DSP Based Data Acquisition System for Nuclear Physics

A Thesis Submitted to the Faculty of Engineering & Technology, Jadavpur University in Partial Fulfillment of the Requirements for the Degree of Master of Nuclear Engineering

Under the guidance of

DR. SANDEEP. S. GHUGRE

Submitted by

ANJANI KUMAR TIWARI

EXAMINATION ROLL NUMBER: M4NUC12-02

REGISTRATION NUMBER: 113876 OF 2010-2011

SCHOOL OF NUCLEAR STUDIES AND APPLICATIONS

JADAVPUR UNIVERSITY, KOLKATA-700032

INDIA

MAY-2012

Jadavpur University

Faculty of Engineering & Technology

The School of Nuclear Studies and Applications

Certificate of Recommendation

*We hereby recommend the thesis, entitled "**DSP Based Data Acquisition System for Nuclear Physics**" prepared and supervised under the guidance of **Dr. Sandeep. S. Ghugre**, by **Mr. Anjani Kumar Tiwari**, be accepted in partial fulfillment of the requirements of the Degree of Master of Nuclear Engineering in The School of Nuclear Studies and Applications of Jadavpur University, Kolkata.*

____

Dr. Sandeep. S. Ghugre

(THESIS ADVISOR)

Countersigned:

____

Prof. Dr. Amitava Gupta

Director

The School of Nuclear Studies & Applications

Jadavpur University

Kolkata.

Faculty of Engineering & Technology

Jadavpur University

Kolkata-700032

Certificate of Approval

The foregoing thesis is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a pre-requisite to the degree for which it has been submitted. It is understood that by this approval expressed or conclusion drawn therein but the thesis only for the purpose for which it has been submitted.

______________________ ___________________________

Signature of Examiner *Signature of Supervisors*

Declaration of Originality and Compliance of Academic Ethics

*I hereby declare that this thesis entitled **"DSP Based Data Acquisition System for Nuclear Physics"** contains literature survey and original research work by the undersigned candidate, as part of my Degree of Master of Nuclear Engineering.*

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name (Block Letters)	: ***ANJANI KUMAR TIWARI***
Roll Number	: ***M4NUC12-02***
Registration number	: ***113876 of 2010-11***
Signature with Date	:

Acknowledgement

I would like to take the opportunity to thank. Dr. Sandeep. S. Ghugre under whose supervision and constant guidance my work has been done. For the past one year he has always provided me with enthusiasm, knowledge and vision. During the times of despairs, he would always stand by my side, providing me support. He is an excellent researcher and the perfect guide, a M.E. student can wish for.

I had the privilege to work in the most sophisticated and advanced laboratory, with excellent computer facility, of the Jadavpur University, for which I must thank The School of Nuclear Studies and Applications Director, Prof. Dr. Amitava Gupta. The office staffs also extended their helping hands and I thank them all for their help.

Last but not the least I am thankful to Dr. Rajarshi Raut, whose active involvement and inspiration helped me to overcome all difficulties. I also like to thank my all classmates and senior scholars for constant support and guidance that inspired me to go for these studies.

Finally, I am indebted to my family, without whose support and encouragement I would never have reached the position, I am today.

(Anjani Kumar Tiwari)

Contents

Abstract

Pulse processing techniques based on digital technology is an emerging trend in the domain of contemporary nuclear physics experiments. This development stems out from the advent of large and multi-detector arrays in nuclear spectroscopy, leading to the need for compact instrumentation as well as data acquisition systems capable of handling the huge data rate. The present thesis has probed the various aspects of digital signal processing (DSP) as applied to the domain of nuclear spectroscopy. The different facets of DSP techniques have been reviewed and the conventional instrumentation for the nuclear spectroscopy experiments has been discussed. The advantages and demerits of the two approaches have been considered. The different types of digital pulse processing techniques, realized through recursive algorithms, have been studied. The aforesaid algorithms have been applied to pulses, ideal as well as realistic, simulated through appropriate analytical expressions. The simulation of the pulses and application of filters has been carried out in MATLAB®. Effect of choice of parameters of the recursive algorithms have been probed at particular instances.

List of Tables

List of Figures

Chapter 1

Digital Signal Processing in Nuclear Spectroscopy

Digital signal processing (DSP) is one of the most powerful technology of 21^{st} century. It is the technology that has developed rapidly over the past four or five decades. This rapid development is the result of advancement in computer technology and electronics industry. With development of VLSI technology in electronics it is possible to design smaller, more powerful, faster and cheaper computers and digital hardware that are capable of faster and reliable digital signal processing tasks and some really complex and tedious calculations which are difficult and expensive if performed by analog signal processing technique and analog hardware system. Thus the current trend is that the task is performed by DSP and its associated hardware system which was earlier realized only by analog circuitry. DSP has wide range of application which includes communication, medical imaging, radars and sonar, high quality music system, image processing to name a few. All of these have their own specialized algorithms. Digital signal processing is concerned with the representation of discrete time signals by a sequence of numbers or symbols and processing of these signals. DSP is very useful technique applied in nuclear physics and engineering for radiation detection and measurement. In this thesis we shall particularly focus on the application of various DSP techniques used in nuclear energy spectroscopy.

1.1 Signal Processing.

A signal is defined as any physical quantity which is a function of a variable or variables like time, space or any other physical quantity. For example

$$y = \sin 2\pi x$$

where y is a function of x, x being an independent variable.

Similarly we can represent a signal with two variables like,

$$z(x, y) = 2x^2 + 5y^3$$

where z is a signal and is a function of two variables x and y.

Signal processing deals with the change in parameters of signal according to our predetermined purpose. The parameters that can be changed are amplitude, phase, frequency etc. The signal whose parameter is to be changed is often referred as input signal and the system that changes the parameter of the input signal is referred as signal processing system and the signal that generally comes out after passing through the signal processing system is known as the output signal. Some of the signal processing tasks is amplification, attenuation, filtering, modulation etc.

Signal processing can be done by two methods, as we shall discuss in the subsequent sections.

1.1.1 Analog Signal Processing

A signal is said to be analog signal if it is continuous in nature. The signal must have time varying flow of information. Typical examples of analog signals are voltage signal, current signal etc. Mathematical realization and representation of this analog system is said to be analog signal processing. Examples of analog processing system include filtering, volume control of stereo etc. Most common analog processing elements include resistors, inductors, capacitors transistor, frequency analyzers etc. These elements change the characteristics of the signal or extract some useful information from the signal then we may say that the signal is processed. The block diagram of analog signal processing is shown below.

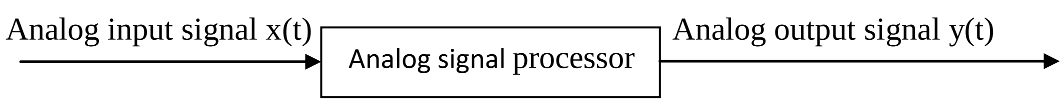


Fig-1.1. Analog processing system

In above figure analog signal processor is the system which makes the desired change to the input signal. In analog signal processing we need various signal processing mathematical tools. Some of the basic tools are discussed below

1.1.1.1 Convolution

Convolution tools in analog signal processing are very basic concept. This is explained as any input signal can be combined with the function of the system to give output. In order to calculate the convolution one signal is reversed and shifted to become identical to the second signal, then we multiply the two signal and perform an integration. Mathematical representation of convolution is shown below.

$$y(t) = (x * \partial)(t) \int_{a}^{b} x(\tau)\partial(t-\tau)d\tau$$

Typical value is a=-∞ and b=∞

Suppose we have two signals x(t) and ∂(t). By calculating convolution we can find how much the reversed ∂(t) function needs to be shifted along particular axis to become identical to the signal x(t). The convolution function reverses and shifts function ∂(t) and calculates the integral of the product of the signal x(t) and reversed signal ∂(t) for each possible shifting. When the two function matches, the value of (x(t)$*\partial$(t)) is maximized because it contributes to the integral when positive area (picks) and negative areas (troughs) are multiplied.

1.1.1.2 Fourier Transform

Fourier transform transforms a complex analog signal into successive individual components over a period of time or frequency. In general we can say that Fourier transform is a mathematical operation that transform time domain signals (signal which is a function of time) into frequency domain (signal is function of frequency). If frequency domain signal is transformed back to time domain signal we use inverse Fourier transform. Necessary condition for a signal to perform Fourier transform is,

$$\int_{-\infty}^{\infty} |x(t)| dt < \infty$$

Fourier transform can be written as,

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

Here the signal is transformed from time domain to frequency domain

Inverse Fourier transform can be written as,

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega) e^{jwt} d\omega$$

1.1.1.3. Laplace Transform

Laplace transform is a very useful transform in analog signal processing. Actually it is the generalized form of above discussed Fourier transform. It allows transform for almost any signal or system. One of the applications of Laplace transform is that it is very useful in solving ordinary differential equations arising in the mathematical modeling of electrical or electronics circuits where it transforms complex differential equation into algebraic equation which is much easier to solve.

Laplace transform is represented as.

$$X(s) = \int_{0}^{\infty} x(t) e^{-st} dt$$

$$s = \sigma + j\omega$$

where σ & ω are having real values and s is a complex

Inverse Laplace transform can be written as,

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(s) e^{st} ds$$

1.1.1.4. Bode Plot

Bode plot gives the frequency response of the system. Its the plot between magnitude versus frequency. The phase between frequency of a system magnitude is measured in DB(decibel) while phase is measured either in degree or radians.

1.1.2 Digital Signal Processing

Digital signal processing is concerned with the processing of discrete signals. The signals that are defined at discrete instants of time, frequency or any other parameter are called as discrete signals. Discrete time signal is shown below.

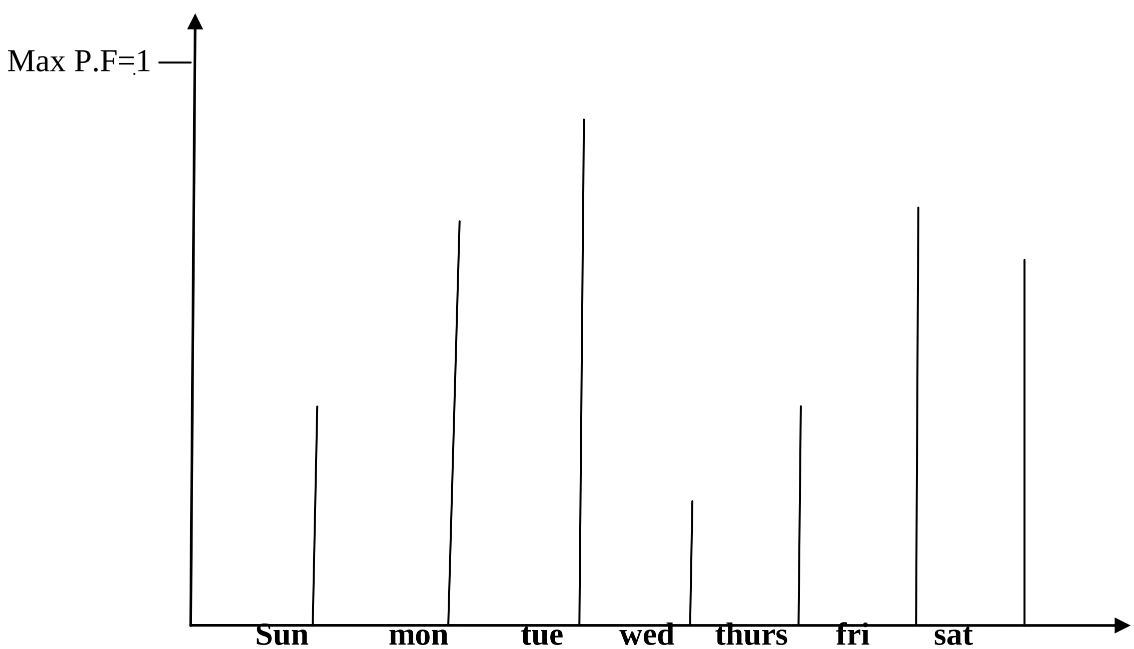


Figure-1.2 Discrete Signal

In the above figure average power factor of a household is shown over the week. Here we see that it's a discrete signal because we are taking the power factor at discrete instants that is over a day. The main task of digital signal processing (DSP) are detection of the signal, recognition and classification of the signal concerned, estimation of the signal concerned, transformation of the signal from one parameter to another parameter, filtering or coding (compression of the signal), synthesis and reproduction of the signal, recording and archiving of the signal and the most important task to analyze and model the concerned signal according to our necessity. A typical block diagram of digital signal processing system is shown below

