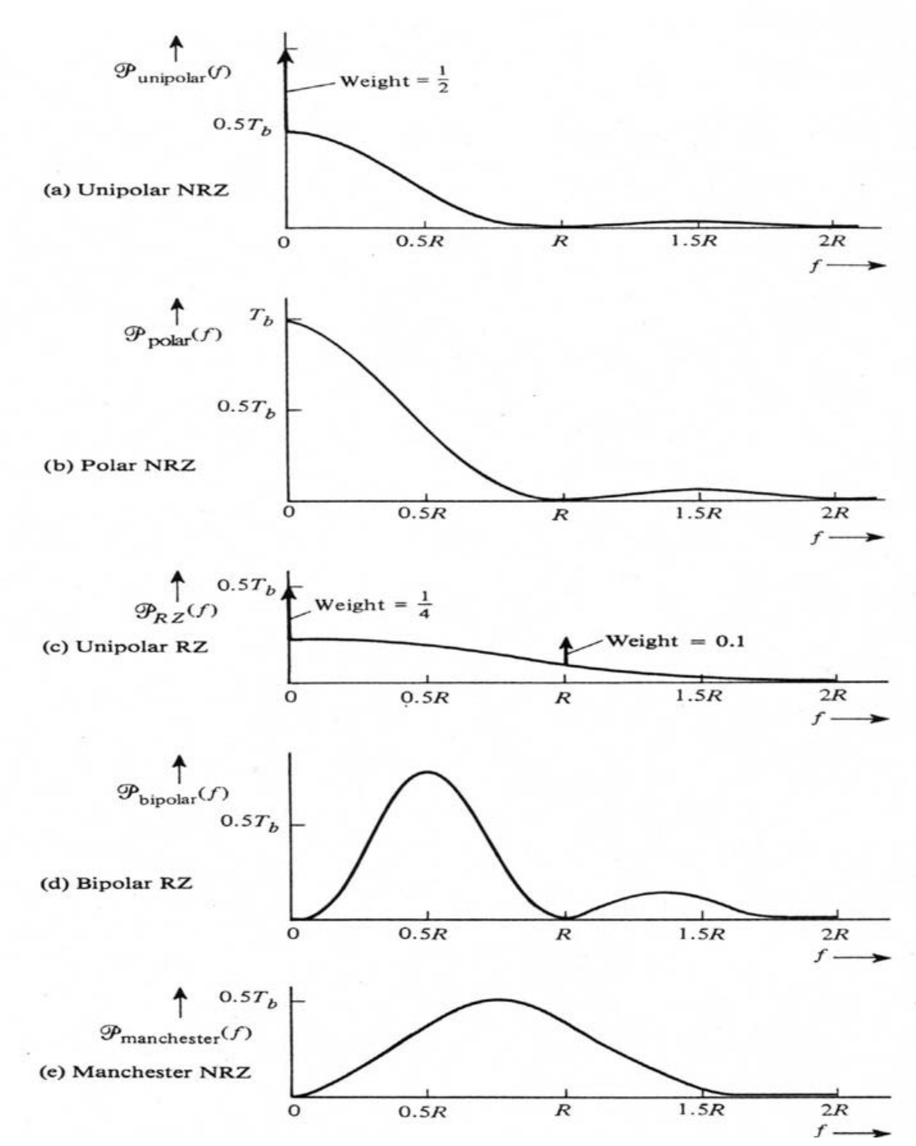


Figure 6.3 PSD’s of Various Line Codes

6.3: DIFFERENTIAL ENCODING

Differential Encoding is especially important for baseband communication [26]. It provides the following advantages:

1) It protects against phase reversal
2) It provides phase reference

Bit streams going through the many communications circuits in the channel can be unintentionally inverted. Most signal processing circuits can not tell if the whole stream is inverted. This is also called phase ambiguity. Differential Encoding is used to protect against this possibility. It is one of the simplest form of error protection coding done on a baseband sequence prior to modulation.

Main purpose of Differential Encoding is to protect against polarity reversals of input bit sequences. Hence Differentially Encoded data sequences have a slightly superior error performance. Differential Encoding is also used to provide a way to decode a BPSK signal, called DEBPSK or DPSK.

6.4: MODULATION:

The messages are usually digital sequences of bits, which are usually not compatible with transmission of physical analog electrical signals through a communication channel. Thus the messages are converted into electrical signals that can be sent through the channel.

The modulator converts the symbol vector x that represents the selected message into a continuous time (analog) waveform that the transmitter outputs into the channel. There is a set of possible M signal waveforms {xi(t)} that is in direct one-to-one correspondence with the set of M messages. The demodulator converts continuous-time channel output signals back into a channel output vector y, from which the detector tries to estimate x and thus also the message sent. The messages then are provided by the receiver to the message sink.

In any data transmission system, the physically transmitted signals are necessarily analog and continuous time. In general, the conversion of a discrete data signal into a continuous time analog signal is called modulation. The inverse process of converting the modulated signal back into its original discrete form is called demodulation. Thus, the combination of encoding and modulation in the transmitter leads to the mapping:

discrete message m_i → x_i (t) continuous waveform .

Conversely, the combination of demodulation and detection in the receiver leads to the mapping:

continuous waveform y(t) → m discrete message .

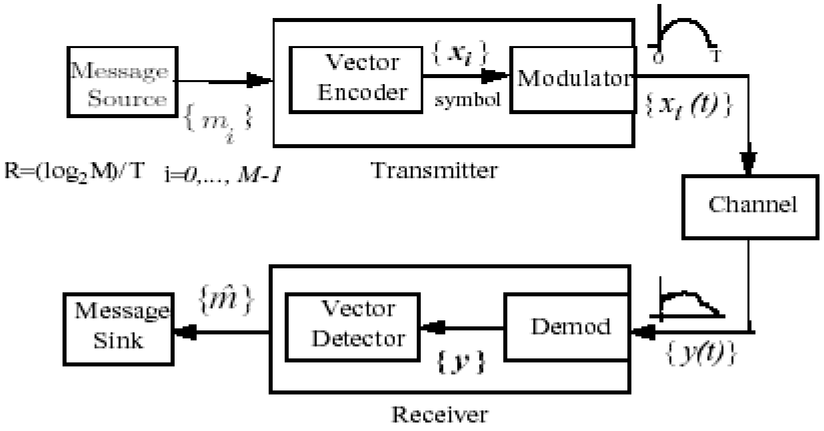


Figure 6.4: The Modulation Process

To understand modulation it would be best to first familiarize with the concepts of baseband and passband signals as eventually modulation is concerned with traversing from baseband to passband.

6.4.1: BASEBAND SIGNALS:

In our project we are mainly concerned with baseband processing so we analysis of baseband signal is very essential for us.

The most salient feature of information signals is that they are generally low frequency. Sometimes this is due to the nature of data itself such as human voice which has frequency components from 300 Hz to app. 20 KHz. Other times, such as data from a digital circuit inside a computer, the low rates are due to hardware limitations.

Due to their low frequency content, the information signals have a spectrum such as that in the figure below. There are a lot of low frequency components and the one-sided spectrum is located near the zero frequency.

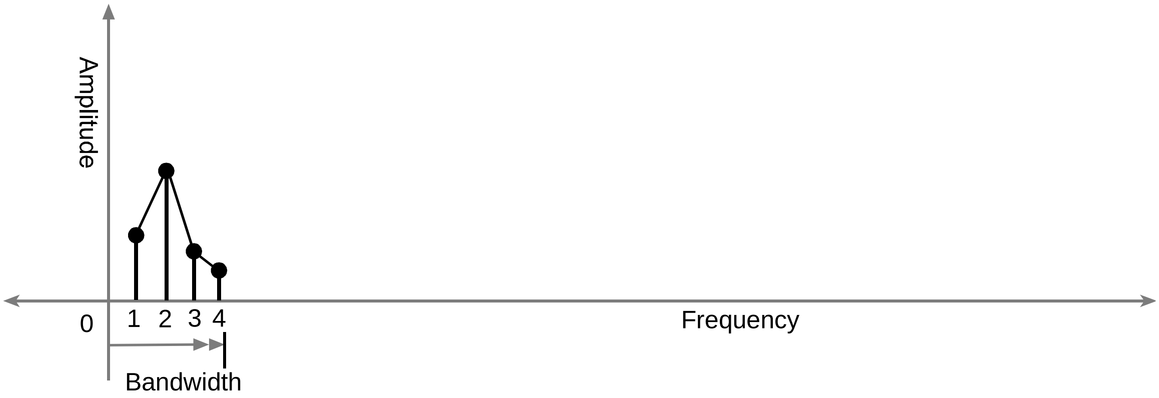


Figure 6.5: The spectrum of an information signal is usually limited to low frequencies

The hypothetical signal above has four sinusoids, all of which are fairly close to zero. The frequency range of this signal extends from zero to a maximum frequency of f_m. We say that this signal has a bandwidth of f_m.

In the time domain this 4 component signal may looks as shown in Figure 6.6.

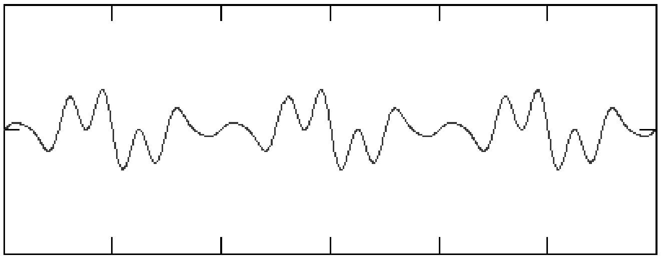


Figure 6.6 - Time domain low frequency information signal

The information signal is called the baseband signal. The bandwidth is always a positive quantity so the bandwidth of this signal is f_m.

6.5: PASSBAND SIGNAL

Passband modulation methods are useful in many applications where transmission occurs over a limited narrow bandwidth, typically centered at or near the carrier frequency of the passband modulation. Digital television transmission on Channel 2 in the US has carrier frequency 52 MHz and non-neglible energy only from 50 to 56 MHz. Digital cellular phones use carrier frequencies in the 900 MHz (and 1800 and 1900 MHz) bands, but have nonzero energy over a narrow band that is typically only 30 kHz (US approximate) to 300 kHz (European GSM) wide.1 Digital Satellite transmission uses QAM and carriers in the 12 and 17 GHz bands with transponder bandwidths of about 26 MHz.

6.5.1: Carrier-Modulated Signal

A carrier-modulated signal is any passband signal that can be written in the following form

$$x(t) = a(t) \cos (\omega_c t + \theta(t)) \tag{6.2}$$

where a(t) is the time-varying amplitude or envelope of the modulated signal and θ (t) is the time-varying phase. ω_c is called the carrier frequency (in radians/sec).

The carrier frequency ω_c is chosen sufficiently large compared with the amplitude and phase variations of a(t) so that the power spectral density (explained later this chapter) does not have significant energy at $\omega = 0$. This can be seen in Figure 6.7, where in the spectrum of X(ω) is concentrated in the passband $\omega_{low} < |\omega| < \omega_{high}$.

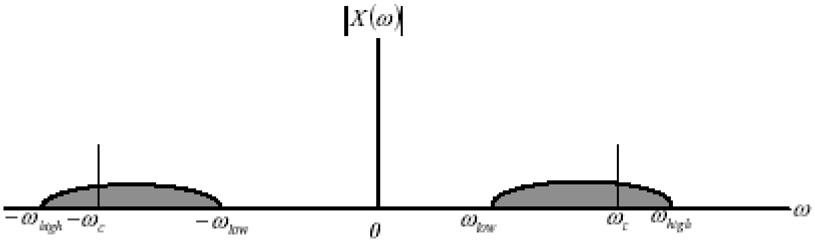


Figure 6.7: Passband signal amplitude.

By definition, a bandpass signal has null at and near DC and In digital communication, x(t) is equivalently written in quadrature form using the trigonometric identity

$$\cos(u + v) = \cos(u) \cos(v) - \sin(u) \sin(v), \quad (6.3)$$

leading to a quadrature decomposition which can be explained as I and Q formats described later in this chapter.

The multiplication of this signal with a sinusoid carrier signal translates the whole thing up to f_c. This signal is now called the passband signal. This signal extends in range from ($-f_c$ - f_m) to (f_c + $f_{m.}$). The new signal has doubled in bandwidth. The passband signal bandwidth is double that of the baseband signal.

The process is described as under:

In Figure 6.8, we see the two sided spectrum of the message signal. After mixing, modulating or heterodyning (all of these terms refer to the same thing), we get a spectrum such as in Figure 6.9. The spectrum is now shifted up to the carrier frequency and we see that it is replicated on both sides of the y-axis.

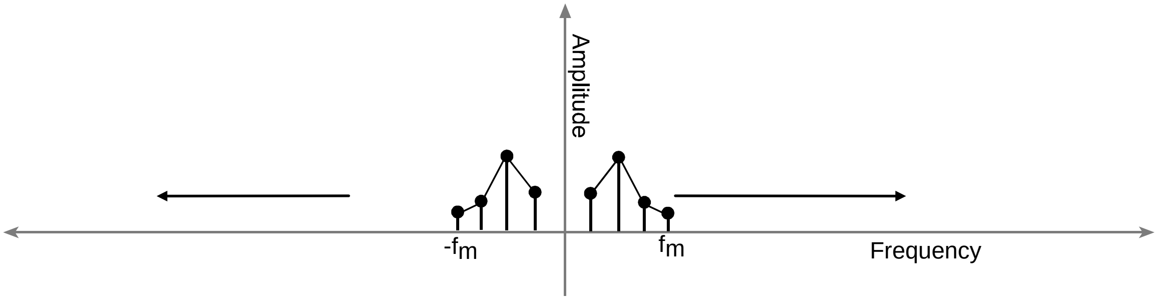


Figure 6.8 - the Baseband Spectrum

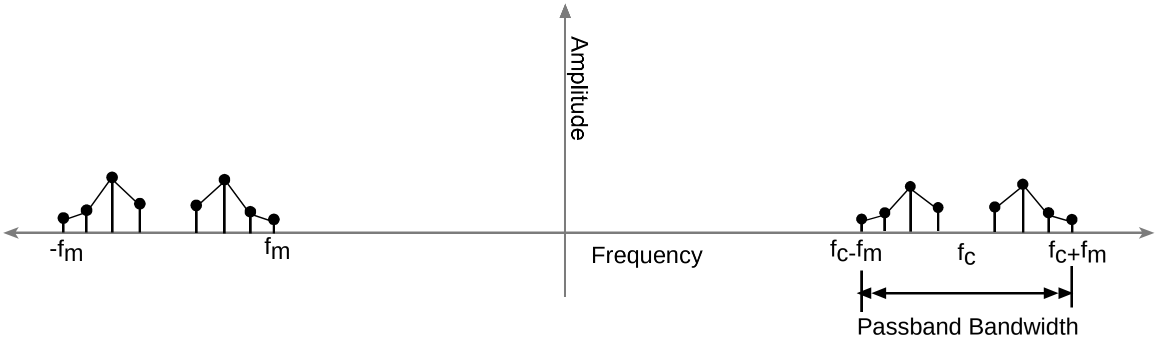


Figure 6.9 - the Passband Spectrum of the same signal

Another way to describe the process is that multiplication by a sinusoid, shifts one copy of the spectrum to f_c and an another to -f_c. Why does this happen? The reason is explained by the Fourier Transform of this signal which is a product of two signals, one of them a sinusoid.

$$f(t) = m(t) \times \cos w_c t \tag{6.4}$$

The Fourier transform of f(t) is just the Fourier Transform of the signal m(t), half of it shifted up and half of it down.

$$F(f) = \frac{1}{2}[M(f - f_c) \quad + \quad M(f + f_c)] \tag{6.5}$$

In time domain, we see that this signal has much higher frequency. But its envelope is still the original low frequency signal of Figure 6.6.

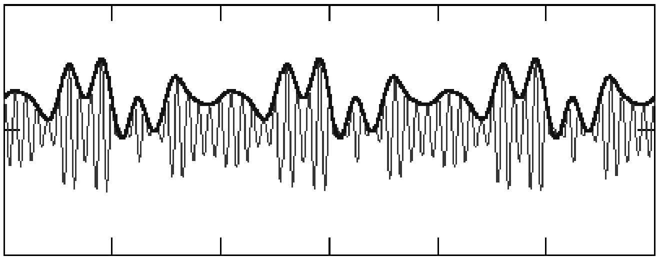


Figure 6.10 - Output signal of a product modulator, the envelope of which is the information signal (see also Figure 2)

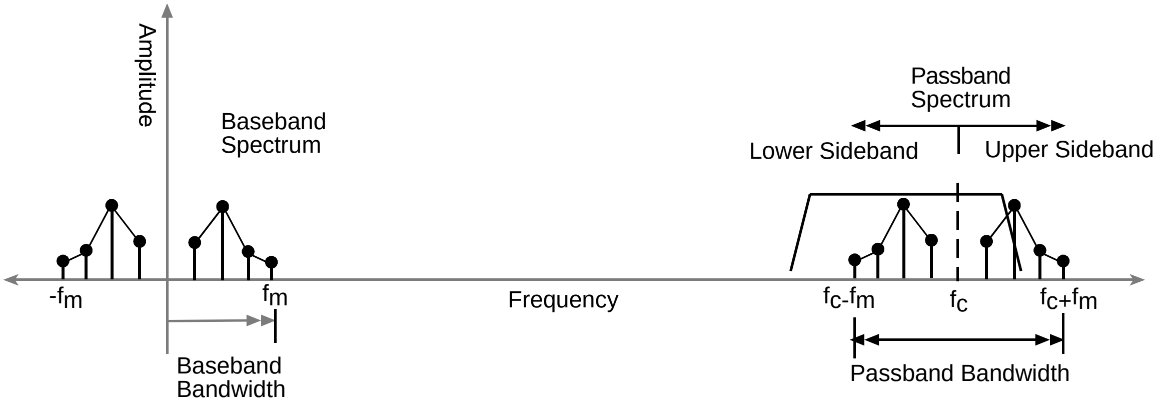


Figure 6.11 - Baseband becomes Passband by translation to higher frequency The positive frequency spectrum becomes the upper side-band and the negative frequency spectrum become the lower side band.

6.6 ADVANTAGE OF MODULATION:

The move to digital modulation provides more information capacity, compatibility with digital data services, higher data security, better quality communications, and quicker system availability.

Developers of communications systems face these constraints:

- available bandwidth
- permissible power
- inherent noise level of the system

The RF spectrum must be shared, yet every day there are more users for that spectrum as demand for communications services increases. Digital modulation schemes have greater capacity to convey large amounts of information than analog modulation schemes.

6.7 SIGNAL CHARACTERISTICS THAT CAN BE MODIFIED

There are only three characteristics of a signal that can be changed over time: amplitude, phase, or frequency. However, phase and frequency are just different ways to view or measure the same signal change.

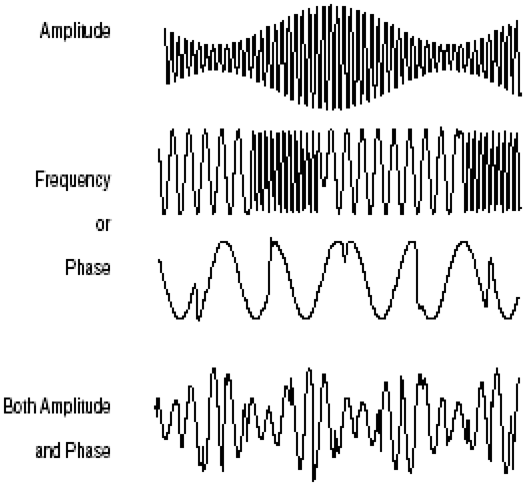


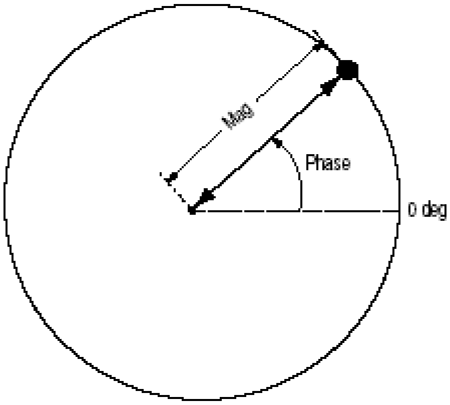
Figure 6.12 Signal Characteristics to Modify

6.8 SIGNAL REPRESENTATIONS

There are various form to represent the signal and their modulation schemes. A few of the more famous methods are the Polar display and the I/Q formats. Both of these are widely used for this purpose and ease the understanding of the communication process.

6.8.1 Polar Display

A simple way to view amplitude and phase is with the polar diagram. The carrier becomes a frequency and phase reference and the signal is interpreted relative to the carrier. The signal can be expressed in polar form as a magnitude and a phase. The phase is relative to a reference signal, the carrier in most communication systems. The magnitude is either an absolute or relative value. Both are used in digital communication systems. Polar diagrams are the basis of many displays used in digital communications, although it is common to describe the signal vector by its rectangular coordinates of *I*(In-Phase) and *Q* (Quadrature)



.

Figure 6.13: Polar Display—Magnitude and Phase Represented Together

Figure 6.14 shows different forms of modulation in polar form. Magnitude is represented as the distance from the center and phase is represented as the angle. Amplitude modulation (AM) changes only the magnitude of the signal. Phase modulation (PM) changes only the phase of the signal. Amplitude and phase modulation can be used together. Frequency modulation (FM) looks similar to phase modulation, though frequency is the controlled parameter, rather than relative phase.

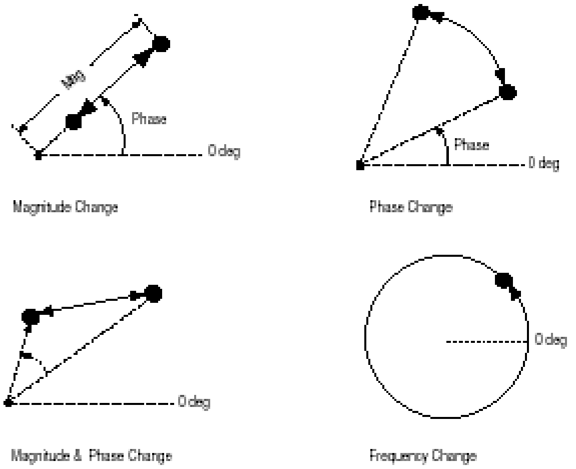


Figure 6.14 Signal Changes or Modifications

6.8.2 I/Q formats

In digital communications, modulation is often expressed in terms of *I* and *Q*. This is a rectangular representation of the polar diagram. On a polar diagram, the *I* axis lies on the zero degree phase reference, and the *Q* axis is rotated by 90 degrees. The signal vector's projection onto the *I* axis is its "I" component and the projection onto the *Q* axis is its "Q" component.

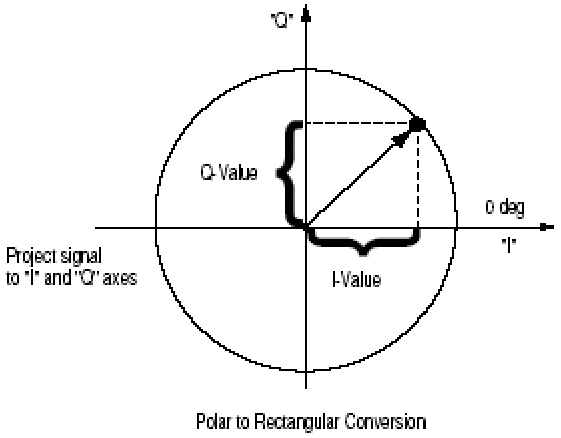


Figure 6.15: "*I-Q*" Format

The quadrature decomposition of a carrier modulated signal can be expressed by the following expression:

$$x(t) = x_I\,(t)\cos(\omega_c t) - x_Q(t)\sin(\omega_c t)$$
$$= A(t)\cos((\omega_c t) + \theta\,(t)) \qquad (6.6)$$

Information is contained in the envelope (amplitude and phase), not the carrier. Distinguish between them with a concise notation :

$$x(t) = \mathrm{Re}\,\{\, z(t)\, e^{\,jwct}\,\} \qquad (6.7)$$

Where

$$z(t) = x_I\,(t) + j\,x_Q\,(t) \;= A(t)\, e^{\,j\theta(t)} \qquad (6.8)$$

z(t) is the complex envelop, or the baseband equivalent of *s*(*t*). It is like a time varying phasor □. The complex rep

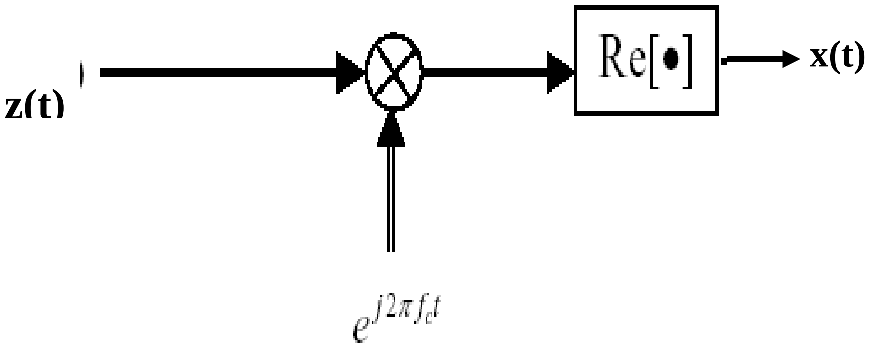


Figure 6.16 complex envelope

6.8.2.1 I and Q in a radio transmitter

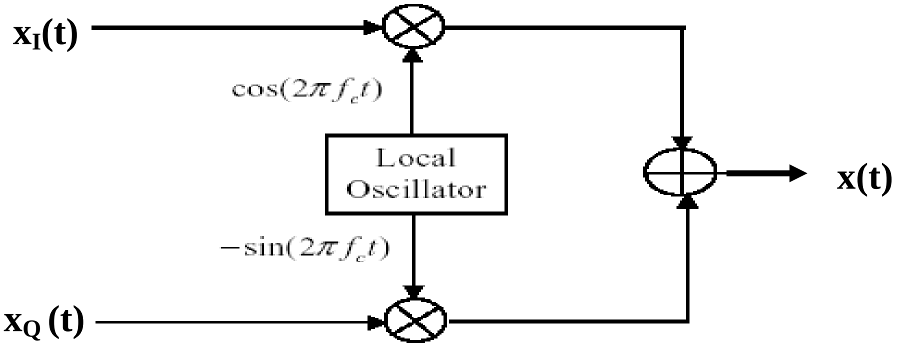


Figure6.17: I and Q channels at the transmitter side.