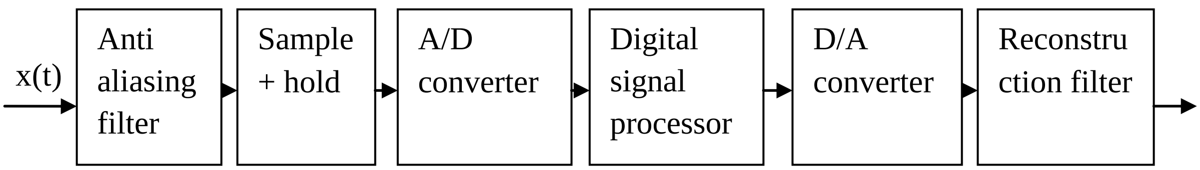


Figure-1.3. Digital signal processing system [1].

In the above figure x(t) is the input signal and the source of the input signal may be from transducer or signal from a radiation detector. The input signal is then passed through the antialiasing filter. This is a low pass filter used to remove the high frequency noise and to band limit the signal. In addition to this we may use 50HZ notch filter (band-stop filter) to remove the power frequency components which is large noise source. Sample and hold circuit provide input to the analog-to-digital converter circuit. Its main task is to sample and hold (freeze, lock) the amplitude of the input signal for a specific period of time because A/D should not be provided with changing signal. The changing signal should be sampled and this sampled signal must be hold on while conversion is taking place. Generally sampled is hold until next sample is taken. Detailed discussion about sampling theorem and its consequences will be discussed later. The output of the sample and hold circuit is fed into A/D (analog-to-digital) which converts the analog signal into digital one. Brief discussion of A/D converter will be taken up later. The output of A/D converter is then fed into the digital signal processor which may be a large programmable digital computer or a microprocessor ex- Intel's 80** series etc. which is programmed to perform desired operation on the input signal; it may be a digital signal processing hardware ex- TMS320c50. Brief discussion about digital signal processor will be done later. The output of digital signal processor is then fed into D/A (digital to analog) converter. This converts already processed digital signal back to the analog signal. Common D/A converters are weighted resistor and the R-2R ladder converters. The output from the D/A converters is not smooth. So for reconstruction of signal output from the D/A converter is fed into reconstruction filter. This filter is usually a low pass filter and it is effective in suppression of the harmonic content of the signal.

1.2. Sampling Theorem

We know in digital signal processing system, the information is represented in digital format. The signal which is to be processed is converted to its equivalent numerical format prior to its processing this conversion is known as sampling. The information contained in an analog signal is first converted to digital samples which are equally spaced in time. The figure below shows an analog signal and its sampled version.

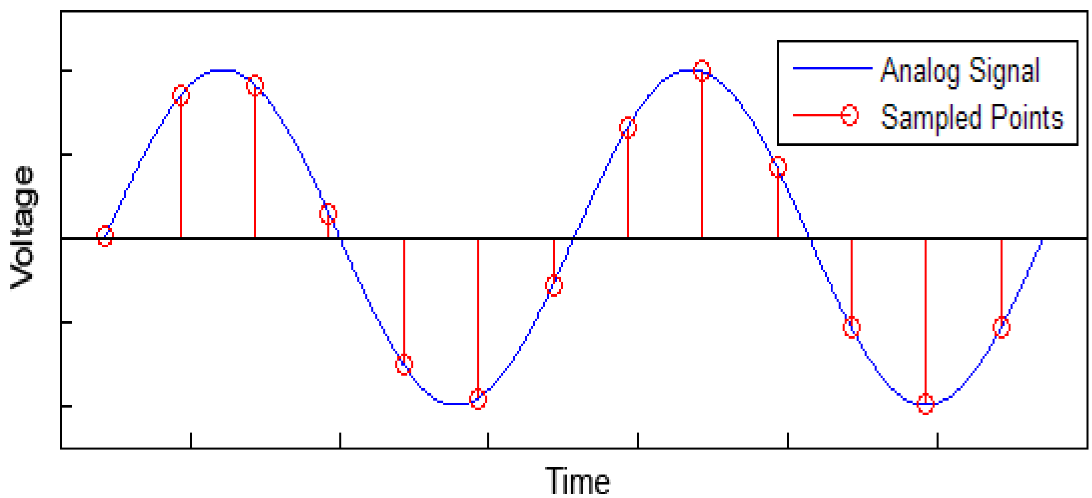


Figure-1.4. Analog signal and its sampled version.

The frequency at which signal is sampled is known as **sampling rate.**

According to sampling theorem, an analog signal can be exactly reconstructed from its samples if the sampling rate is at least twice the highest frequency component present in the signal. If the signal contains the highest frequency component of 1 KHz, the sampling rate must be at least 2 Kilo Samples/Second. This sampling rate is also known as Nyquist rate. Violation of sampling theorem, *viz*. sampling the signal at a sample rate less than nyquist rate, leads to unacceptable distortion in the signal. This distortion is known as aliasing. Due to aliasing, the components present in the input signal whose frequency is higher than the nyquist frequency will be sampled as a signal of lower frequency. And this aliased samples in turn corrupts the original signal which is lower than nyquist frequency due to the presence of aliasing, a signal that contains frequency components higher than nyquist frequency cannot be reconstructed from its samples. The effect of aliasing is shown in the figure below.

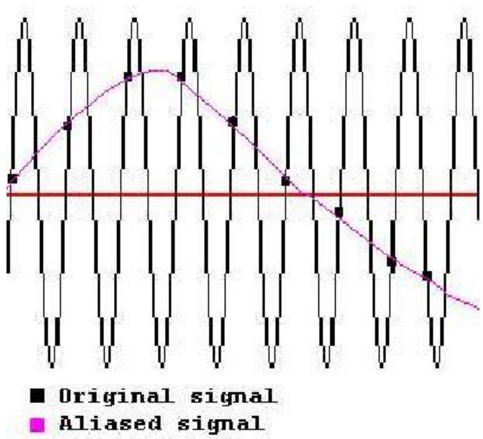


Figure-1.5. Original signal and aliased signal.

The black colour is the original signal while the signal in violet colour is aliased signal from the original signal. We can see that the aliased signal will be present in sampled data as lower frequency signal and this will effect the original signal. It should be noted that the information in between two consecutive samples is lost forever. But still the entire signal can be reconstructed from the samples as long as sampling theorem is satisfied.

Thus aliasing refers to one signal "pretending" to be another signal when their samples are the same. It is the undesired effect due to under sampling whereby one signal can "masquerade" as another.

1.3. A/D Converter

The analog to digital converter must take into account many factors, such as input signal, sampling rate, throughput, resolution, range, and gain. An A/D converter takes a continuous analog input signal, and converts it to a digital signal. A digital signal is either ON (Logic 1) or OFF (Logic 0), like a light switch. One of the factors to be considered in converting analog signals to digital is the sampling rate. The sampling rate determines how often conversions take place. The higher the sampling rate, the better is the conversion. A problem with using too low of a sampling rate is that aliasing might occur. Aliasing is when the acquired signal gets distorted by a sampling rate that is too small (as discussed in the subsequent section). In this example, the original signal is sampled so slowly that the sampled signal looks like a completely different frequency than the original signal. Another factor to consider with A/D conversion is throughput. The effective rate of each individual channel is inversely proportional to the number of channels sampled. For example, if the total maximum throughput of a system is 100 kilohertz and 16 channels of data must be used, then the total throughput for each channel is 6.25 kilohertz. The range of an A/D converter is the minimum and maximum voltage levels that the A/D converter can quantize. Ranges are selectable either through converter hardware or software to accurately measure the signal.

1.3.1. A/D Converter Resolution

The resolution of a converted signal is the number of bits that are used to store each sample of data. For example, a two bit resolution will allow 2^2 number of values for the data, meaning that the data can take 4 possible values. For 3 bit conversion, 8 values are possible, and for 5 bit conversion, 32 values are possible. The higher the resolution, the closer the digitized signal will resemble the original analog waveform. It is clearly illustrated in the picture below.

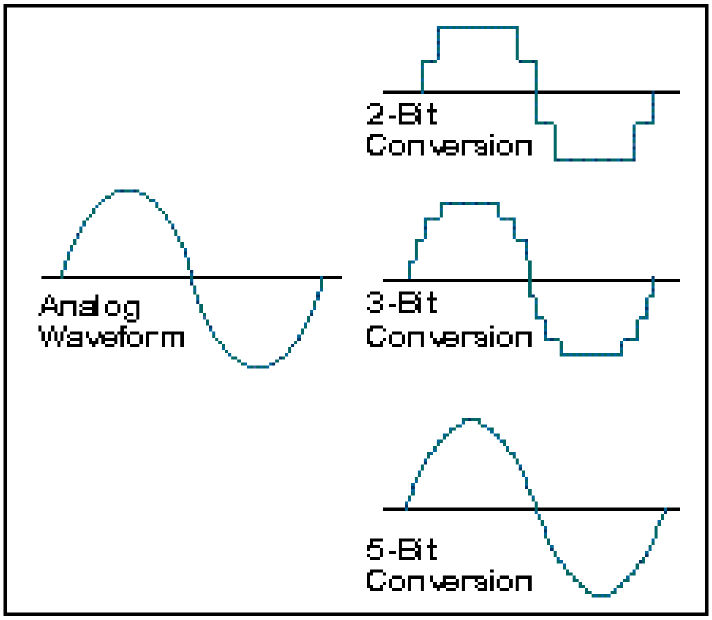


Figure-1.6. Analog to digitally converted waveform [3]

1.4. Code Width

Code width is a smallest signal that a DAQ system can detect.