I/Q diagrams are particularly useful because they mirror the way most digital communications signals are created using an *I/Q* modulator. In the transmitter, *I* and *Q* signals are mixed with the same local oscillator (LO). A 90 degree phase shifter is placed in one of the LO paths. Signals that are separated by 90 degrees are also known as being orthogonal to each other or in quadrature. Signals that are in quadrature do not interfere with each other. They are two independent components of the signal. When recombined, they are summed to a composite output signal. There are two independent signals in *I* and *Q* that can be sent and received with simple circuits. This simplifies the design of digital radios. The main advantage of *I/Q* modulation is the symmetric ease of combining independent signal components into a single composite signal and later splitting such a composite signal into its independent component parts.

6.8.2.2 I and Q in a radio receiver

The composite signal with magnitude and phase (or *I* and *Q*) information arrives at the receiver input. The input signal is mixed with the local oscillator signal at the carrier frequency in two forms. One is at an arbitrary zero phase. The other has a 90 degree phase shift. The composite input signal (in terms of magnitude and phase) is thus broken into an in-phase, *I*, and a quadrature, *Q*, component. These two components of the signal are independent and orthogonal. One can be changed without affecting the other. Normally, information cannot be plotted in a polar format and reinterpreted as rectangular values without doing a polar-to-rectangular conversion. This conversion is exactly what is done by the in-phase and quadrature mixing processes in a digital radio. A local oscillator, phase shifter, and two mixers can perform the conversion accurately and efficiently.

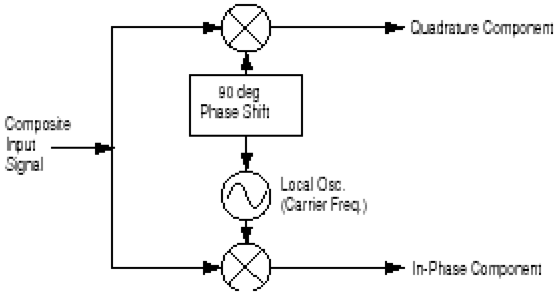


Figure 6.18: *I* and *Q* in a Radio Receiver

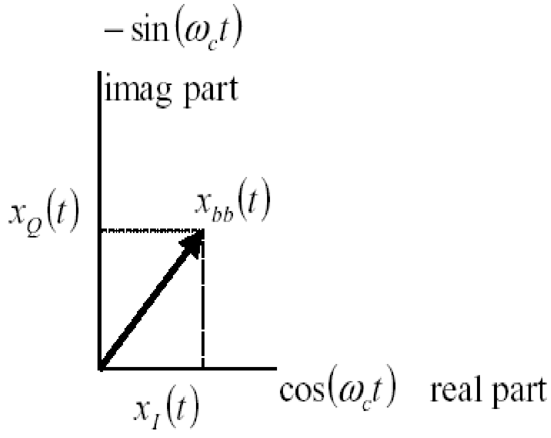


Figure 6.19: Decomposition of baseband-equivalent signal.

Here we have to write about the real and imaginary part.

$$x_I\ (t) = a\ (t) \cos\ (\theta(t)) \qquad (6.9)$$

is the time-varying inphase component of the modulated and

$$x_Q\ (t) = a\ (t) \sin(\theta(t)) \qquad (6.10)$$

is the time-varying quadrature component.

Relationships determining (a(t), ω(t)) from (xI (t), xQ(t)) are

$$a(t) = \sqrt{x_I^2(t) + x_Q^2(t)} \quad ,$$

$$\theta(t) = \mathrm{Tan}^{-1}\left[\frac{x_Q(t)}{x_I(t)}\right] \quad .$$

In passband processing and analysis, the objective is to eliminate explicit consideration of the carrier frequency .c and directly analyze systems using only the inphase and quadrature components. These inphase and quadrature components can be combined into a two-dimensional vector, or into an equivalent complex signal. By convention, a graph of a quadrature-modulated signal plots the inphase component along the real axis and the quadrature component along the imaginary axis as shown in Figure 2.3. The resultant complex vector xbb(t) is known as the complex baseband-equivalent signal.

6.8.3 Why use *I* and *Q*?

Digital modulation is easy to accomplish with *I/Q* modulators. Most digital modulation maps the data to a number of discrete points on the *I/Q* plane. These are known as constellation points. As the signal moves from one point to another, simultaneous amplitude and phase modulation usually results. To accomplish this with an amplitude modulator and a phase modulator is difficult and complex. It is also impossible with a conventional phase modulator. The signal may, in principle, circle the origin in one direction forever, necessitating infinite phase shifting capability. Alternatively, simultaneous AM and Phase Modulation is easy with an *I/Q* modulator. The *I* and *Q* control signals are bounded, but infinite phase wrap is possible by properly phasing the *I* and *Q* signals.

6.8.4 Constellation diagrams

A twodimensional diagram of the carrier magnitude and phase (a standard polar plot) can be represented differently by superimposing rectangular axes on the same data and interpreting the carrier in terms of in-phase (*I*) and quadrature-phase (*Q*) components.
It would be possible to perform AM and PM on a carrier at the same time and send data this way; it is easier for circuit design and signal processing to generate and detect a rectangular, linear set of values (one set for *I* and an independent set for *Q*).

The constellation diagram shows a repetitive “snapshot” of that same burst, with values shown only at the decision points. The constellation diagram displays phase errors, as well as amplitude errors, at the decision points. The transitions between the decision points affects transmitted bandwidth. This display shows the path the carrier is taking but does not explicitly show errors at the decision points. Constellation diagrams provide insight into varying power levels, the effects of filtering, and phenomena such as Inter-Symbol Interference.

The relationship between constellation points and bits per symbol is $M=2^n$ *where* M = *number of constellation points n = bits/symbol or n = log2 (M)*

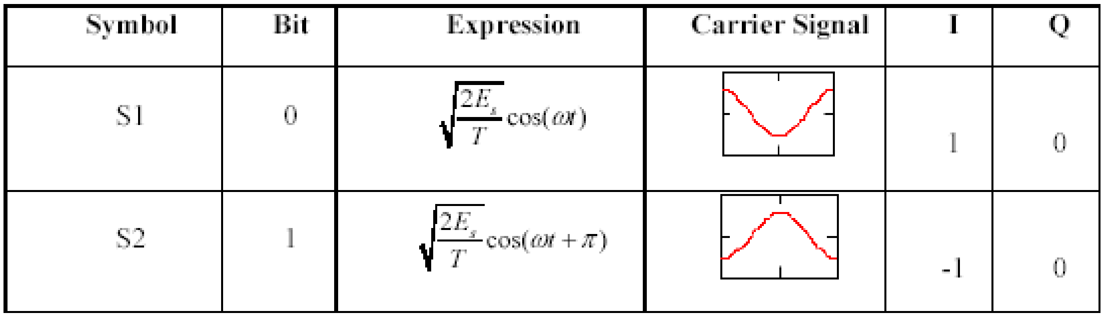
6.9 Binary Phase Shift Keying (BPSK)

In signal space, BPSK has a one -dimensional signal constellation with 2 message points. Each message symbol represents one bit. Signaling states are at 0 and 180 degrees. We utilize just one sinusoid as the basis function. We vary the phase of this signal to transmit information Each symbol is signaled by a change in phase In BPSK we define two little packets of the cosine wave, one with zero phase and second one with a 180 degree different phase. Table 1 lists the two symbols and the signals used to represent them. (The carrier signal shown is for f = 1.) The I and Q amplitudes are the x and y projections computed as follows.

I amplitude = (symbol expression x cos(phase)

Q $_{\text{amplitude}}$ = (symbol expression x cos(phase)

From this we get, I = 1 for the first symbol and -1 for the second symbol. Q amplitude is zero for both symbols because sin of both 0□ and 180□is zero.



Symbol	Bit	Expression	Carrier Signal	I	Q
S1	0	$\sqrt{\frac{2E_s}{T}}\cos(\omega t)$		1	0
S2	1	$\sqrt{\frac{2E_s}{T}}\cos(\omega t+\pi)$		-1	0

Table 6.1 . Mapping rules for BPSK which uses two signals, both a variation of a cosine signal

As energy of a signal is equal to

$$E_s = \frac{A^2T}{2}$$

So instead of writing a amplitude term to make the expression general, we write the equation in terms of energy, where $A = \sqrt{2E_s/T}$.

When referring to carrier signals, we typically talk in terms of signal or bit energy, so it makes sense to write the equations in terms of energy, which is what this scaling factor is.

6.10 BPSK Mapping:

To send a bit sequence 0111 0101 0010 1011 using BPSK signaling Technique, we need 16 symbols since each BPSK symbol stands for one bit. These are s1 s2 s2 s2 s1 s2 s1 s2 s1 s1 s2 s1 s2 s1 s2 s2 This is followed by stringing together the appropriate symbol signal packets from Table 1 in the right order. Figure below is the modulated carrier that would be transmitted for this sequence if we use the mapping in Table I.

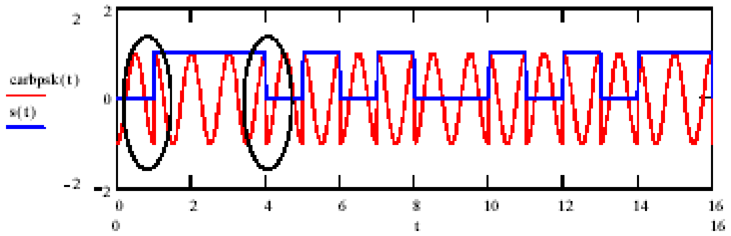


Figure 6.20 - A BPSK signal for bit sequence 0111 0101 0010 1011

Figure 11 is seen once the modulated carrier is caught and looked at it on a network analyzer, However, the above picture is at a carrier frequency of 1 Hz, which is not realistic. In real systems, the carrier frequency is very high and we would see a signal that covers a lot of cycles between each transition.

Importance of Transition:

A transition is the time at which we switch from one symbol to the next. What happens at the transition boundary is different for various modulations and is quite an important thing. In the case of BPSK, at every bit transition the signal does a 180 degree phase shift.

We worry about what the signal does at transitions because of amplifier non- linearities. Amplifiers used in communications have a very hard time with sudden changes in signal amplitudes and introduce distortions. Since this makes it harder to decode the symbol, we try to control these transitions.

6.11 Quaternary Phase Shift Keying (QPSK)

In signal space, QPSK has a two-dimensional signal constellation with 4 message points.

Each message symbol represents 2 bits, one that modulates the in-phase carrier and a second that modulated the quarature carrier. Signaling states are at ± 45° and ± 135° and the receiver thresholds are set at 0°, 90°, 180° and 270°. The bits (*ak* , *bk*) are assigned in a Grey code to the four signal points as: 11 to 45°, 01 to 135°, 00 to –135° and 10 to –45°. A received symbol error at any threshold results in only one bit error. In this assignment, *ak* modulates the in-phase carrier while bit *bk* modulates the quadrature carrier. The signal point constellation of QPSK is illustrated in figure

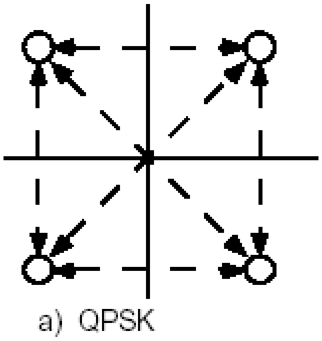


Figure 6.21 Signal point constellations for QPSK,

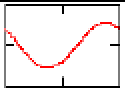
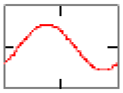
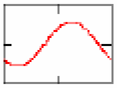
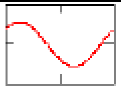
Symbol	Bits	Expression	Phase, (Deg.)	Carrier Signal	I	Q
S1	00	$\sqrt{\frac{2E_s}{T}}\cos(\omega t+\pi/4)$	45		$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
S2	01	$\sqrt{\frac{2E_s}{T}}\sin(\omega t+3\pi/4)$	135		$-\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
S3	11	$\sqrt{\frac{2E_s}{T}}\cos(\omega t+3\pi/4)$	225		$-\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$
S4	10	$\sqrt{\frac{2E_s}{T}}\sin(\omega t+\pi/4)$	310		$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$

Table 6.2 . Mapping rules for QPSK

In QPSK the four symbol definitions, written in sine or a cosine, can be decomposed further so we can compute the I and Q channel amplitudes. For example for the symbol S1.

Using this trigonometric identity,

$$\cos(x+y)=\cos x\cos y-\sin x\sin y \tag{6.12}$$

‘

We can write this equivalent expression

$$\begin{aligned}\sqrt{\frac{2E_s}{T}}\cos(\omega t+\pi/4)&=\sqrt{\frac{2E_s}{T}}\left(\cos\omega t\cos\pi/4-\sin\omega t\sin\pi/4\right)\\&=\sqrt{\frac{E_s}{T}}(\cos\omega t-\sin\omega t)\end{aligned} \tag{6.13}$$

We see that the little packet of carrier signal representing a particular symbol can be created by a free running sine and a cosine wave of certain amplitudes. This makes hardware realization possible.

6.12 CONSTELLATION OF QPSK:

Figure no. is created by plotting the values of I and Q amplitudes shown in Table 3. Each point is a pair of (I, Q) values representing a modulated signal or symbol. These I and Q values are computed by multiplying the signal expression (its amplitude) by sine or cosine of the phase angle.

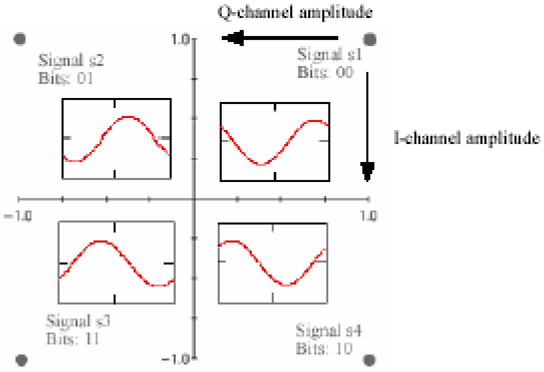


Figure 6.22. Constellation points are the tips of the modulating signal

This constellation diagram is created just prior to combining both I and Q into a composite signal. Constellation diagrams are always at baseband, that is at carrier frequency equal to zero. Eye diagrams similarly are also at baseband and show the same information but in time-domain. The constellation diagram of Figure 6.23 is a perfect square because the channel has not suffered any degradation due to the amplifier or the medium. When a constellation diagram is created at the receiver, the picture is not so perfect looking and can tell us the type of distortions experienced by the signal. The pulse shaping also changes this perfect square constellation.

To transmit the bit sequence: 11 00 11 10 00 10 00 01 00 . using QPSK.

Step 1 . Convert bit sequence to a polar signal (convert 0 to -1)

1 1 -1-1 1 1 1-1 -1-1 1-1 -1-1 -1 1 -1-1

Step 2 . Send alternating bits to I and Q channels

I 1 -1 1 1 -1 1 -1 -1 -1 -1 -1 1 -1 -1 1

Q 1 -1 1 -1 -1 -1 -1 1 -1 -1 -1 1 1 -1 -1

Set the amplitude of I and Q based on Table 3. (The following have been scaled to 1.0 instead of .707)

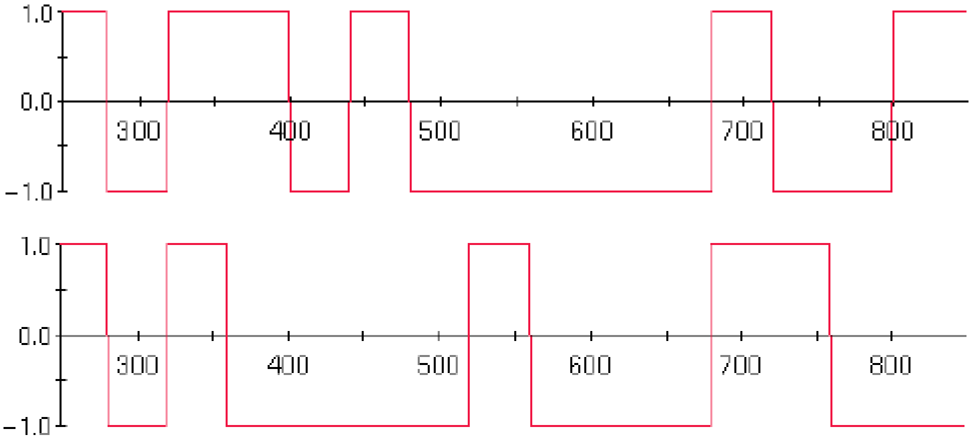


Figure 6.23 The bit stream mapped to I and Q channel levels

Multiply I channel with a cosine of frequency fc (fc = 1 in this example) We note that the signal transitions at symbol bit boundary solely because of I channel amplitude changes.

Multiply Q channel with the same free running cosine but shift it first by 90° so it is a sine wave. This way we can use just one piece of hardware to get both a sine and a cosine signal. We get the following two modulated carrier signals.

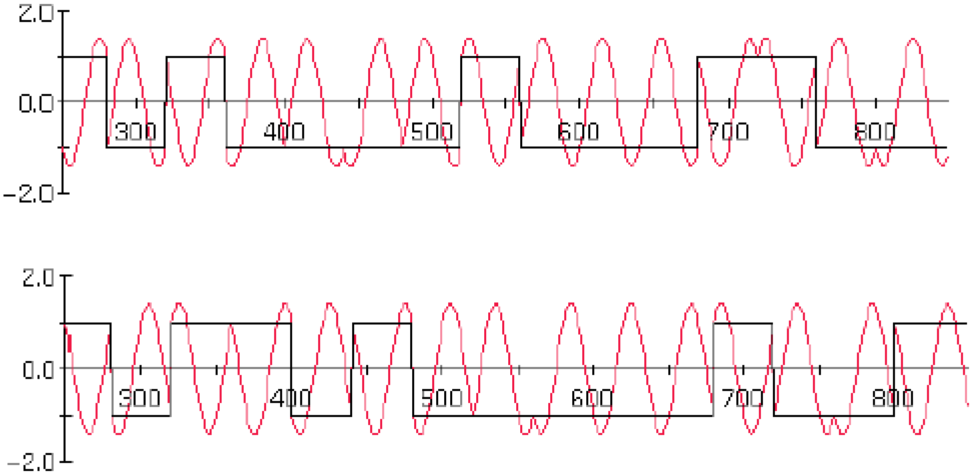


Figure 6.24 . The I channel multiplied by a cosine of frequency =1 and the Q channel multiplied by a sine of frequency =1

Step 4 .by Adding I and Q channels we get the transmitted carrier.

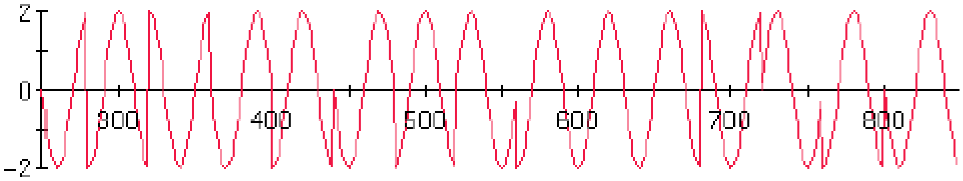
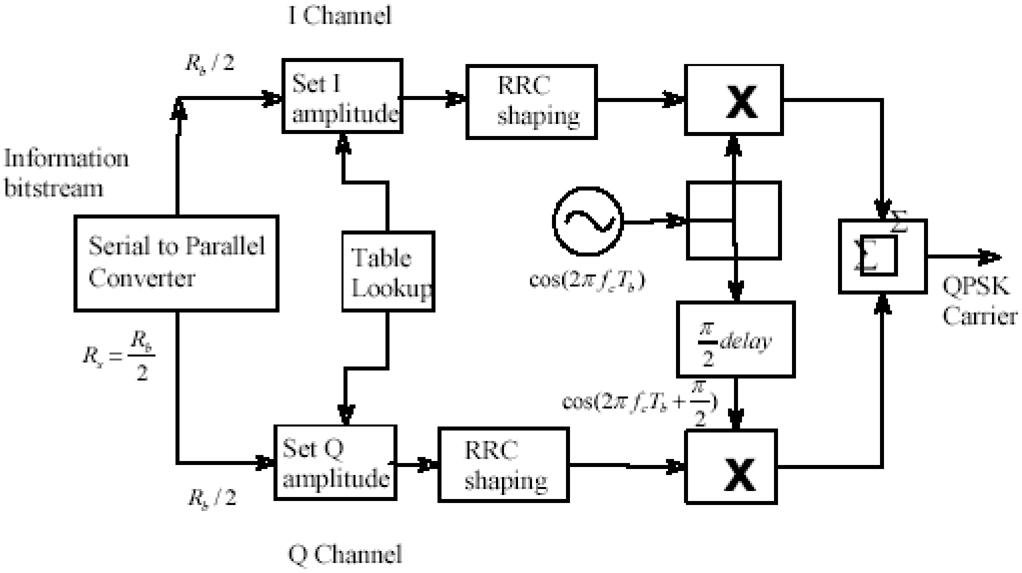


Figure 6.25 . Sum I and Q channels to create the composite transmitted signal

The following diagram shows how we do the modulation in Hardware



Summary: The Serial to parallel converter takes the bit stream coming in at a bit rate of Rb and splits it into two streams, each of half the bit rate. Depending on the dual bit pattern coming in, I and Q amplitudes are set from a table lookup function. Each of these is then individually modulated by a sine or a cosine wave of carrier frequency ù after being shaped into a root raised cosine pulse. These are added together to get the transmitted signal.

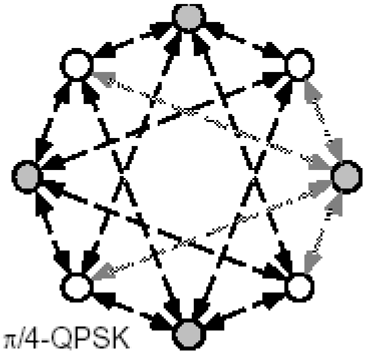
How QPSK differs from BPSK: QPSK has two basis signals. It can be considered to be composed of two BPSK signals each of one half the bit rate.

6.13 Issue of Dimensionality:

QPSK uses two basis functions, a sine and a cosine whereas BPSK uses just one. By varying the phase of each of these carriers (in the ship example, the position) we can send two bits per each signal. The dimensionality of a modulation is defined by the number of basis functions used. That makes QPSK a two- dimensional signal. Not because it sends two bits per symbol, but because it uses **two** independent signals (a sine and a cosine) to create the symbols.

Pi/4 QPSK

Like QPSK, π/4-QPSK transmits two bits per symbol. So only four carrier signals are needed but this is where the twist comes in. In QPSK we have four signals that are used to send the four two- bit symbols. In π/4-QPSK we have eight signals, every alternate symbol is transmitted using a π/4 shifted pattern of the QPSP constellation. Symbol A uses a signal on Path A as shown below and the next symbol, B, even if it is exactly the same bit pattern uses a signal on Path B. So we always get a phase shift even when the adjacent symbols are exactly the same.



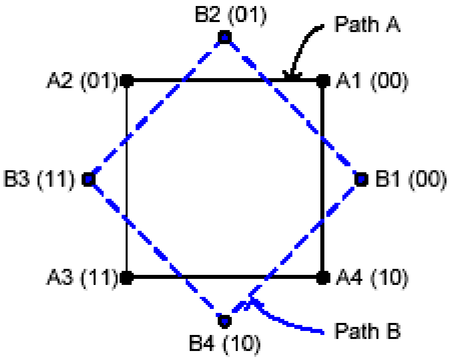


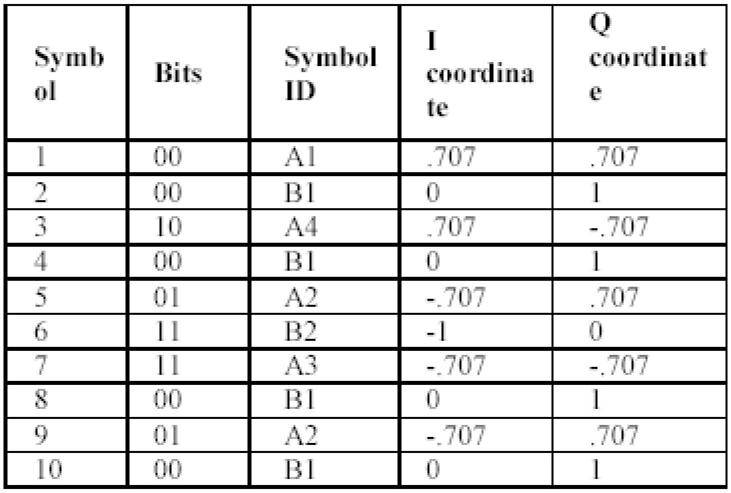
Figure 6.26 Signal point constellations for π/4 QPSK

ffig 41

We divide the bit sequence into 2-bit pieces just as we would do for QPSK

Bit sequence: 00 00 10 00 01 11 11 00 01 00

Transmit the first symbol using the A constellation shown in Figure 41 and the next symbol uses the B constellation. For each 2-bit, the I and Q values are the signal coordinates as shown below.



Symbol	Bits	Symbol ID	I coordinate	Q coordinate
1	00	A1	.707	.707
2	00	B1	0	1
3	10	A4	.707	-.707
4	00	B1	0	1
5	01	A2	-.707	.707
6	11	B2	-1	0
7	11	A3	-.707	-.707
8	00	B1	0	1
9	01	A2	-.707	.707
10	00	B1	0	1

Table 6.3 - π/4-QPSK symbols mapping to I and Q

The I and Q channels for a **π/4-QPSK signal** are shown below in Figure 42. Note that there are five possible levels (1, .707. 0 -.707, -1) and I and the Q channel show this variation in response to the symbols.