$$\text{Code width} = D\frac{1}{2^N}$$

where D = DAQ device input range and N = resolution

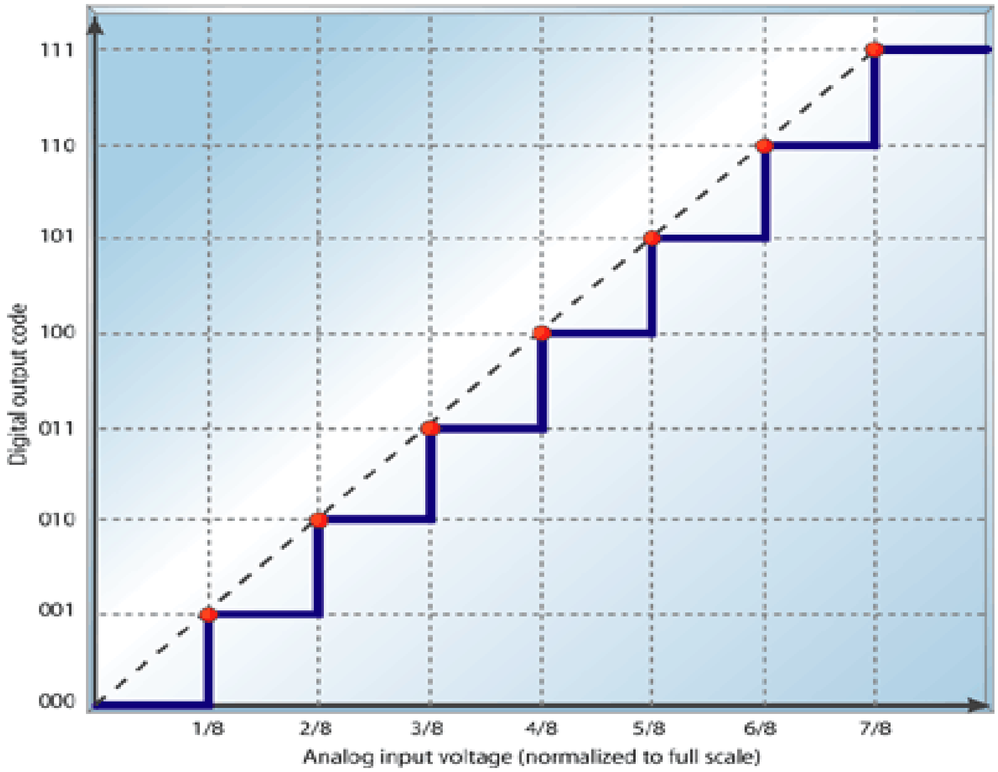


Figure-1.7 Analog input voltage normalized to full scale

The above figure depicts an ideal transfer function for a 3-bit ADC with reference points at code transition boundaries. The output code will be its lowest (000) at less than 1/8 of the full-scale (the size of this ADC's code width). Also, note that the ADC reaches its full-scale output code (111) at 7/8 of full scale, not at the full-scale value. Thus, the transition to the maximum digital output does not occur at full-scale input voltage. The transition occurs at one code width—or least significant bit (LSB)—less than full-scale input voltage (in other words, voltage reference voltage).

Larger resolution implies lower code width and more accurate representation of signal. For example consider a 8 bit DAQ device with a 0 to 10 v input range detects a 39mv change, while the same device with range -10 to 10 volt detect a change of 78mv

$$C = 10.\frac{1}{2^8} = 39mv.$$

$$C = 20.\frac{1}{2^8} = 78mv.$$

1.5. Quantization.

In digital signal processing quantisation refers to mapping of large set of input values into smaller set. It means approximating a continuous or very large range of discrete values into relatively smaller set of discrete values or integers. A common use of quantisation is in conversion of discrete signal into digital signal by quantizing and explained below.

If we multiply the original signal with train of pulses of unit amplitude we visualize it as a sampling. This leaves the original signal with information at discrete points with magnitude of signal present in the original signal at that position. This is illustrated in the figure below

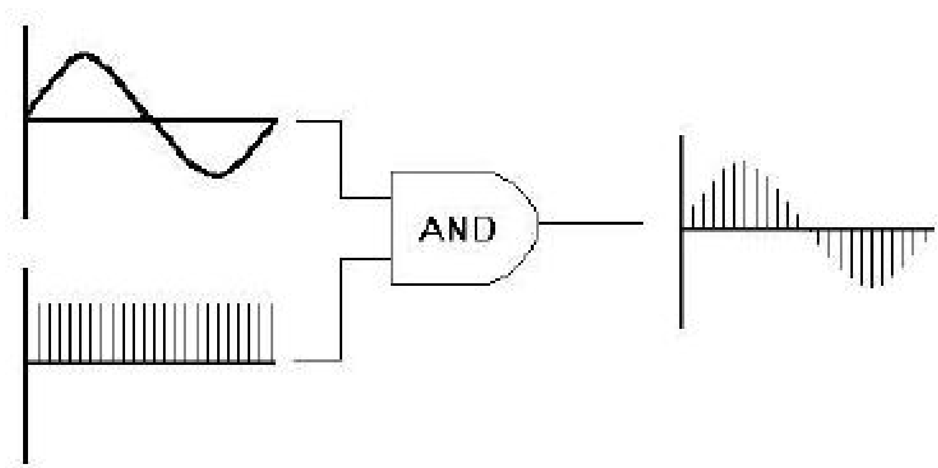


Figure-1.8 Discretisation of analog signal

Now samples of the input signal are taken at discrete points. But to analyze the signal with a microprocessor or micro controller, these pulses has to be converted in to numbers. Before this conversion, the pulses are quantized to some finite number of quantization levels. For example, if the quantization levels are 0,1,2,3, etc... , a pulse with magnitude between 1 and 2 will be quantized to either 1 or 2. This in fact introduces a noise in the sampled signal. Such noise in known as quantization noise

1.6. Filters

Filter is said to be a physical device that can extract some useful information and suppress unwanted information from a signal. Some purpose of use of filter is shown below.

(1) Separation of signal that have been combined.
(2) Restoration of signal that have been distorted.

Analog (electronic) filters can be used for these same tasks. However, digital filters can achieve far superior results. Hence, digital filters are usually used to discriminate a frequency or a band of frequency from a given signal which is normally a mixture of both desired and undesired signals. The undesired portion of the signal commonly comes from noise sources such as power line hum etc. or other signals which are not required for the current application. The key difference between analog and digital filters is that instead of calculating capacitor, resistor and inductor values for an analog filter, coefficient values are calculated for digital filters. Thus for digital filter numbers replace the physical resistor, capacitor and inductor components of the analog filter. Analog filters are cheap, fast, and have a large dynamic range in both amplitude and frequency. Digital filters, in comparison, are vastly superior in the level of performance that can be achieved.

The purpose of these filters is to allow some frequencies to pass unaltered, while completely blocking other frequencies. The passband refers to those frequencies that are passed, while the stop-band contains those frequencies that are blocked. The transition band is between. A fast roll-off means that the transition band is very narrow. The division between the pass-band and transition band is called the cutoff frequency**.** In analog filter design, the cutoff frequency is usually defined to be where the amplitude is reduced to 0.707. Some of the commonly used filters are given below

1) Low-pass filter (passes low frequency component through it and high frequency components are attenuated).

2) High-pass filter (passes high frequency component through it and low frequency component is attenuated while passing through it).
3) Band-pass filter (only frequency components within certain range of frequency is passed while all other frequency components are attenuated).
4) Band-reject filters (only frequency component within certain band is attenuated while all other frequency components are passed).

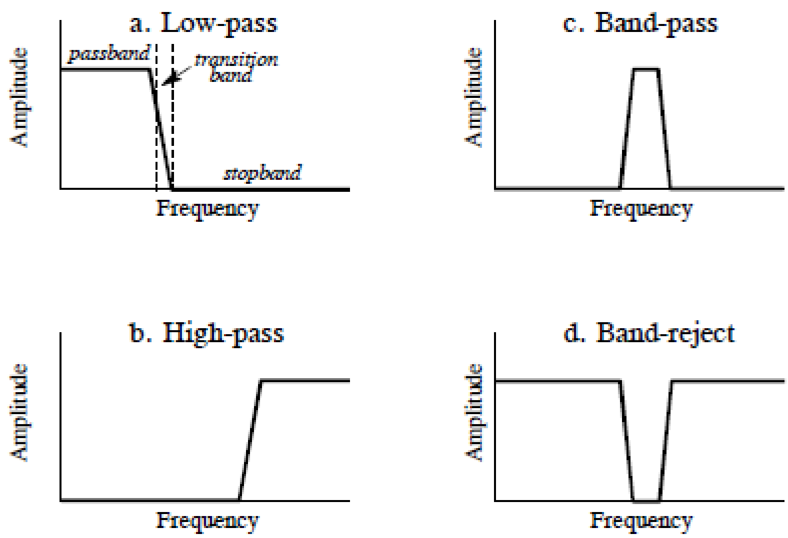


Figure-1.9 Different types of filter

1.7. Digital Filter

Digital filtering is one of the most powerful tools in DSP. Its main task is to transform a given sequence into another sequence by numerical method or certain concerned algorithm so that transformed sequence has more meaningful properties. The major advantage is the virtual elimination of errors in the filters associated with passive component, fluctuation over time and temperature, op-amp drift (active filters) and other effects. In addition the characteristics of digital filters can be easily changed through software. Block diagram of digital filtering action is shown below.

Figure-1.10. Digital filtering action

Here in the above figure x(n) is the input sequence which after filtering action produces output sequence y(n).

There are two basic types of digital filters,

1. Infinite Impulse Response (IIR)
2. Finite Impulse Response(FIR)

The general form of the digital filter difference is

$$y(n) = \sum_{i=0}^{n} a_i x(n-i) - \sum_{i=1}^{n} b_i y(n-i)$$

where $y(n)$ is the current filter output, the $y(n-i)$'s are previous filter outputs, the $x(n-i)$'s are current filter input if value of i is zero or previous filter's inputs if the value of i is more than zero, the a_i are the filter's feed forward coefficients, corresponding to the zeros of the filter, the b_i are the filter's feedback coefficients corresponding to the poles of the filter, and n is the filter's order . IIR filters have one or more nonzero feedback coefficients. That is, as a result of the feedback term, if the filter has one or more poles, once the filter has been excited with an impulse there is always an output. Examples of IIR filter is recursive filter.

Block diagram of a second order IIR filter is shown below.

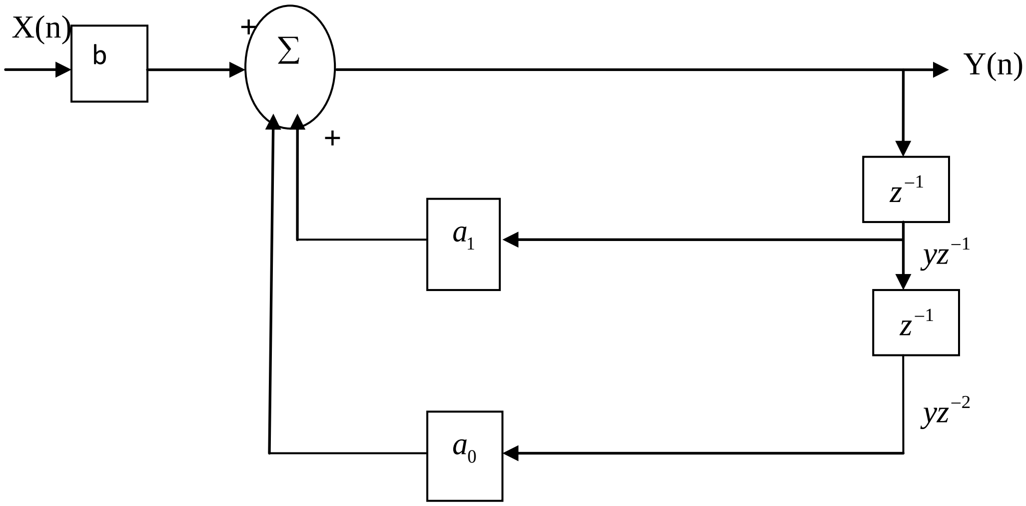


Figure-1.10. Block diagram of 2nd order IIR filter.