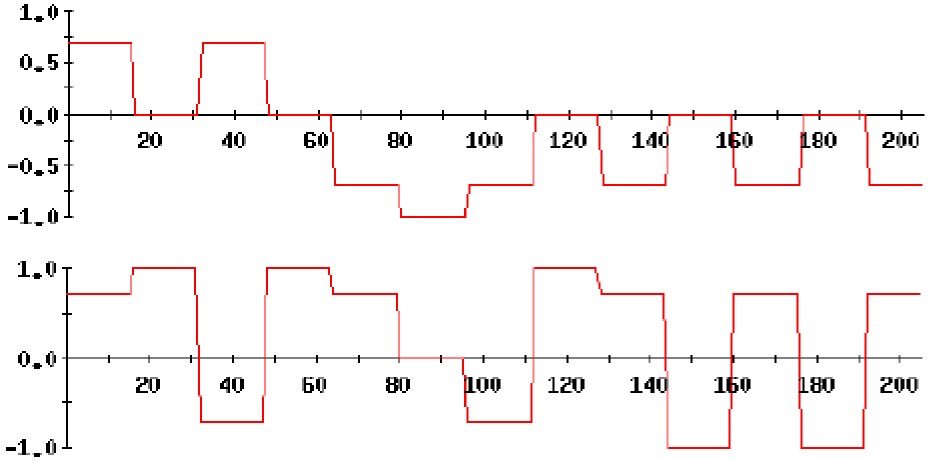


Figure 6.27 . I and Q mapping of π/4-QPSK symbols

Step 1 . Map bits to symbols

Bits	00	00	10	00	01	11	11	00	01	00
Symbols	A1	B1	A4	B1	A2	B2	A3	B1	A2	B3

Step 2 - Multiply the I and Q with a carrier (in the example below, the carrier frequency is 1 Hz.) and you get an 8-PSK signal constellation.

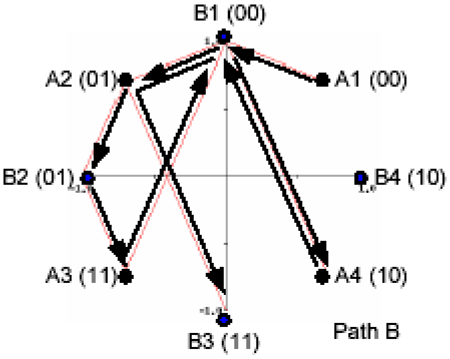


Figure 6.28 . of π/4-QPSK symbols traverse over a 8-PSK constellation

The constellation diagram is a path that the symbols have traced in time as we can see in the above diagram of just the symbols of this signal. The path stars with symbol A1, then goes to B1 which is on path B. From here, the next symbol A2 is back on Path A. Each transition, we see above goes back and forth between Path A and B.

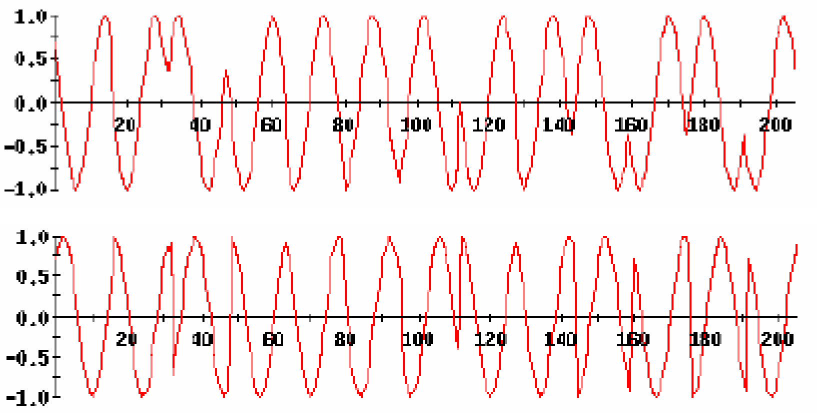


Figure 6.29 π/4-QPSK modulated I and Q Channels

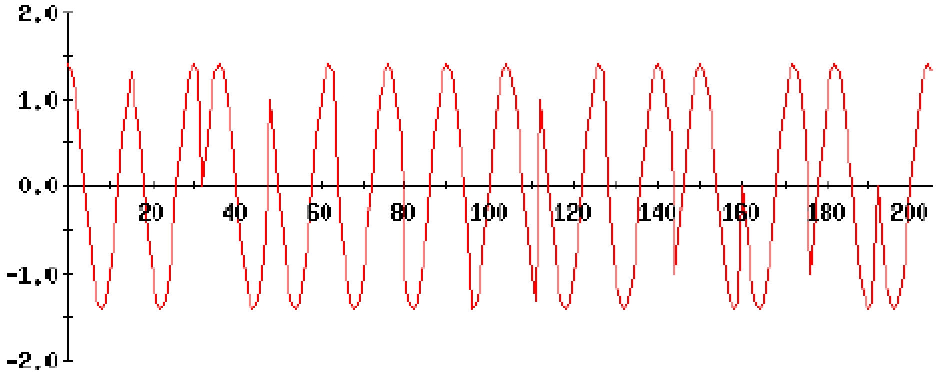


Figure 6.30 . π/4-QPSK modulated carrier

the advantage of doing this? On the average, the phase transitions are somewhat less than a straight QPSK and this does two things, one is that the side lobes are smaller so less adjacent carrier interference. Secondly the response to Class C amplifiers is better. This modulation is used in many mobile systems.

6.14 Constant Envelope modulation

QPSK is part of a class of signals called **constant-envelope** signals. There is no rigorous definition of a constant envelope signal. One definition is; when sampled at the symbol rate, the sampled value of the amplitude is constant. Another is that there are no discontinuous phase changes. Yet another is that the maximum and minimum amplitude attained by the signal over one period is constant. The sine wave is an ideal constant envelope signal.

Constant envelope signals suffer less distortion in high power amplifiers and are preferred for wireless applications. The reason is that amplifiers work by changing a signal's amplitude, either increasing or decreasing it. To increase a signal.s power is to increase its amplitude. A non-linear amplifier changes the signal amplitude by differing amount depending on the instantaneous amplitude of the signal. The more the amplitude of a signal varies, the more non-linear amplification occurs and this results in a distorted signal. QPSK is not technically a constant envelope because of its discontinuous phase shifts but is considered nearly so.

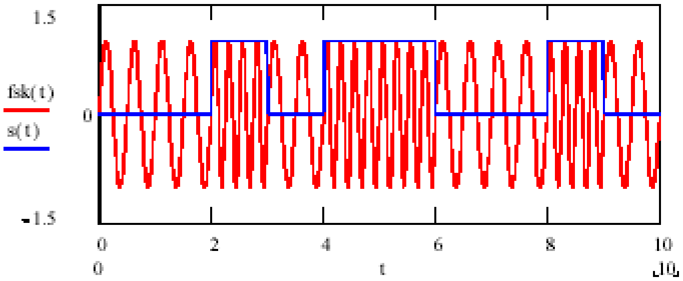


Figure 6.31 . FSK is definitely a constant envelope modulation.

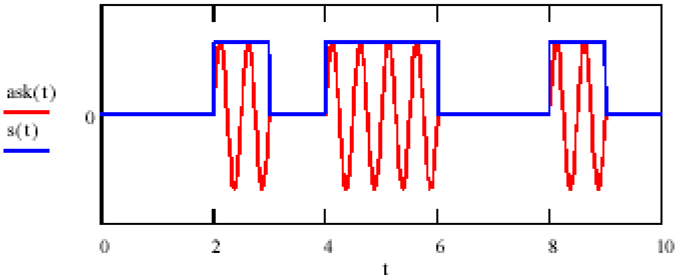


Figure 6.32 - ASK is definitely not a constant envelope modulation

6.15 Power Spectrum and Power Spectral Density

Another way to look at the Power is to look its spectrum. The spectrum of a signal shows how much power is contained in each of its harmonic or spectral components or the frequency spectrum of the signal. A sine wave PSD has only a delta function at the carrier frequency since the signal contains just one spectral component namely the carrier frequency. A random signal on the other hand has a rich PSD with power occurring in many different components. Power spectrum of white noise is flat to show that power occurs in all frequencies.

A plot of the frequency components on the x-axis and attendant Power in that frequency on the y-axis is called the Power Spectrum of the signal. Its units are watts per Hertz. The Power Spectrum is also referred to as the Power Spectral Density. The two terms refer to the same thing. The Power spectrum does not directly give us the total or average power in the signal, only power in a particular spectral component. To obtain the total power in the signal or in a particular range, we must integrate the Power Spectrum over the range of frequencies of interest and include both negative and positive frequencies.

We can define total power now as the integral of the Power Spectral Density.

$$P_x = \frac{2}{2\pi}\int_0^\infty S_x(\omega)d\omega \qquad (6.14)$$

and

$$\hat{S} = 2S_x(\omega) \qquad (6.15)$$

where Sω is called the two-sided spectral density. We multiplied it by 2, because our integration limits were only the positive frequencies.

6.16 Properties of PSD and Power Spectrum

1. The total area under the Power Spectrum or PSD is equal to the total avg. power of the signal.

2. The PSD is always positive.

3. The PSD is an even function of frequency or in other words, it is symmetrical.

4. The auto-correlation function and PSD are a Fourier transform pair. To compute PSD, we compute the auto-correlation of the signal and then take its FFT. (Another estimation method called "periodogram" uses sampled FFT to compute the PSD.)

1. 5. The value of the auto-correlation function at zero-time equals the total power in the signal.

Keep in mind that total or the average power in a signal is often not of as great an interest. We are most often interested in the PSD or the Power Spectrum. We often want to see is how the input power has been redistributed by the channel and in this frequency-based redistribution of power is where most of the interesting information lies.

6.17 Periodogram

Periodogram is a computationally economically way of estimating the Power Spectrum. Mathematically we see that one needs to compute the auto-correlation of the whole sequence in order to get an accurate PSD. But for large sequences, this takes too long and a averaged PSD is computed instead. This averaged PSD is referred to as the Periodogram.

Periordogram method of computing the power spectrum also makes sense when the signal FFT is very noisy and a desired signal level can not be. In such cases, the inherent averaging of the Periodogram can help extract the signal.

DSP ARCHITECTURE

7.1 INTRODUCTION

The TMS320C6000 platform of digital signal processors (DSPs) is part of the TMS320 family of DSPs. The TMS320C62x (’C62x) devices are fixed-point DSPs in the TMS320C6000 platform. The TMS320C67x (’C67x) devices are floating-point DSPs in the TMS320C6000 platform. The TMS320C62x and TMS320C67x are code compatible and both feature the VelociTIE architecture. The VelociTI architecture is a high-performance, advanced, very-long-instruction-word (VLIW) architecture developed by Texas Instruments, making these DSPs excellent choices for multichannel and multifunction applications. VelociTI, together with the development tool set and evaluation tools, provides faster development time and higher performance for embedded DSP applications through increased instruction-level parallelism.

The TMS320 family consists of 16-bit and 32-bit fixed- and floating-point devices. These DSPs possess the operational flexibility of high-speed controllers and the numerical capability of array processors. The following characteristics make this family the ideal choice for a wide range of processing applications:

_ Very flexible instruction set

_ Inherent operational flexibility

_ High-speed performance

_ Innovative, parallel architectural design

_ Cost-effectiveness

7.1.1 History, Development, and Advantages of TMS320 DSPs

In 1982, Texas Instruments introduced the TMS32010 — the first fixed-point DSP in the TMS320 family. Before the end of the year, the Electronic Products magazine awarded the TMS32010 the title “Product of the Year”. The TMS32010 became the model for future TMS320 generations. Today, the TMS320 family consists of three supported platforms including the TMS320C2000, TMS320C5000, and TMS320C6000. Within the ’C6000 platform there are two generations, the TMS320C62x and TMS320C67x, with performance and features that are reflective of Texas Instruments’ commitment to lead the world in DSP solutions.

Each generation of TMS320 devices uses a core central processing unit (CPU) that is combined with a variety of on-chip memory and peripheral configurations. These various configurations satisfy a wide range of needs in the worldwide electronics market. When memory and peripherals are integrated with a CPU into one chip, the overall system cost is greatly reduced, and circuit board space is reduced. Figure 1–1 shows the progression of the TMS320 family of devices.