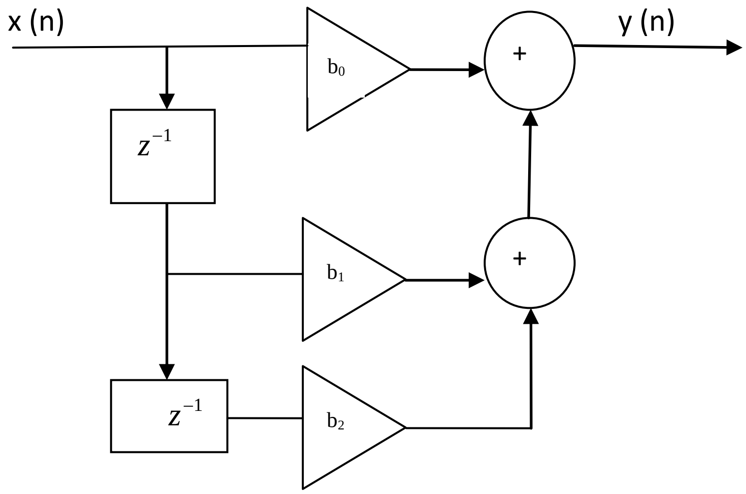
Mathematical equation describing above filter is

$$\frac{y(n)}{x(n)} = \frac{bz^2}{z^2 - a_1 z - a_0}$$

Here in the above figure z^{-1} means delay by one unit.

FIR filters have no non-zero feedback coefficient. That is, the filter has only zeros, and once it has been excited with an impulse, the output is present for only a finite (n) number of computational cycles. For an FIR filter, the response due to an impulse input will decay within a finite time. But for IIR filter, the impulse response never dies out, it theoretically extends to infinity. Commonly FIR filters are known as non-recursive filters and IIR filters are known as recursive filters. These names are derived from the nature of algorithms used for these filters. In comparison, implementation of FIR filters is easy, but it is slower when compared to IIR filters. Though IIR filters are fast, practical implementation is a bit complex compared to FIR filters.

Block diagram of a FIR filter is shown below



Figure–1.11 Block diagram of a FIR filter

Mathematical equation that describes above diagram can be written as

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2)$$

FIR filters practical application includes that it is used as anti-aliasing filter, low pass filter etc.

Examples of digital filters are moving average filter windowed-sinc filters etc.

1.8. Moving Average Filter.

This very common type of filter used in signal processing. This filter gives output as a average of present input and some specific number of previous input samples. Let’s consider a example where filter averages the current input and last 4 input samples. Let the sampled input values are shown in table 1 below. Here T is the sampling period.

Table 1.1 Time And Input Sample for moving average filter

Time	Input sample
0	1
T	6
2T	4
3T	8
4T	12
5T	5
6T	7
7T	9
8T	13
9T	10

As we have to average the current sample and the past four input samples. The processor will need four registers to store previous four input values. The content of the processor will update every time a new sample is taken. Let for simplicity the registers are initially reset means they have zero initial value. The following sequence shows how the first four samples ‘1’,’6’,’4’ and ‘8’ are processed.

Time=0; input sample=1

Current sample	register 1	register 2	register 3	register 4
1	0	0	0	0

Output value= (1+0+0+0+0)/5 =0.2

Similarly for time=T; input sample=6

Current sample	register 1	register 2	register 3	register 4
6	1	0	0	0

The previous input sample ‘1’ is now shifted to be stored in register 1

Output value= (6+1+0+0+0)/5=1.4

Similarly for time =2T; input sample=4

Current sample	register 1	register 2	register 3	register 4
4	6	1	0	0

Output value= (4+6+1+0+0)/5=2.2

Similarly for time =3T; input sample=8

Current sample	register1	register 2	register 3	register
8	4	6	1	0

Output value=(8+4+6+1)/5=3.8

And so on we can get the output values

Table 1.2 shows the output values after the averaging.

Time	Input sample	Output sample
0	1	0.2
T	6	1.4
2T	4	2.2
3T	8	3.8
4T	12	6.2
5T	5	7.0
6T	7	7.2
7T	9	8.2
8T	13	9.2
9T	10	8.8

The first four output values of ‘0.2’,’1.4’,’2.2’ and ‘3.8’ represents the initial transient i.e. the part of the output signal where the initial four zero are being shifted out of four storage register. The output values are valid only if these initial values are cleared out of the storage register.

A moving average filter is realized as low pass filter because it tends to smooth out any rapid changes in a signal.

1.9. Windowed Sinc Filter

It is considered as an ideal low pass filter completely allowing low frequency components to pass through it and stopping high frequency component. These filters are very stable and excellent frequency domain characteristics can be found but at the cost of poor performance on time domain with lots of ripple in its step response and overshoot.

1.10. Digital Signal Processor

Digital signal processor is a specialized microprocessor with a sophisticated design and architecture needed to perform various tasks of digital signal processing. Digital signal processing algorithms require large number of mathematical operation to be performed quickly and repetitively over a set of data. Signals are constantly changed from analog to digital and manipulated digitally and converted back to analog form this requires very fast processing and this is fulfilled by digital signal processors. Most of the digital signal processors have constrain upon latency which means that a certain task must be completed before some specific period. Typical use of digital signal processor is to following real time task data manipulation and mathematical calculations, filtering and estimation

1.11. Architecture of Digital Signal Processor

Architecture of digital signal processor represents the procedure by which the elements of digital signal processor are interconnected and exchanges data between themselves. Every type of digital signal processors works in a unique architecture designed for it and cannot work in another architecture. Every type of architecture requires special hardware features from its c.p.u and bus systems.

Each type of architecture includes.

1) At least one central processing unit for processing of require data

2) At least one peripheral for I/O

3) At least one bus system connected between central processing unit and its peripheral for data transfer.

1.11.1. Von Neumann Architecture.

This type of architecture posess single bus system. The same bus carries all the information to be exchanged between central processing unit and it's peripheral including the instruction codes and data processed by the central processing unit. The single bus system of the three buses is shown below

1) The data bus - this bus conveys data between central processing unit and its peripherals. It is bi-directional.the central processing unit can read as well as write data on its peripherals.

2) The address bus - the address bus is uni-directional. The central processing unit access the address bus to indicate which peripherals it wants to access and within peripherals which specific memory locations it wants to access. The central processing unit always writes the data which is read by its peripherals.

3) The control bus - this buses carries the instructions that are used to manage and synchronize the exchanges between the central processing unit and its peripherals. For example the signal that is used to instruct central processing unit to read and write or synchronize the exchange is a part of control bus. The main signals of the buses are originated at central processing unit and sent to the peripheral however special lines of the control bus, such as interrupt or wait state signals carry information from peripherals to the central processing unit.

Von Neumann architecture performs satisfactorily serial tasks we only need some other architecture if we require fast processing but at the same time complexity will increase [3,4].

The block diagram of Von Neumann architecture is shown below.

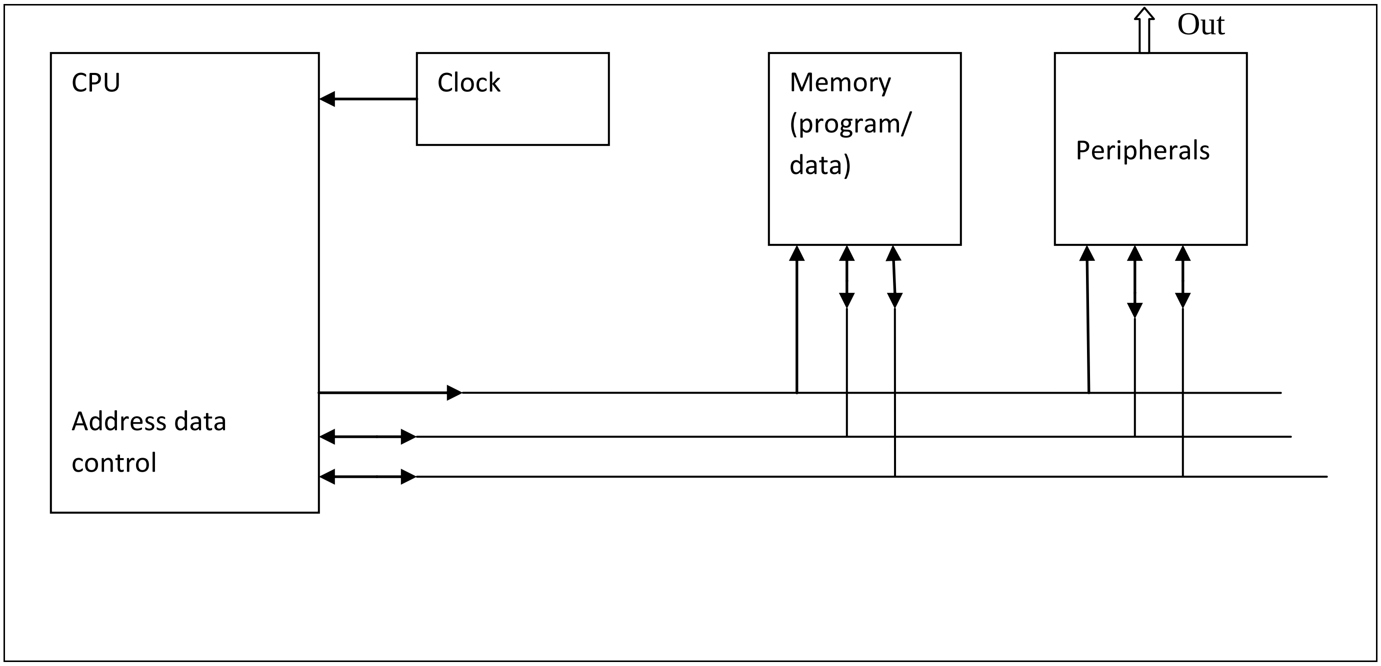


Figure-1.12. Block diagram of Von Neumann architecture [4].

1.11.2. Harvard Architecture.

As the name tells, Harvard Architecture was developed at the Harvard University. Harvard architecture uses two separate bus system to transport the instruction codes and the data being processed by the central processing unit. The buses are shown below.

1) The program bus system - is used to transport instruction codes from the program memory to central processing unit during the fetch cycle.

2) The data bus system - it is used to transfer data to and fro the central processing unit and from the memory and peripherals.

Since Harvard architecture poses two independent bus systems it can read instruction code and simultaneously read or write on memory or peripherals as part of the execution of previous instruction. It is faster than the Von Neumann architecture. Harvard architecture is comparatively safer than its Von Neumann counterpart because central processing unit cannot mistakenly write codes into the program memory and therefore corrupt the code while it is executing. However the Harvard architecture is less flexible than the Von Neumann architecture because it needs two independent memory units this two resources are not interchangeable. Block diagram of Harvard architecture is shown below.

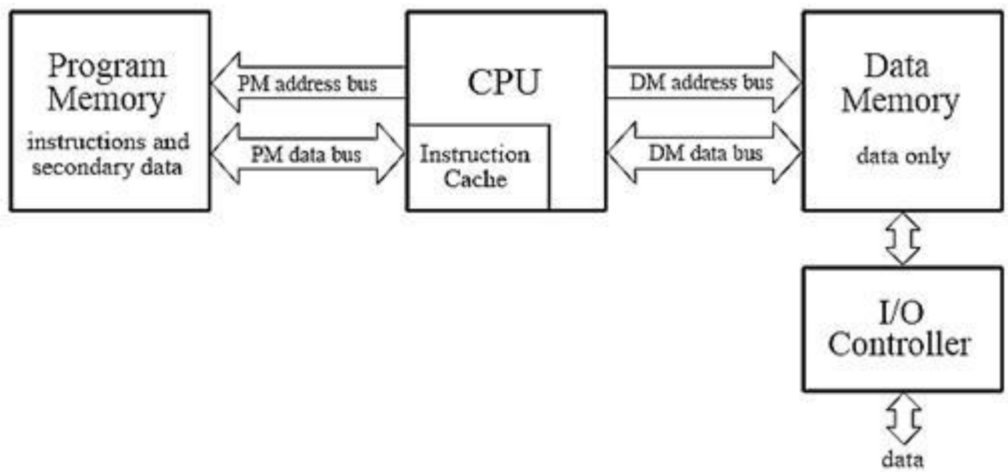


Figure-1.13. Block diagram of Harvard architecture[3].

1.11.3. Super Harvard Architecture.

Super Harvard architecture is the advance version of Harvard architecture by adding some more features to improve the throughput like an instruction cache and an I/O controller. In Harvard architecture out of two buses one bus is used for fetching instructions while second bus is used to fetch operand but this is inadequate for typical DSP operations which usually involve at least two operands. Thus, super Harvard architecture includes cache memory which can be used to store instructions which will be reused leaving both buses free of fetching operands this extension of Harvard architecture plus cache is often termed as extended Harvard architecture or super Harvard architecture (SHARC).

1.12. Comparison between Digital Signal Processing and Analog Signal Processing.

Till now we have discussed about digital signal processing and about its elements now we will discuss pros and cons of DSP and how it is advantageous to use DSP than analog signal processing and its application.

Advantages of digital signal processing over analog signal processing are discussed below [1]

1) **Greater accuracy:** in analog signal processing due to tolerance of the circuit elements accuracy is affected while in DSP accuracy is quite superior.

2) **Cheaper:** in many applications digital realization is quite economic when compared with its analog counterpart.

3) **Easier data storage:** digital signals can be easily stored in magnetic or optical media for further use in remote laboratory without the loss of fidelity.

4) **Implementation of sophisticated algorithm:** we can implement sophisticated and complex algorithm is digital domain which is much faster and easier than analog domain.

5) **Flexibility in configuration:** if you want to change or reconfigure analog system you have to redesign its hardware system which is bit tedious while in digital domain this can be easily done just by changing the program or software.

6) **Applicability of vlf system:** DSP can process very low frequency signals like the signal which is produced after tremor. While in analog domain it is very difficult and requires very large hardware's like capacitor inductor which is neither cost effective nor space saving.

7) **Time sharing:** sharing of processor is possible among no of signals in DSP.

Disadvantages or limitations of digital signal processing over analog signal processing are as follows.

1) **Power consumption:** in analog signal processing algorithm can be implemented with the passive elements like capacitors, resistors and inductor which does not require much power. On the other hand, a DSP chip contain over 0.1 million transistors which in turn consumes substantial power.

2) **System complexity:** in digital signal processing of an analog signal system complexity increases due to use of A/D and D/A converters and its associated filters.

3) **Bandwidth limitation:** we know that in digital signal processing in a certain band signals can be sampled without information loss if the sampling frequency is greater equal to twice the bandwidth of the signals. So we require very fast sampling device to sample a wide bandwidth which is very much difficult task to attain.

1.13. Applications of DSP.

Digital signal processing has wide range of application few of which are mentioned below [1].

1) **Telecommunication:** in simple words we can say that without digital signal processing present day telecommunication industry is hard to imagine. DSP has wide range of application in telecommunication industry like, echo cancellation, telephone dialing application, modems, line repeaters, video conferencing, cell phone, fax etc.

2) **Consumer electronics:** DSP has applications in digital audio/TV, electronic music synthesizer, F.M radio, sound recording applications, mixer, grinder etc. to name a few.

3) **Instrumentation and control:** one of the most important application of DSP is digital filters apart from this spectrum analyzer, phase locked loop (PLL), servo control, robotics, process control, function generator etc.

4) **Image processing:** pattern reorganization, image compression, image enhancement and analysis are the typical example of DSP.

5) **Speech processing:** by application of DSP recognization speech, text to speech and speech to text conversion is possible.

6) **Seismology:** as we discussed earlier that seismological event produces very low frequency vibration which is practically not possible for analog processing system to detect and process but with the use of DSP it is easily done.

7) **Military:** radar signal processing, sonar signal processing, navigation, encryption , secure communication are the few fields of military where there is a wide application of digital signal processing.

8) **Medical industry**: medical diagnostic instruments like computerized tomography(CT) scan, X-ray scanning, magnetic resonance imaging(MRI), ECG(electro cardiogram), EEG(electroencephalography) are the applications of DSP used to detect various disorders in heart and brain of patients.

9) **Nuclear industry**: there are wide ranges of application is evident in nuclear industry which will be discussed later in brief radiation detection and measurement is the typical application of DSP in nuclear industry.

The current thesis will demonstrate the effect of different digital signal processing techniques, as realized through recursive algorithms, on typical detector output pulses used in nuclear spectroscopy.

Chapter 2

Application of DSP in Nuclear Spectroscopy

Spectroscopy generally means measurement of radiation intensity in terms of wavelength. The ultimate goal of nuclear spectroscopy signal processing is to produce such a digital signal that can completely describe the characteristics of radiation and effects produced by the radiation. Nuclear spectroscopy system consists of some electronics circuitry colloquially referred as ‘front end electronics’. The purpose of front end electronics is to acquire electrical pulses generated from radiation detectors. In a typical nuclear spectroscopy experiment, radiation energy is absorbed in the radiation detector. By the interaction of the incident radiation with the detector medium, a burst of charge is produced which when integrated across the summed capacitance presented by the detector, connecting cable and input of the recording circuitry resulting in a voltage pulse. Apart from the energy these pulses may contain information about the type of radiation or particle involved, its time of arrival, its position etc. The quality of signals are degraded and mixed with noise during its passage through non-ideal ‘front end electronics.’ Noise power spectral density can be found if we modeled all the elements in signal processing chain. In signal processing chain each block of chain have some specific function. So we need to interconnect several blocks if we want to extract all the information, as we shall see in the subsequent discussions.

2.1. Elements of Front-End Electronics.

For most of the detectors, the aforesaid charge is so small that it is impractical to process the resulting signal pulses without an intermediate step that improves the signal-to-noise ratio. The subsequent steps of the pulse-processing exercise operate on the signal pulse resulting from the said intermediate step. The key aspects of this process are elaborated below along with the illustration in Fig. 2.1.

1) **Preamplifier:** the first element in the signal processing chain is typically a preamplifier that acts as an interface between the detector and the pulse-processing electronics in the subsequent stage. The preamplifier is usually located in the close proximity of the detector. This is to minimize the capacitive loading of the preamplifier, that would result from long interconnecting cables, and thus improve upon the signal-to-noise (S/N) ratio. One function of the preamplifier is to terminate the capacitance quickly, thus maximizing the S/N ratio. Owing to the factors of safety and convenience, the remaining components of the pulse-processing electronics are located at some distance from the detector and the preamplifier. Thus, the output stage of the preamplifier should be capable of driving its signal into the large capacitance presented by the long interconnecting cables, and thus required to have low output impedance. The preamplifier does not provide any pulse shaping. The rise time of the output pulse is kept as short as possible, commensurate with the charge collection time of the detector itself while the decay time of the pulse is kept large (~ 50 or 100 μs) so as to allow the full collection of the charge from the detectors, with widely different collection times, before the onset of significant decay of the pulses.

2) **Shaping Amplifier:** pulse shaping amplifier modifies the shape and (increases) the amplitude of the signal. Another purpose of shaping amplifier is to increase the signal to noise ratio (S/N) by filtering out the noise from the signal. It includes fast amplifier in order to retain the timing characteristics of the pulse. The output from the shaping amplifier is fed into discriminator (fast output) and the ADC (described below).

3) **Discriminator:** discriminator distinguishes different types of pulses on the basis of their pulse height. It selects the minimum or maximum pulse height that is to be counted. The output of discriminator is fed into the TDC and/or scalar circuit for timing/counting as well as to the logic unit to generate triggers for the data acquisition.

4) **Analog-to-Digital Converter (ADC):** Analog-to-Digital Converter (ADC) is used for energy spectroscopy and converts the analog pulse waveform, with incorporated energy and timing informations, into digital data. The type of ADC commonly used in energy

spectroscopy is the peak sensing ADC. A peak sensing ADC accepts an input pulse, under specified trigger conditions, and produces an output value that is proportional to the peak amplitude of the input pulse. The performance of the ADC can be characterized by the speed with which the conversion is carried out, linearity of conversion and resolution of conversion.

Time-to-Digital Converter (TDC) converts the time difference between two pulses (start and stop) into digital data. It is thus applied to record the timing information of the input pulse.

5) **Scaler:** this is simply a device to count the number of any logic pulse in the pulse processing system. It is often applied to count the number of triggers generated for the data acquisition system or the discriminator output.

The schematic of the chain of signal processing is depicted in the figure below.

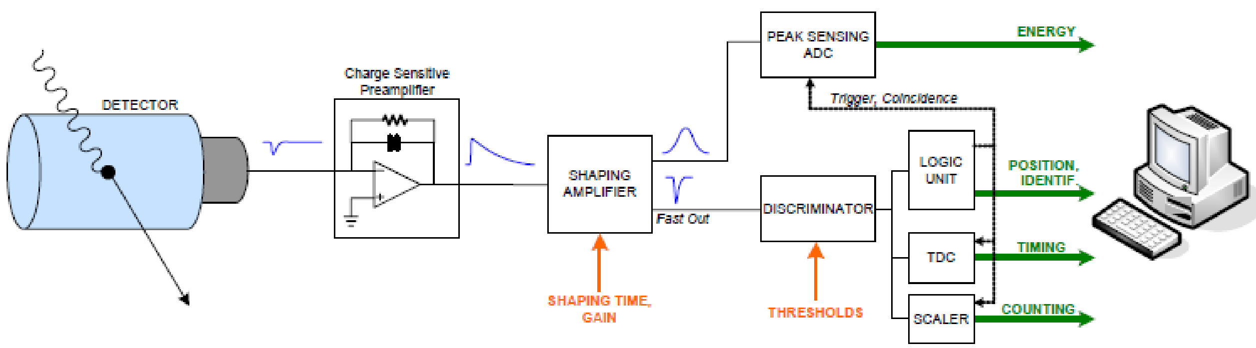
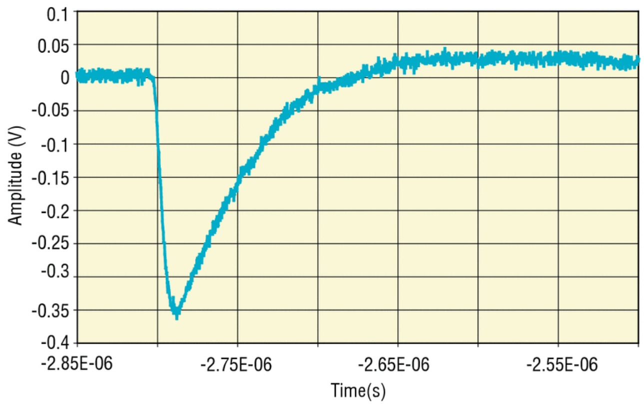
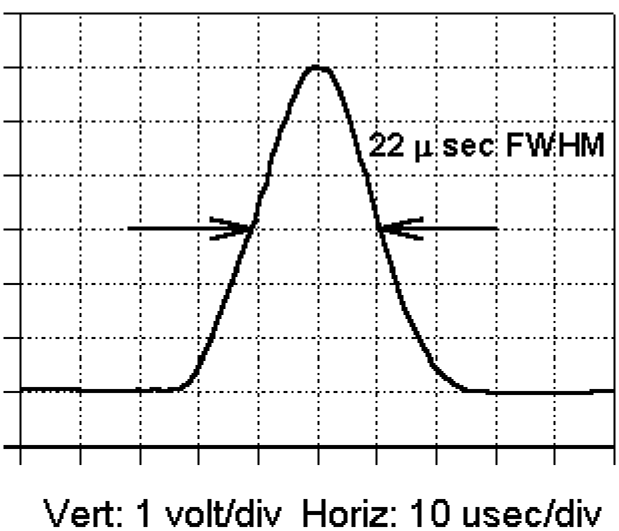


Figure-2.1 Schematic diagram of the signal processing chain.