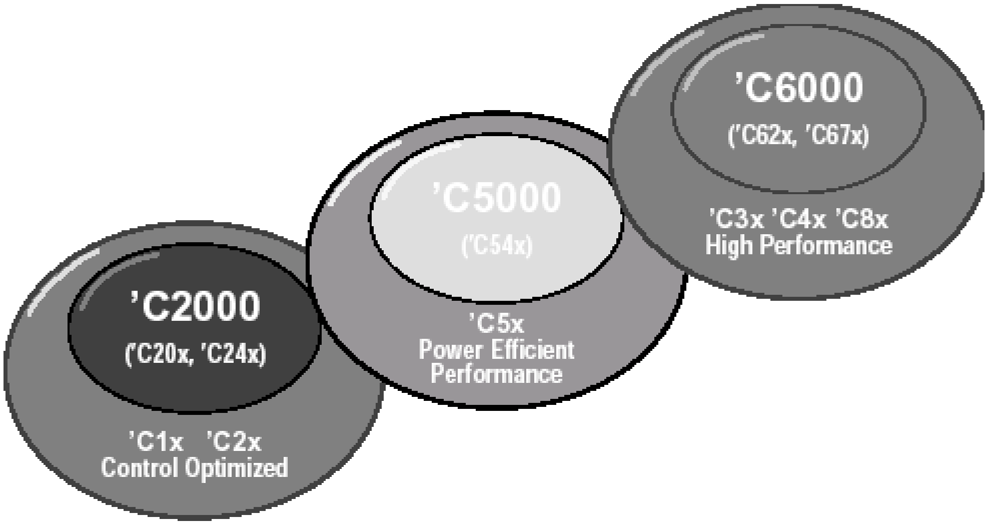


Figure 7.1 The TMS320 Family

Figure 7. 1 THE TMS320 FAMILY OF DIGITAL SIGNAL PROCESSOR

7.1.2 Typical Applications for the TMS320 Family

The TMS320 family of DSPs offers adaptable approaches to traditional signal processing problems, such as vocoding, filtering, and error coding. Furthermore, the TMS320 family supports complex applications that often require multiple operations to be performed simultaneously.

7.2 Introduction to the TMS320C6000 Platform of Digital Signal Processors

With performance of up to 2000 million instructions per second (MIPS) at 250 MHz and a complete set of development tools, the TMS320C6000 DSPs offer cost-effective solutions to high-performance DSP programming challenges. The TMS320C6000 development tools include a new C compiler, an assembly optimizer that simplifies programming and scheduling, and a WindowsE debugger interface.

The TMS320C6000 DSPs give system architects unlimited possibilities to differentiate their products. High performance, ease of use, and affordable pricing make the TMS320C6000 platform the ideal solution for multichannel, multifunction applications, such as:

_ Pooled modems

_ Wireless local loop base stations

_ Beam-forming base stations

_ Remote access servers (RAS)

_ Digital subscriber loop (DSL) systems

_ Cable modems

_ Multichannel telephony systems

_ Virtual reality 3-D graphics

_ Speech recognition

_ Audio

_ Radar

_ Atmospheric modeling

_ Finite element analysis

_ Imaging (examples: fingerprint recognition, ultrasound, and MRI)

The TMS320C6000 platform is also an ideal solution for exciting new applications; for example:

_ Personalized home security with face and hand/fingerprint recognition

_ Advanced cruise control with global positioning systems (GPS) navigation and accident avoidance

_ Remote medical diagnostics

7.3 CPU ARCHITECTURE

The Velocity architecture makes the ’C6000 DSPs the first off-the-shelf DSPs to use an enhancement of traditional VLIW to achieve high performance through increased instruction-level parallelism. A traditional VLIW architecture consists of multiple execution units running in parallel that perform multiple instructions during a single clock cycle. Parallelism is the key to extremely high performance and takes these next-generation DSPs well beyond the performance capabilities of traditional superscalar designs. Velocity is a highly deterministic architecture, with few restrictions on how or when instructions are fetched, executed, or stored. This architectural flexibility is key to the breakthrough efficiency levels of the ’C6000 compiler.

7.3.1 CENTRAL PROCESSING UNIT (CPU)

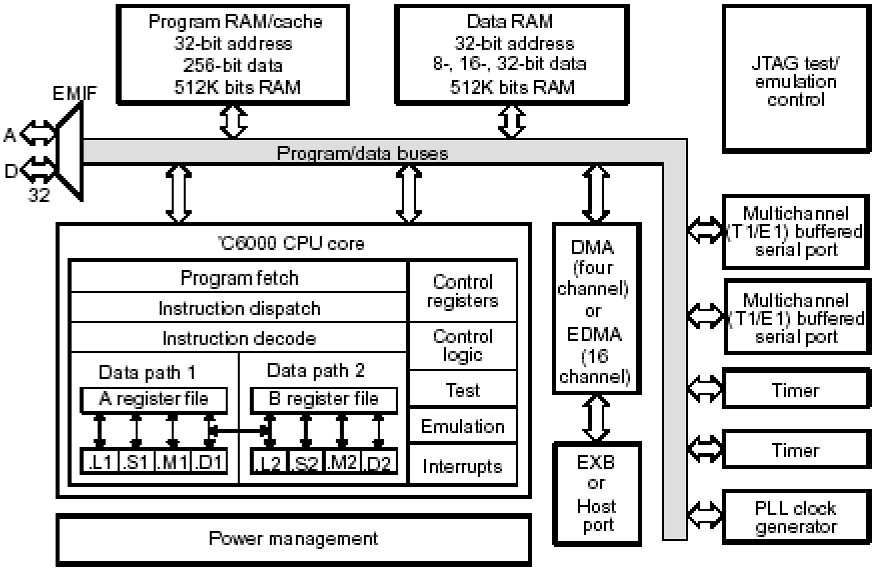


Figure 7.2. TMS320C62x/C67x Block Diagram

The ’C62x/C67x CPU, in Figure 2–1, is common to all the ’C62x/C67x devices. The CPU contains:

_ Program fetch unit

_ Instruction dispatch unit

_ Instruction decode unit

_ 32 32-bit registers

_ Two data paths, each with four functional units

_ Control registers

_ Control logic

_ Test, emulation, and interrupt logic

The CPU has two data paths (A and B) in which processing occurs. Each data path has four functional units (.L, .S, .M, and .D) and a register file containing 16 32-bit registers. The functional units execute logic, shifting, multiply, and data address operations. All instructions except loads and stores operate on the registers. The two data-addressing units (.D1 and .D2) are exclusively responsible for all data transfers between the register files and memory.

The four functional units of a data path have a single data bus connected to registers on the other side of the CPU so that the units can exchange data with the register file on the opposite side. Register access across the CPU supports one read and write operation per cycle. The two sets of functional units include the following items:

_ Two multipliers

_ Six arithmetic logic units (ALUs)

_ Two register files, each containing 16 32-bit registers

Each functional unit is controlled by a 32-bit instruction. The instruction fetch, instruction dispatch, and instruction decode blocks can deliver up to eight 32-bit instructions from the program memory to the functional units every cycle. The control register file provides methods to configure and control various aspects of processor operation. Access to the control registers is provided from datapath B.

The VLIW processing flow begins when a 256-bit-wide instruction fetch packet (IFP) is fetched from the internal program memory. The instructions linked together for simultaneous execution (up to eight instructions) form an execute packet. For more details on the processing, see the data sheet for your particular device.

7.4 ADDRESSING MODES

The addressing modes on the ’C62x and ’C67x are linear by default, but circular addressing is available. The mode is specified by the addressing mode register (AMR).

All registers can perform linear addressing. Only eight registers can perform circular addressing: A4–A7 are used by the .D1 unit and B4–B7 are used by the .D2 unit. No other units can perform circular addressing. **LDB/LDH/LDW**, **STB/STH/STW**, **ADDAB/ADDAH/ADDAW**, and **SUBAB/SUBAH/SUBAW** instructions all use the AMR to determine what type of address calculations are performed for these registers. The ’C62x/C67x CPU has a load/store architecture, which means that the only way to access data in memory is with a load or store instruction.

7.5 MEMORY

The TMS320C6000 platform of devices includes on-chip memory for both program and data, some of which may be selected as cache. In addition, an external memory interface (EMIF) may be used to include external memories in a ’C6000 system.

7.5.1 External Memory Interface (EMIF)

The external memory interface (EMIF) connects the CPU and external memory, such as synchronous dynamic RAM (SDRAM), synchronous burst static RAM (SBSRAM), and asynchronous memory. The EMIF also provides 8-bit-wide and 16-bit-wide memory read capability to support low-cost boot ROM memories (flash, EEPROM, EPROM, and PROM). The EMIF supports high throughput interfaces to SDRAM, including burst capability.

7.6 PERIPHERALS

In addition to on-chip memory, the TMS320C62x and TMS320C67x devices contain peripherals for communication with off-chip memory, coprocessors, host processors, and serial devices. These peripherals are briefly described here, but each ’C6000 device has only a specific subset of them.

7.6.1 Direct Memory Access (DMA) Controller

The direct memory access (DMA) controller transfers data between regions in the memory map without intervention by the CPU. The DMA allows movement of data to and from internal memory, internal peripherals, or external devices to occur in the background of CPU operation. The DMA has four independently programmable channels allowing four different contexts for operation.

In addition, a fifth (auxiliary) channel allows the DMA to service requests from the host-port interface (HPI) or the Expansion Bus (XB). In discussing DMA operations, the following terms are important:

_ **Read transfer:** The DMA reads the data element from a source location in memory.

_ **Write transfer:** The DMA writes the data element that was read during a read transfer to its destination location in memory.

_ **Element transfer:** The combined read and write transfer for a single data element.

_ **Frame transfer:** Each DMA channel has an independently programmable number of elements per frame. In completing a frame transfer, the DMA moves all elements in a single frame.

_ **Block transfer:** Each DMA channel also has an independently programmable number of frames per block. In completing a block transfer, the DMA moves all frames it has been programmed to move. The DMA has the following features:

_ **Background operation:** The DMA operates independently of the CPU.

_ **High throughput:** Elements can be transferred at the CPU clock rate.

_ **Four channels:** The DMA can keep track of the contexts of four independent block transfers.

_ **Auxiliary channel:** This channel allows the host port to make requests into the CPU's memory space. This chapter discusses how the auxiliary channel requests are prioritized relative to other channels and the CPU.

Detailed explanation of how it is used in conjunction with a peripheral is found in that peripheral's documentation.

_ **Split operation:** A single channel may be used to simultaneously perform both the receive and transmit element transfers to or from two peripherals and memory, effectively acting like two DMAs.

_ **Multi-frame transfer:** Each block transfer can consist of multiple frames of a programmable size.

_ **Programmable priority:** Each channel has independently programmable priorities versus the CPU.

_ **Programmable address generation:** Each channel's source and destination address registers can have configurable indexes for each read and write transfer. The address may remain constant, increment, decrement, or be adjusted by a programmable value. The programmable value allows a distinct index for the last transfer in a frame and for the preceding transfers. This feature is used for multichannel sorting.

_ **Full-address 32-bit address range:** The DMA can access any region in the memory map:

_ The on-chip data memory.

_ The on-chip program memory when mapped into memory space.

_ The on-chip peripherals.

_ The external memory interface (EMIF).

_ **Programmable-width transfers:** Each channel can be independently configured to transfer either bytes, 16-bit halfwords, or 32-bit words.

_ **Autoinitialization:** Once a block transfer is complete, a DMA channel may automatically reinitialize itself for the next block transfer.

_ **Event synchronization:** Each read, write, or frame transfer may be initiated by selected events.

_ **Interrupt generation**: On completion of each frame transfer or of an entire block transfer, as well as on various error conditions, each DMA channel may send an interrupt to the CPU.

7.6.2 **Enhanced Direct Memory Access (EDMA)** Peripherals

The enhanced direct memory access (EDMA) controller, like the DMA controller, transfers data between regions in the memory map without intervention by the CPU. The EDMA allows movement of data to and from internal memory, internal peripherals, or external devices to occur in the background of CPU operation. The EDMA has sixteen independently programmable channels allowing sixteen different contexts for operation.

In addition to the features of the DMA controller, the EDMA also has the following features:

_ **Sixteen channels:** The EDMA can keep track of the contexts of sixteen independent transfers.

_ **Linking:** Each EDMA channel can be linked to a subsequent transfer to perform after completion.

_ **Event synchronization:** Each channel is initiated by a specific event. Transfers may be either synchronized by element or by frame.

7.6.3 **Host-Port Interface (HPI)** The Host-Port Interface (HPI) is a 16-bit wide parallel port through which a host processor can directly access the CPU's memory space. The host device functions as a master to the interface, which increases ease of access. The host and CPU can exchange information via internal or external memory. The host also has direct access to memory-mapped peripherals.

The HPI is connected to the internal memory via a set of registers. Either the host or the CPU may use the HPI Control register (HPIC) to configure the interface. The host can access the host address register (HPIA) and the host data register (HPID) to access the internal memory space of the device. The host accesses these registers using external data and interface control signals. The HPIC is a memory-mapped register, which allows the CPU access.

7.7 CODE COMPOSER STUDIO

The ("code Composer Studio (CCS) provides an integrated development environment (IDL:.) to incorporate the software tools. CCS includes tools lor code generation, such as a C compiler, an assemble!, and a linker. It has graphical capabilities and supports real-time debugging. It provides an easy-to-use software tool to build and debug programs.

The C compiler compiles a C' source program with extension *.c* to produce an assembly source file with extension . asm. The assembler assembles an . *a:^:r.* source file to produce a machine language object file with extension . ofoj.The linker combines object files and object libraries as input to produce an executable file with extension .out. This executable file represents a linked common object file form,it (COFF), popular in Unix-based systems and adopted by seveial maker, ol <li;>it.il signal processors |2I|. This executable file can be loaded and run directly on the C6711 processor.