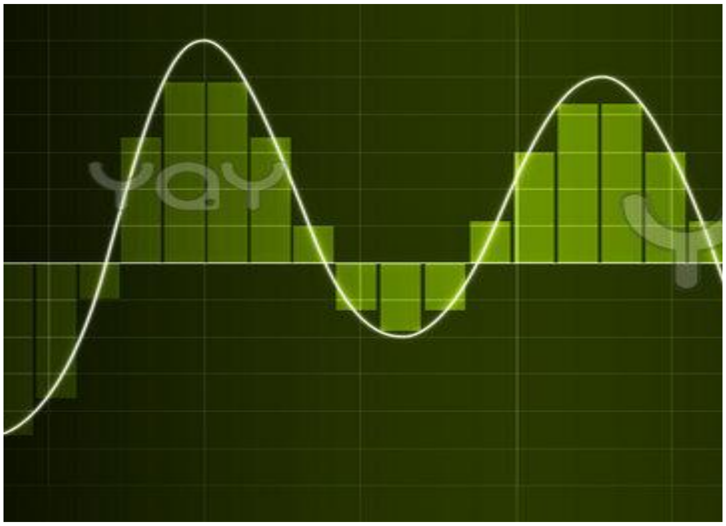
To remove the noise from contaminated pulse, optimal digital filter [5] can be designed who gives the best signal to noise ratio (S/N), where amplitude is proportional to the measured value. For example we know that cusp filtering is ideal for exponential input [6], which are produced by standard charge sensitive preamplifier with resistive feedback. In general optimal filtering is not possible but that close to optimal filtering or shaping from the detector is realized. Previously close to optimal filtering or shaping was performed only by analog electronics, for example, Gaussian pulse shaping or filter has been used for past few decades in conventional or analog electronics. The set of figures below illustrates the processing undertaken to extract the information from these signals emerging out of the detector and being processed by the conventional electronics.



The output from the detector is obtained via a pre-amplifier is a sharply rising pulse whose height contains the information regarding the energy of the incident radiation, and whose rise time allows us to perform timing spectroscopy. The pulse has a relatively long fall time.



The output from the preamplifier is shaped (to improve the S/N ratio) and then amplified. This processing is usually referred to as "slow processing", as the time range is a few micro-seconds. The Gaussian pulse shaping has been exploited by nuclear spectroscopy for more than 4 decades.



These analog signals are finally digitized and stored in the computer for a detailed off-line (at a later stage) analysis.

Figure-2.2. Signal processing with conventional electronics.

2.2. Digital Signal Processing (DSP)

In our present scenario, due to development of fast analog to digital converter (ADC), digital signal processors (DSPs), field programming gate array (FPGA) it is possible to digitize them even after preamplifier or phototube and process them in real time. To extract quantities of interest like pulse height, pulse shape, pulse arriving time etc. DSP techniques were soon explored in nuclear spectroscopy, and the first digital pulse processors for X-ray and γ- ray spectroscopy appeared on the market in the last decade of 20^{th} century [7].

During these last years, the analog to digital converters (ADCs) as well as more complex digital signal processing units are increasingly being used for nuclear spectroscopy applications. This is due to the fact that these systems are becoming fast enough for applications in the digital real time processing of nuclear radiation detection data. The recent R&D approach is to simply digitize the incoming pulse signal (at sampling rates over 50 Mega samples-per-second) from a radiation detection system and deconvolute it into individual pulses on-line in real time. This might be accomplished by taking several deterministic approaches including fitting the intensity and delay time of individual convolved pulses by a linear-nonlinear least- squares method or by a sophisticated transform approach. It is expected that soon there will be possibility to eliminate almost all pulse pile-up distortion associated with nuclear radiation spectroscopy with a dedicated PC combined with ADC for an individual detector system.

The application of DSP systems leads to new possibilities in processing of measured data, development of automatic measurement setup, and offline data correction related to other, measurement parameters temperature and magnetic field influence. New software for processing the detector data can replace the task of analog modules such as single channel analyzer, constant fraction discriminators, time to amplitude converters and /or multichannel analyzer. Another role is to add new possibilities such as automatic measurement setup capability and off-line data filtering. The significant advantage of DSP is to use different approximation method without the need of repetitive measurement as in analog system. For example small change in channel window or other parameter in analog system requires repetitive measurement. But in digital system same set of data can be acquired by relevant software

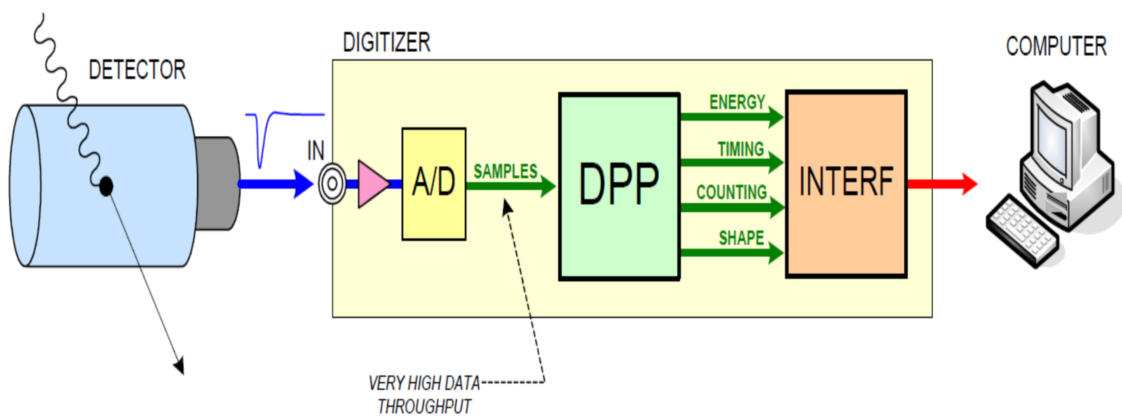
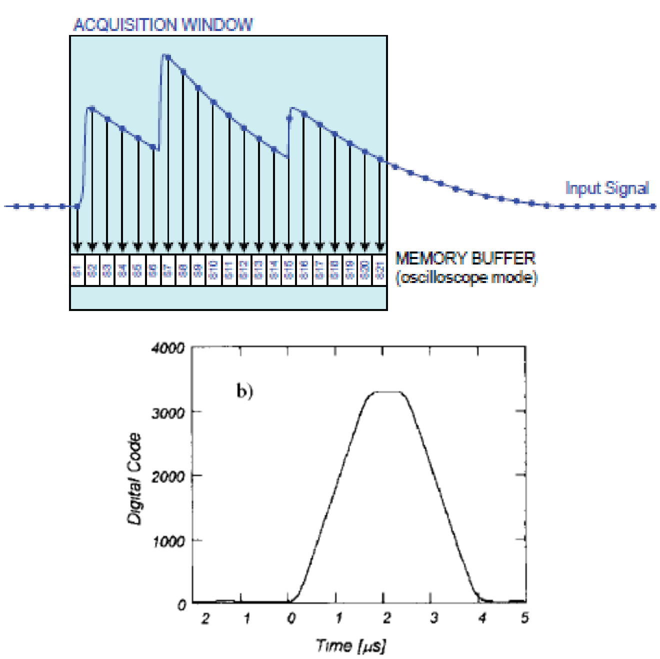


Figure-2.3. Conceptual block-diagram of a digital equivalent of the analog signal processing chain.

In the figure we have shaper for conditioning of the signal and digitizer to digitize the signal for the further processing of the signal. The major problem of the completely digital approach is that the amount of data to read out is extremely high and some elementary online data reduction (zero-suppression) is essential. The input data are continuously sampled and stored in a circular buffer and can be retrieved after a trigger. A field programmable gate array (FPGA) will handle the readout, trigger decisions, and simple signal processing. .Depending on the I/O and processing requirements the FPGAs can be replaced or augmented by DSPs for more complex calculations. These programmable units would extract the required physical quantities from the signal such as energy, time, position or particle identification. Complex triggers such as multiplicity or sum energy can be implemented digitally with a small latency. Data from this processing unit are transferred to a host computer or storage medium via a data and control bus. Generally, such a DAQ system can be characterized by quantities such as number of parameters, number of channels, event rate, event length, trigger rate, allowable dead time, etc., which depend on the physics requirements. Accordingly, these systems will have different input characteristics (voltage/current input, AC/DC coupling), dynamic range, bandwidth, noise, and sampling rate as well as processing and data transfer capability [8].

The important steps in the digital processing are illustrated in the figure below.



During a pre-set time window the signal is sampled at the desired sampling rate (usually few hundred Mega samples per second [16].

The sampled input pulse is then processed by a trapezoidal filter to obtain the following shape which is very similar to the output of the conventional Spectroscopic Amplifer [17]

Figure-2.4 Digital Signal Processing.

2.3. Data Acquisition System

Data acquisition is the process of measuring an electrical or any other physical phenomenon like voltage, current, temperature, pressure etc. by sampling the signal involved into equivalent digital numeric value that can be further processed by computer. A data acquisition system consists of multiple block that to have different function and all the blocks are integrated to sense physical variables, by using transducers to convert the physical variable to an electrical signal, then condition that electrical signal to make it readable by an analog-to-digital converter. After being converted by the A/D, a computer can then read that data and process, analyze, store and display the acquired data with the help of software. Block diagram of a typical data acquisition system is shown below.

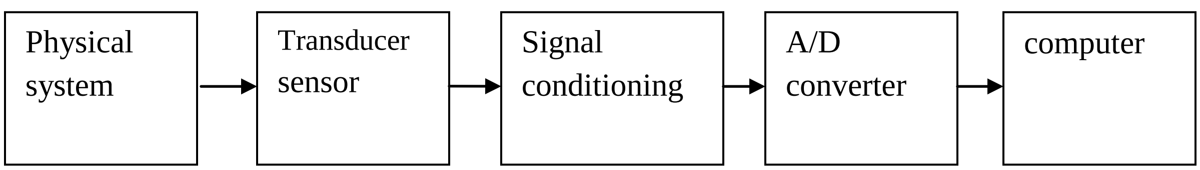


Figure-2.5 Block diagram of data acquisition system.

At times we use some pre-trigger conditions to reduce the data rate, so that we do not end up with a data-rate that is higher than the accepted handling/processing rate of the computer hardware. This ensures minimal dead-time of the system, and prevents loss of genuine data. Most components of a data acquisition system have a minimum amount of time in which two input signals may be resolved as separate events. If one input is being processed by a device and another input arrives before the minimum required temporal separation, the second event may be lost, or cause a pile-up effect within the device. This minimum required temporal separation of input signals is known as the dead time of a device.