To create an application project, one can "add" the appropriate files to the project. Compiler/linker options can readily be specified. A number of debugging features are available, including setting breakpoint[1] and watching variables, viewing memory, registers, and mixed C and assembly code, graphing results, and monitoring execution time. One can step through a program in different ways (step into..or over, or out).

Real-lime analysis can be performed using real-time data exchange (RJDX) associated with DSP/B1OS (Appendix G). RTDX allows lor data exchange between the host and the target and analysis in real time without stopping the- tamet. Key statistics and performance can be monitored in real time. Through the Joint 'learn Action Group (JTA(i), communication with on-chip emulation support occurs to control and monitor program execution. The C6711 DSK board includes a .If AC emulator interface.

IMPLEMENTATION

Test Bench of the Project

8.1 Analog to Digital Domain

An analog voice signal in the range of 0.3 to 3.5 KHz is input to the on board analog to digital converter AD535. AD535 works on Sigma Delta principle with a sampling rate of 8 KHz and resolution of 16 bit. Anti aliasing filter with a cut off frequency of 3.5 KHz at the input of the ADC limits the frequencies to the Nyquist criterion.

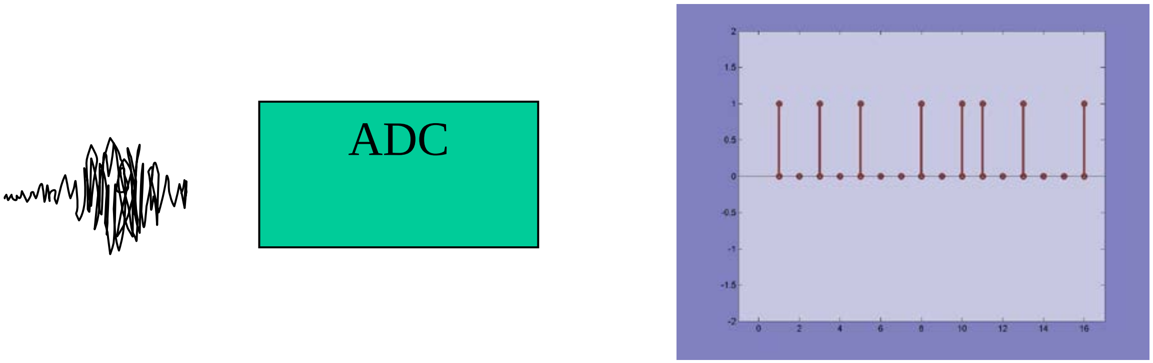


Figure 8.1 Analog to Digital Domain

Although the bit format of the ADC is 2's complement but the data available for the processing is in data type short (-2 ^ 15 to 2 ^ 15 -1 i.e -32768 to 32768). This data is converted into binary form using the integer to binary converter block of our design

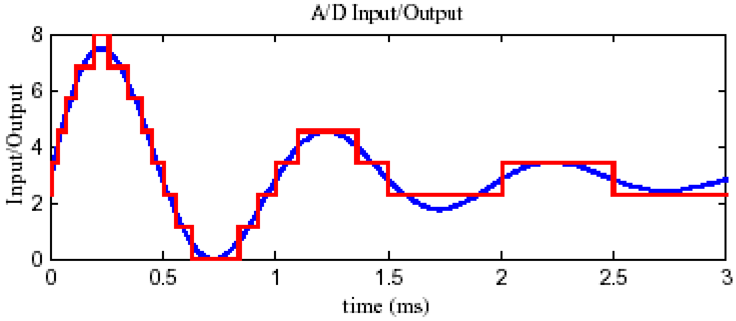


Figure 8.2: ADC's Input / Output

which recursively uses the following algorithm to generate the binary data from the data type short.

```
bit   =  temp & 1;
temp = temp >> 1;
```

Matlab simulation result of this block is as shown

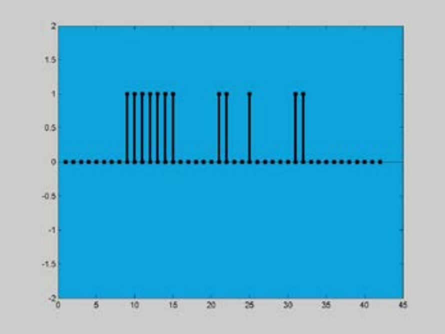


Figure 8.3: Pulse Code Modulated Data

Above data when fed to the DSP board the output of the processor as seen on the oscilloscope is as following

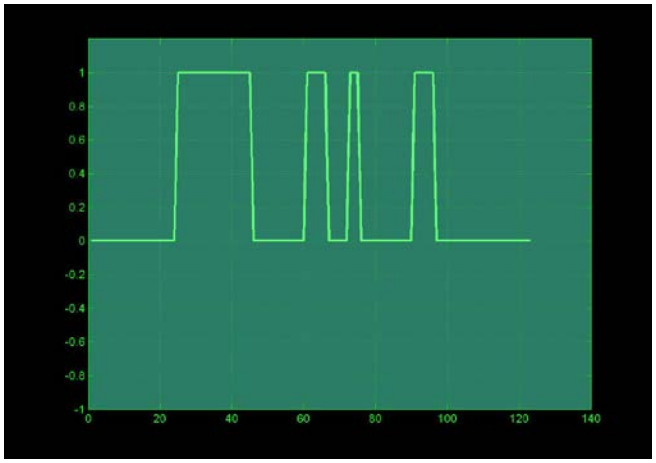


Figure 8.4 : Output of the Oscilloscope

8.2 **Source Coding**

After analysis of different techniques we came out with the conclusion to use ADPCM because of its low complexity algorithm and high quality signal.

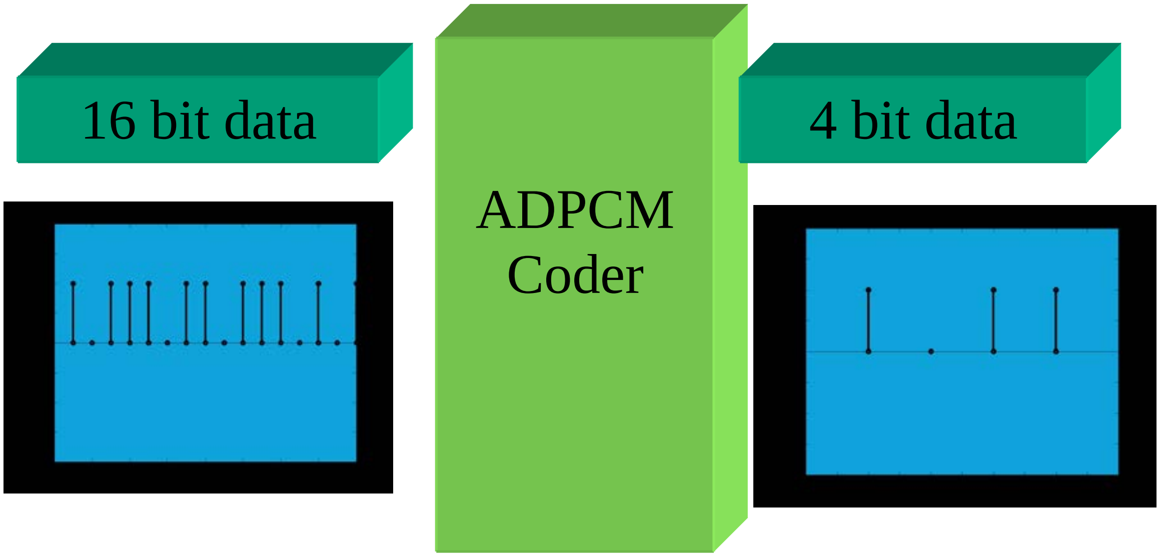


Figure 8.5 : Block diagram of ADPCM Coder

The design has already been discussed in the previous chapters however it is necessary to mention here that the algorithm [1] used for the ADPCM coding is as

```
let B3 = B2 = B1 = B0 = 0
if (d(n) < 0)
then B3 = 1
d(n) = ABS(d(n))
if (d(n) >= ss(n))
then B2 = 1 and d(n) = d(n) - ss(n)
if (d(n) >= ss(n) / 2)
then B1 = 1 and d(n) = d(n) - ss(n) / 2
if (d(n) >= ss(n) / 4)
then B0 = 1
L(n) = (1000$^2$ * B3)+(100$^2$ * B2)+(10$^2$ * B1) + B0
```

The algorithm for the decoding is as

```
d(n) = (ss(n)*B2)+(ss(n)/2*B1)+(ss(n)/4*BO)+(ss(n)/8)
if (B3 = 1)
then d(n) = d(n) * (-1)
X(n) = X(n-1) + d(n)
```

The input original signal and the reconstructed signals are shown as below we can see the reconstructed signal tends to follow the original signal but in cases of steeper slopes we get larger errors.

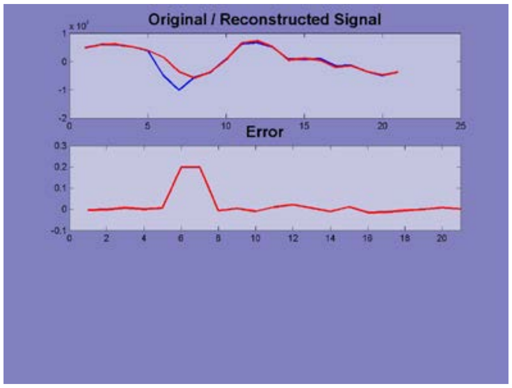


Figure 8.6 : Original / Reconstructed Signal

8.3 Binary to Polar Converter

The source coded data is fed into this block to be converted into the polar data in case of BPSK. However for Pi /4 QPSK we do not require this block because this is automatically achieved when we map the signal on to the constellation diagram.

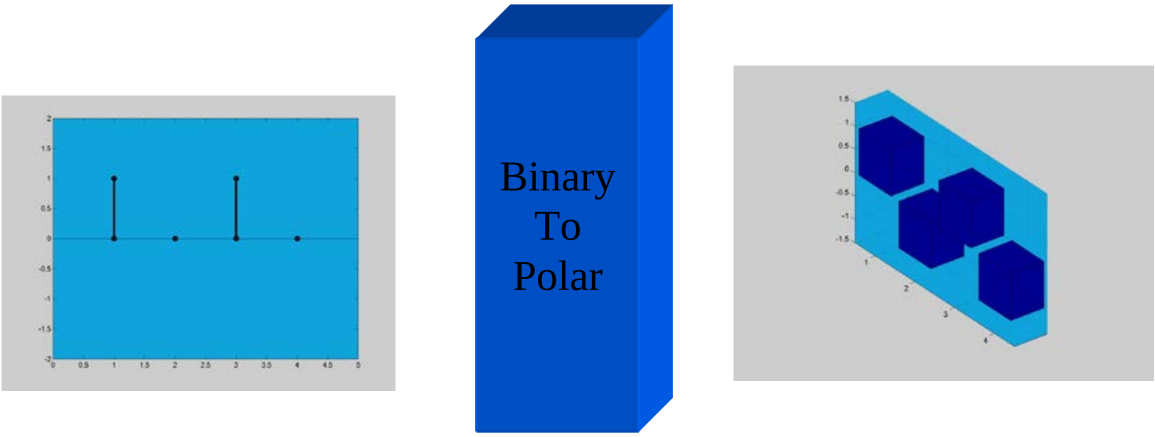


Figure 8.7 : Block diagram of Binary to Polar Converter

8.4 Pi / 4 QPSK Mapper

In case of Pi / 4 QPSK we used the following Look Up table to generate the two constellations.