2.4. Data Acquisition Standards

In present scenario of data acquisition we use three types of data acquisition standards. The first one is NIM standard and this standard was used to provide power to the electronics. The CAMAC standard was introduced which allowed data transfer from the backplanes of the crates. The current standard is VME standard which is the extension of CAMAC with more options and faster bus speed [9]. These standards are briefly explained below

2.4.1 NIM (Nuclear Instrumentation Module)

NIM standard is the first crate standard which is widely accepted by the nuclear industry scientists. It provided a way to provide power to its electronics. In this power is distributed through series of connectors on the back of the crate. NIM crates have 12 slots and typical voltage distributed is +/- 6 V, +/- 12 V or +/- 24 V. However, many customized power supplies is made for NIM crate. NIM module cannot communicate with each other through backplane of the crate. One of the major concern of the NIM crate is the low channel density because NIM module have knobs and switches in the front side which take lots of space so to adjust parameter the number of channel per module is always low [10]. A illustration of a standard NIM bin is shown in the figure below.

Figure-2.6 NIM 8301 7U 12 slot smart fan unit 300/600 W crate [13].

Here in the above figure we can see that there are 12 slots present as mentioned earlier.

2.4.2. CAMAC (Computer Automated Measurement and Control)

CAMAC overcomes many shortcomings of NIM standard by providing back plane that can transfer data. That means we can adjust each parameter through the back end. Each CAMAC crate has 25 slots means twice as much modules than NIM can have. The last two crates are always reserved for CAMAC crate controllers. The backplane had one card style connector is present for each slot to which modules are plugged in. As the demand increased, more modules was needed with a higher channel density which CAMAC bus was not able to keep up and extension is made to the CAMAC standard which will be illustrated later [10]. A CAMAC crate is shown below.

Figure-2.7 A CAMAC crate [12]

2.4.3. VME (Versa Module Europa)

VME is a flexible open ended bus system .Like CAMAC, where rear backplane is offered, VME also offer rear backplane for fast and efficient data transfer. However the main advantage of VME is that it offers more efficient and versatile bus. The faster bus system means more data can be transferred and channel density increases.VME crates have slots for 21 modules where first one is made for crate controller. The VME back plane has two 3-row connectors per module and these are for data as well as power distribution [10]. A typical VME chassis is shown in the figure below.

Figure-2.8 A typical VME chassis [11].

2.5. Advantages of using Digital Technology in Nuclear Spectroscopy and New Possibilities.

Now we will discuss why DSP techniques in nuclear spectroscopy is more superior than its analog counterpart why should we use DSP technique rather than analog technique what are its advantages new possibilities and what are its disadvantages in comparison with analog technique. These are the few questions which will be discussed in this section.

Advantage of using DSP technology in nuclear spectroscopy has prominent effect on following

Energy resolution: with the advanced DSP technology electrical pulses are digitized earlier that signifies that less analog components are used which in turn improves noise immunity, temperature stability and due to this we will get better energy resolution [6].

Reduced size: by digital technology higher density and low power integrated circuits can reduce the size of the instruments and make it portable which is very useful in space application and where large no of detectors are present it may reduce the no of cable crates. Because state of art of equipment used in nuclear can have size a desktop to as big as a accelerator so reduction in size have an edge [6].

Throughput: in nuclear spectroscopy usually pulses which come from the detector does not arrive uniformly. Thus, if high counting rate is involved then overlapping of two or more pulses (pile-up) will occur. With analog technology it is difficult task to separate them and one or more pulses may get rejected which decreases the number of processed events that is the system throughput. To keep throughput high we should decrease the width of the pulses but this will reduce the energy resolution which is not affordable and since pileup rejection is more efficient through digital technology it is have an edge over analog technology [6].

Easy upgrade and protection of intellectual properties: today's digital circuit chips can be reprogrammed while be in circuit without any soldering or hardware reduction and logics are also protected against any reverse engineering so intellectual properties is secured [6].

Automation of critical adjustments: in nuclear spectroscopy several parameters are to be adjusted in order to keep the setup compatible for certain type of detectors. For this reason analog systems has various switches knobs and peripherals to adjust according to the purpose and this requires manual intervention and expertise for best selection. In contrast, a digital system can be programmed to do critical adjustments while internal modules can be saved and recorded on hard disk. In case of complex experiment this features can minimize setup errors [6].

Multifunction operation version control and diagnostic capability: digital system can perform multiple tasks in parallel. For example, a digital spectrometer may feature multi-channel analyzer, multichannel scaling and oscilloscope simultaneously.

Components of digital filters are parameter coefficients which is stored in memory or registers this in turn confirms that parameters of filter will not change from instrument to instrument and precision of digital filters are dependent upon its coefficient while in analog system its depends upon the circuit element which suffers from the ageing mean parameters of analog filter may change over time.

It is common for nuclear spectroscopy instruments to implement number of software and hardware diagnostic to identify nature of any fault in operation. This includes D.C line faults, low internal battery voltage, detector faults, mains failure microprocessors fault etc [6].

Now we will discuss some of the extra performance that can be achieved by digital technology which is not explored yet due to some limitation in current technology and these are enlisted below

Pulse shape discrimination and particle identification with scintillator detectors: in response to different ionizing radiation, some advance scintillator detector can produce different light components that can have decay time of several hundred nanoseconds. The high speed and high resolution ADCs are typically of 14 bit, 100-150 MHz. However, the aforesaid scintillator applications require fast charge sensitive preamplifiers with sub nanosecond rise time, faster and higher resolution ADCs with sampling rate of several giga samples per second and faster FPGAs for continuous time processing [6]

Nonlinear DSP algorithms for amplitude filtering: in order to increase throughput linear filtering technique used in present scenario decreases the pulse shaping time constant. However, a point will be reached from where majority of data will be corrupted due to pile-up of the pulses and which in turn degrade the throughput or energy resolution or both. Using the multistage non linear DSP algorithm this method is able to decode pileup events in real time improving the count rate throughput and energy resolution but due to the complexities of the algorithms to be used [14] , and the due to the complex calculations involved, this technique is not yet explored.

Time coincident measurements in the picoseconds range: analog system can achieve timing resolutions below 200ps, while using scintillator detectors and photomultiplier tubes it has been show recently [15] that the hardware ‘intrinsic’ time resolution with a typical digital system can be lower than several tens of picoseconds and limited by timing jitters. The system compromised 14-bit 75 Mega samples per second ADC and FPGA, and was tested with exponential input pulses from a pulse generator with a 50 ns rise time and a 2.5 μsec decay time. The resolution was an order of value worse than when using real pulses from a photomultiplier tube. The improvements can be obtained by reducing the sampling time and keeping the effective number of bits or improving the algorithm [6].

2.6. Disadvantages of Replacing Analog Technology with Digital Technology.

The limitation which is posed by digital technology in comparison to analog technology is enlisted below [6].

Limited amplitude precision: we know that value of measured amplitude is dependent on quantization precision and sampling rate the higher the sampling rate lower is the quantization precision and we know that sampling rate is quite higher in nuclear spectroscopy in the order of several mega samples per second. So this poses certain limitation.

Rounding effect: some kind of filters has stability problem example is IIR filter due to rounding errors that are caused by finite mathematical precision during quantization. So this presents limitations of using digital technology.

Complex, expensive and specialized design tools: in order to perform mathematically complex digital filter with FPGAs one should have good working knowledge of hardware and description language and FPGA architecture. On the other hand, some design tools like XILINK®, MATLAB® and SIMULINK® are available but they are too expensive.

Repairing and maintenance and non adequate specialized training in spectroscopy: many of the electronics components essential for nuclear spectroscopy are non functional if not configured properly so a simple replacement is not going to fix the fault component. And due to the circuit complexity fault identification is also a very much difficult task which need expertise and the proper training is needed which is not adequate specially in the developing country. So a simple replacement is not the solution because it is not economic.

Apart from the leading vendors in FPGA and DSP chips that offer frequent up to date and high quality training to their users of their tools and products, alternate and unified course on digital spectroscopy are rare.

The disadvantages notwithstanding, digital signal processing is in growing demand in the contemporary nuclear physics experimental research. In the following chapters, we shall describe the recursive algorithms for real-time digital pulse shaping, widely applied in the domain of nuclear spectroscopy.

Chapter 3

Algorithms for Digital Pulse Shaping

The objective of this chapter is to study the general principle of pulse processing in nuclear spectrometry. We know that pulses are of two types, one being the linear pulse which carry information through its amplitude or shape while the other being a logic pulse which holds information merely by its presence or absence. Generally linear pulses are produced by radiation interaction and converted into logic pulses.

3.1. Pulse Shaping

The question arises why we need to do pulse shaping and answer for this is that pulses produces by the radiation detectors carries information by virtue of its amplitude or shape. These pulses that are produced by the preamplifier and fed into the signal processing unit are generally small in amplitude and due to inherent noise of detectors and preamplifier useful information may get lost. Thus, in order to retain the amplitude and shape we need pulse shaping. Typical example is illustrated in Fig. 3.1.

For complete charge collection preamplifiers are normally adjusted to give a decay time for the pulse which is quite long (typically 50µs) [18]. Now if the rate of interaction of detector is not small these pulses tend to overlap on one another. And as discussed earlier that the information carried out by pulses are generally done by virtue of its amplitude, the pileup of pulses on the tail of preceding pulses which have not fully decayed to zero, can lead to serious error. And since the time spacing between nuclear pulses is random, each pulse can be superimposed on the residual pulse and so the amplitude is no longer a good measurement of Q (charge deposited in the detector). So for information not to be lost, we must shape the pulse by eliminating its long tail due to large decay time constant without affecting the amplitude of the pulse. Example is shown in the figure below where pileup process and its shaped output is shown. We can clearly see that in the output the long tail has been eliminated.

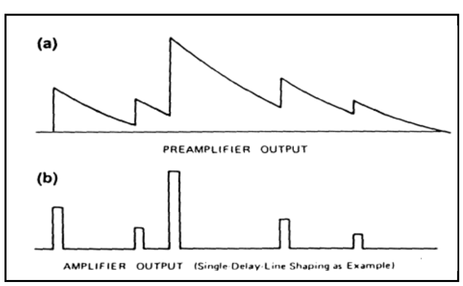


Figure-3.1 Pile up and its eliminated output [19].

3.2. Recursive Algorithm in Digital Signal Processing

Recursion in computer science is a method where the solution to a problem depends on solutions to the smaller instances of the same problem. Thus, a recursive algorithm is an algorithm which calls itself with “smaller (or simpler)” input values, and which obtains the result for the current input by applying simple operations to the returned value for the smaller (or simpler) input. Recursive algorithms for digital pulse shaping in pulse height measurements have been developed in early 90s. The practical realization of the digital processors depends on the complexity of the algorithms they implement. It is generally desired that the algorithms be suited for real-time implementation apart from being commensurate with the requirements for high resolution and high throughput spectroscopy.

The subsequent sections in this chapter describe recursive algorithms for synthesis of optimal pulse shapes. Application and results of these algorithms would be discussed in the next chapter.

3.3. CR-RC Pulse Shaping

The simplest form of pulse shaping is CR-RC pulse shaping. This filter is the combination of CR filter followed by RC filter. CR filter is also known as high pass filter while RC filter is known as low pass filter brief explanation of CR and RC filter is explained below.

3.3.1. CR Filter

First let us take CR filter circuit diagram is shown below.