LUT = [1 , j , -1 , -j
exp(j*pi*0.25), exp(j*0.75*pi), exp(-j*pi*0.25), exp(-j*.75*pi)];

A simulation in matlab was run and the two different constellations (A & B) are as following

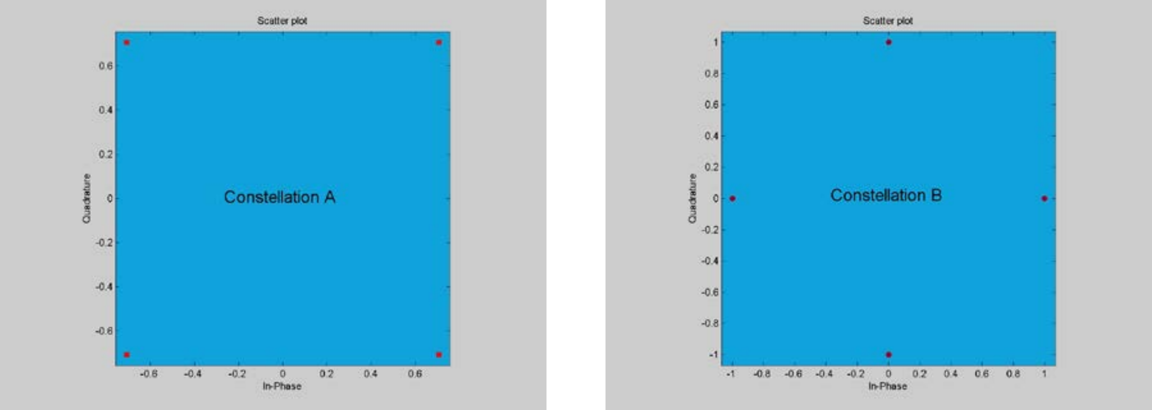


Figure 8.8 : Constellation Plot of Pi / 4 QPSK

8.5 Pulse Shaping

Pulse shaping is done with the use of raised cosine filter because it meets the zero ISI criterion. We have split it in two parts, root raised cosine filter implemented at transmitter and also at receiver to give the overall response of the raised cosine filter. Design parameters for the filter are:

Upsampling	=	2
Roll Off Factor	=	0.5
Group Delay	=	2

The impulse response of the filter as simulated by the matlab is as :

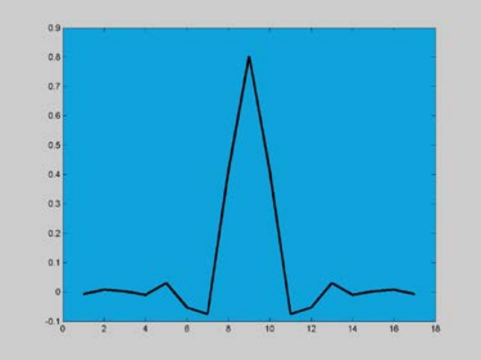


Figure 8.9 : Impulse Response of Root Raised Cosine Filter

Same filter coefficients are when input to the DSP board the output of the DSP board as seen on the oscilloscope is as :

Figure 8.10 : Output of the Oscilloscope

Random binary data was generated and filtered through the above filter. The matlab simulation results are as :

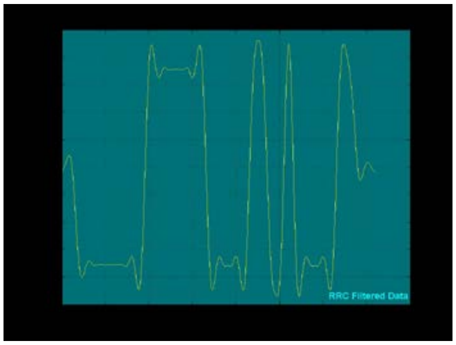


Figure 8.11 : Pulse Shaped Signal

Same data is processed through the processor and sent to the oscilloscope through DAC and it yields the following results.

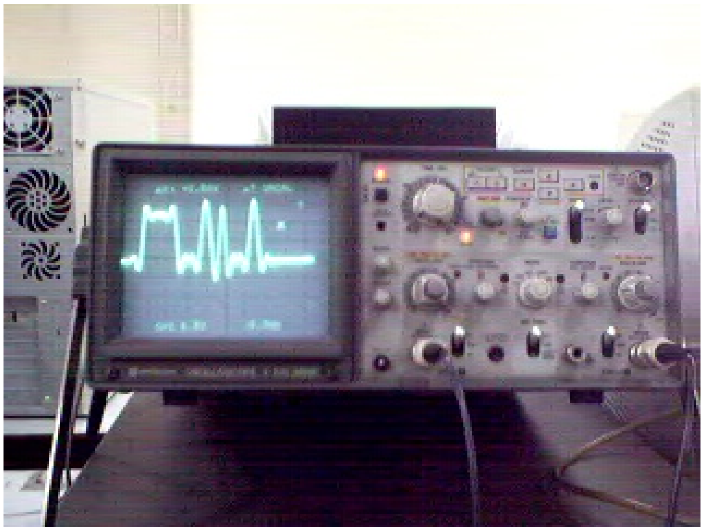


Figure 8.12 : Output of the Oscilloscope

8.6 **Match Filtering.** At the receiver end the signal is first matched filtered with the help of the half part of the raised cosine filter (i.e root raised cosine filter). This not only maximizes the SNR as the transmitter filter and the receiver filter exactly match in impulse response but also helps in the symbol timing recovery loop.

8.7 **Symbol Timing Recovery.** In this we have recovered the symbol timings so that we can sample at the symbol interval. This was the most difficult phase as with all digital communication systems. We tried Muller and Mueller algorithm, Early Late Method and Gardner Timing Error Detector. We also developed our own algorithm which proved to be better than above three. In case of Muller and Mueller we require three samples per symbol but in our case we could not go beyond two samples per symbol so the only choice was the Gardner. The problem with Gardner is that it takes about 600 samples to converge so it is not suitable for the systems where we are dealing with very small number of samples. So we designed our own synchronization scheme and our Project Supervisor has suggested us to write a paper on our synchronization scheme. The Matlab simulation results of our algorithm are as:

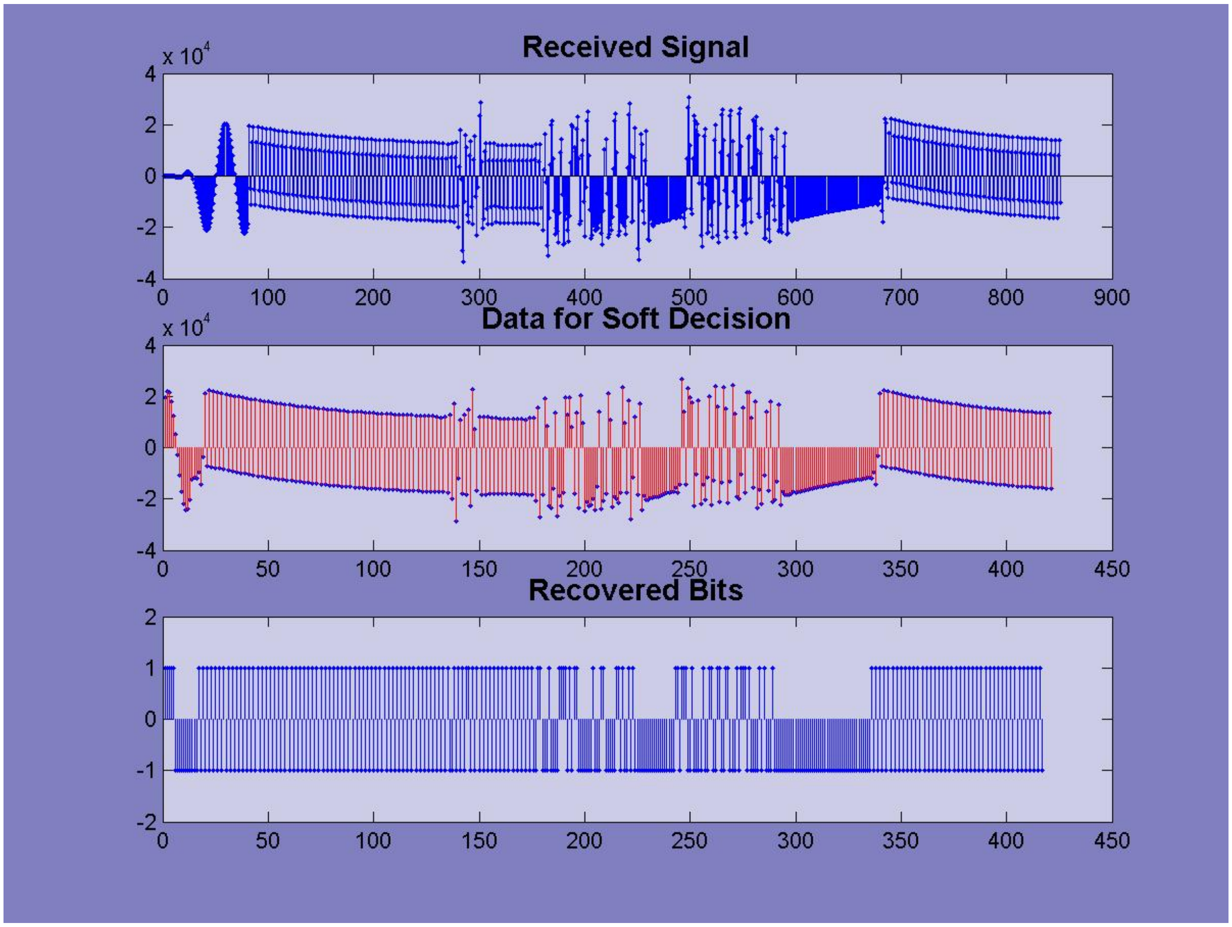


Figure 8.13 : Symbol Timing Recovery Loop

8.8 Problems

There were a lot many which we faced and they were overcome by the guidance and help provided by our teachers. But still there were few problems which we could not overcome.

8.8.1 No Guidance.

When we started out working on DSP boards no body in the college had any idea how to use them. So we had to work ourselves out from the dregs to the top. There is no person qualified in Digital Signal Processor nor enough books are available in the college. So it took us two months before we were able to take in the audio and play it out without any processing and after that the real work started.

8.8.2 Sampling Rate of available Boards

The prototype designed by us aims to achieve all the functionality of the radio in digital domain and to execute all the software in the real time. We could not achieve the latter goal, due to non availability of daughter card for the digital signal processor used by us. The daughter card more specifically PCM 3003 of Texas Instruments could provide the desired speed and memory to achieve real time execution and implementation of the design. Our design is complete and if this card or preferably one with more speed and memory could be provided we would still be willing to perform the real time audio transmission.

8.9 Recommendations

Software radio can bring benefits to any organization involved in the telecommunications. So there is a need for R & D efforts on a large scale to reduce the hardware platform. We have worked in base band domain and there is still a lot of roam for development.

8.9.1 Optimization of C Code / Assembly Level Routines.

Code Composer Studio gives an option to mix the assembly and C code, so the next step is to write a mixed code of C and Assembly and then finally convert whole project in the assembly language. Although our design is optimized to function in real time but maximum optimization can be achieved by this process.

8.8.2 Passband Modulator.

Our main emphasis in this project was on baseband modulation / demodulatioon. This project can be taken ahead by performing implementation (modulation / demodulation) in passband too.

8.8.3 Equalizer.

Channel Impairments can be dealt with performing Equalization.

8.8.4 Purchase of AED 101 Cards.

At the moment the DSP Kits available with the college has the sampling rate of 8 KHz which is not even sufficient for the audio processing. AED 101 cards provide the sampling rates upto 80 MHz and they give two channels. So with the help of these cards even sampling at IF and RF is possible. If these cards are made available by the college it can really boost the R & D of Software Radio in College.

References

[1] Duško Zgonjanin, Kiril Mitrevski and Ljubomir Zelenbaba’s , “Software Radio: Principles and Overview”., Faculty of Electrical Engineering, Belgrade

[2] Shinichiro Haruyama, “Software Defined Radio Technologies”, Advanced Telecommunication Laboratory, SONY Computer Science Laboratories, Inc.

[3] Robert H. Walden, “Analog-to-Digital Converter Survey and Analysis”, 16th IEEE Instrumentation and Measurement Technology Conference, pp.1558-1562, 1999.

[4] R. J. W. T. Tangelder, H. de Vries, R. Rosing, H. G. Kerkhoff, and M. Sachdev, “Jitter and Decision-level Noise Separation in A/D Converters”, IEEE Communications Magazine, pp.90-95, February 1999.

[5] Steven W. Smith, “The Scientist and Engineer's Guide to Digital Signal Processing”

[6] Dr. Yun Q. Shi , Lectures of Dept of Electrical & Computer Engineering New Jersey Institute of Technology

[7] http://www.data-compression.com/speech.html

[8] http://www.bell-labs.com

[9] http://cslu.cse.ogi.edu/HLTsurvey/ch10node4.html

[10] M. H. Johnson and A. Alwan, “Speech Coding: Fundamentals and Applications”, to appear as a chapter in the Encyclopedia of Telecommunications, Wiley, December 2002.

[11] http://www.ti.com/corp/docs/company/history/pmos.shtml

[12] http://www-mobile.ecs.soton.ac.uk

[13] EIA/TIA-PN2398 (IS-54), “The 8 kbits/s VSELP algorithm”, 1989.

[14] I. Gerson and M. Jasiuk, “Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 8 kbits/s”, Proc. ICASSP-90, pp. 461-464, New Mexico, Apr. 1990.

[15] M.R. Schroeder and B. Atal, “Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates”, Proc. ICASSP-85, p. 937, Tampa, Apr. 1985

[16]. A. Oppenheim and R. Schafer, Discrete Time Signal Processing (Englewood Cliffs, NJ: Prentice- Hall, 1989): 80-87.

[17]. K. Pohlman, Principles of Digital Audio (Indianapolis, IN: Howard W. Sams and Co., 1989).

[18]. CCITT Recommendation G.711: Pulse Code Modulation (PCM) of Voice frequencies (Geneva: International Telecommunications Union, 1972).

[19] Kevin McClaning, “an Intuitive Look at Matched Filtering”

[20] Lee and Messerchaesit the noise book

[21] Louis Litwin, “Matched filtering and timing recovery in digital receivers practical look at methods for signal detection and symbol synchronization”

[22] Digital Filter Design in G:\Didital Communication\Pulse Shaping

[23] Nader Sheikholeslami Peter Kabal, “A Family of Nyquist Filters Based on Generalized Raised-Cosine Spectra”

[24] Lee and Messerchaesit the Noise Book

[25] www.complextoreal.com

[26] Bernard Sklar, “Digital Communications, Fundamentals and Applications”, Prentice Hall, 2nd Edition, 2001.

[27] Technical Brief C6000, Code Composer Studio Version 2.0 Documentation