Time constant (τ) =R$\times$C

Figure-3.2. Circuit diagram for CR filter.

Mathematical equation describing CR shaping is

$$v_{in}(t) = \frac{Q(t)}{C} + v_{out}(t)$$

Taking differentiation on both sides we have

$$\frac{dv_{in}(t)}{dt} = \frac{i(t)}{C} + \frac{dv_{out}(t)}{dt}$$

Now taking $v_{out}(t) = i(t) \times R$ and $\tau = R \times C$ we have

$$\tau \frac{dv_{in}(t)}{dt} = v_{out}(t) + \tau \frac{dv_{out}(t)}{dt} \qquad \text{………………………………………… (3.1)}$$

Now if we make time constant τ small we can neglect second term of the right hand side of the equation (1) so our equation is reduced to

$$v_{out}(t) \cong \tau \frac{dv_{in}}{dt} \qquad \text{…………………………………………… (3.2)}$$

From the above equation (3.2) we can see that the output is proportional to time derivative of the input voltage v_{in} hence we can also call CR filter as DIFFERENTIATOR.

Now if we take input pulse from preamplifier as idealistic pulse which have sharp rise time with long tail we take step voltage as input to the shaper.

So we have

v_{in} = V for (t$\geq$ 0)

v_{in} = 0 for (t<0)

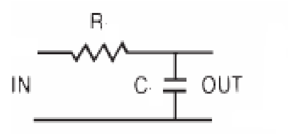
Now after applying the step input to the shaper we have

$$v_{out}(t) = Ve^{-t/\tau}$$

Shape of the output is shown in the next chapter.

3.3.2. RC Filter

Circuit diagram of RC filter is shown below



Time constant(τ)= R$\times$C

Figure-3.3 Circuit diagram for RC shaper

Mathematical equation describing RC shaper is shown below

$$v_{in}(t) = iR + v_{out}(t) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \; (3.3)$$

Now,

$$i = \frac{dQ}{dt} = C\frac{dv_c}{dt} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \; (3.4)$$

v_c = capacitor or output voltage so we can write

$$i = C\frac{dv_{out}(t)}{dt}$$

Now by combining equation (3.3) and (3.4) and providing time constant (τ) =RC we have

$$v_{in}(t) = \tau\frac{dv_{out}(t)}{dt} + v_{out} \quad \ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots\ldots\ldots\ldots\ldots.. \; (3.5)$$

After rearranging the equation (5) we have

$$\frac{dv_{out}(t)}{dt} + \frac{1}{\tau}v_{out}(t) = \frac{1}{\tau}v_{in}(t) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \; (3.6)$$

Now if RC is made sufficiently large only the first term of the left hand side of the equation (3.6) is of significance so our equation is reduced to

$$\frac{dv_{out}(t)}{dt} \cong \frac{1}{\tau}v_{in}(t)$$

Or,

$$v_{out}(t) = \frac{1}{\tau}\int v_{in}(t)dt$$

Here we see that provided time constant RC is made large the output is integral of input hence RC filter is also called integrator

Now if input is provided with step voltage

$$v_{in} = V \quad \text{for } (t \geq 0)$$

$$v_{in} = 0 \quad \text{for } (t<0)$$

Output is given as,

$$v_{out} = v_{in}(1 - e^{-t/\tau})$$

The out plot of the RC shaper will be shown in next chapter.

3.3.3. CR-RC Filter

Circuit diagram of CR-RC filter is shown below

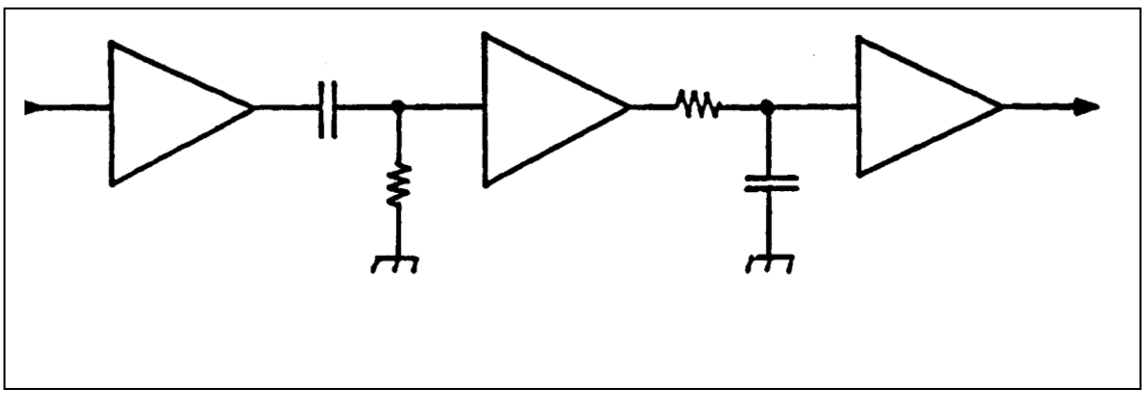


Figure-3.4. Circuit diagram for CR-RC shaper

Here first unit is a CR filter which is followed by the RC filter as discussed earlier.

Digital Recursive algorithm [20] which describes the CR-RC filter is given below

$$y[n]_{cr-rc} = 2\alpha y[n-1] - \alpha^2 y[n-2] + x[n] - \alpha(1+at)x[n-1] \quad \text{....................(3.7)}$$

$a = \dfrac{1}{\tau}$, $\alpha = e^{-t/\tau}$ here τ is the time constant of the circuit

$x(n)$ = input pulse

3.4. Triangular Shaping

Triangular shape is hard to implement from analog hardware so it is digitally implemented and the recursive algorithm [21] that implement triangular shaping is shown below.

$$y(n) = y(n-1) + x(n) - 2x(n-k) + x(n-2k) \quad \text{................................. (3.8)}$$

$x(n)$ = input pulse

Here k = rise time of the output pulse. After applying this recursive filter algorithm to the input will be shown in next chapter.

3.5. Trapezoidal Shaping

For detectors where charge collection time is variable depending upon the position of radiation interaction ballistic effect (amplitude get attenuated) is inevitable but this can greatly be reduced if we provide output shape as flat top and trapezoidal filter is good choice for that . We can implement trapezoidal filter both by analog or digital means [18]. A digital recursive algorithm [21] is shown below for the implementation of the trapezoidal filter.

$$y(n) = y(n-1) + x(n) - x(n-k) - x(n-l) + x(n-l-k) \quad \text{.................(3.9)}$$

$x(n)$ = input pulse

$l = k + m$ is the delay, m = width of flat top, k = rise time of output pulse.

Output pulse is shown in next section with explanation and various consequences.

3.6. Flat Top Cusp Shaping

Cusp shaping is similar to trapezoidal shaping where flat top is provided but one of the advantages of cusp shaping is that it provides optimum shaping [6] with exponential signal as input. A digital recursive algorithm [21] is shown below for implementation of cusp shaping.

$$y(n) = y(n-1) + p(n) - [x(n-k) + x(n-l)]k - x(n-l) + x(n-l-k) \quad \ldots\ldots\ldots(3.10)$$

$$p(n) = p(n-1) + x(n) - x(n-k) + x(n-l) - x(n-l-k)$$

$l = k + m$ is the delay, m = width of flat top, k= rise time of output

$x(n)$ = input pulse.

Output shape of cusp shape is shown in the next chapter.

This chapter provided the theoretical framework for the functioning of the digital signal processors, based on the various recursive algorithms. These algorithms shall be eventually tested on different types of pulses, both ideal as well as realistic, in the next chapter.

Chapter 4

Results and Discussions

In previous chapter we have introduces pulse shaping and different types of filters (Trapezoidal, Flattop Cusp, triangular, CR-RC) that can be applied to the input signal and its mathematical formulation. In this chapter we shall discuss about various types of simulated input signals (through MATLAB® platform) subjected to various forms of pulse shaping techniques, realized through the recursive algorithms discussed in chapter 3, and check for the resulting output.

4.1. Step Pulse

First let us take step function as input pulse and apply filters, discussed earlier. Step pulse can be considered as idealistic pulse with a very fast rise time and infinite tail. We represent step pulse mathematically as shown below.

$$V\,(t-t_0) = V \text{ for } t \geq t_0$$

$$V\,(t-t_0) = 0 \text{ for } t < t_0$$

Simulated step pulse of amplitude 100 is shown below with time delay of 1000.

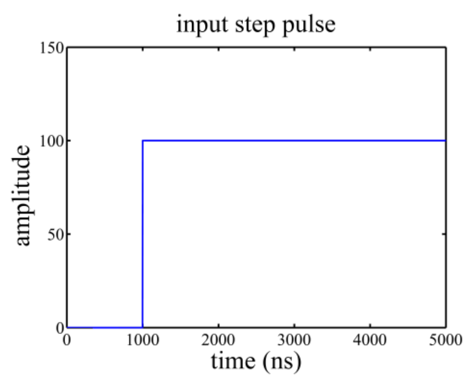


Figure - 4.1 Simulated step pulse

With this simulated step pulse we shall now investigate the effects of various filters.

4.1.1 CR Filter

If we apply CR as discussed in equation (3.3) to the step input with time constant of the shaping τ = 1.6μs. Then the simulated output will be like figure shown below. The corresponding algorithm as implemented in MATLAB® is given in Appendix A-1.

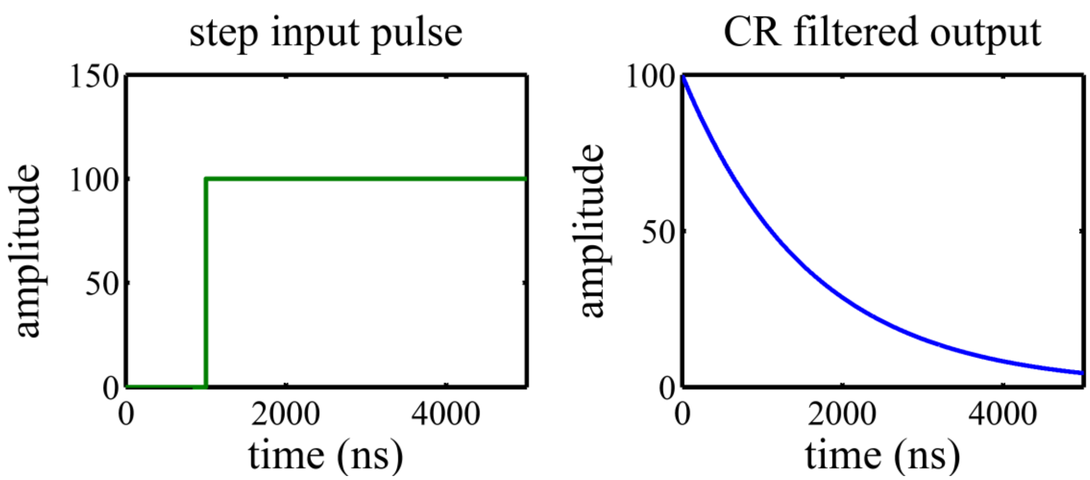


Figure - 4.2 CR filtered pulse with step input pulse

Above figure approximately represents the shaping of idealistic input signal i.e. step pulse (having very fast rise time). Here leading edge of the step function is not differentiated i.e. remains untouched because time constant τ > rise time of the input pulse. Here we see that amplitude of the signal is also not affected if τ > finite rise time of the input pulse.

4.1.2 RC Filter

Simulation of RC filter with time constant τ = 1.6 μs is shown below. The corresponding algorithm as implemented in MATLAB® is given in Appendix A-2.

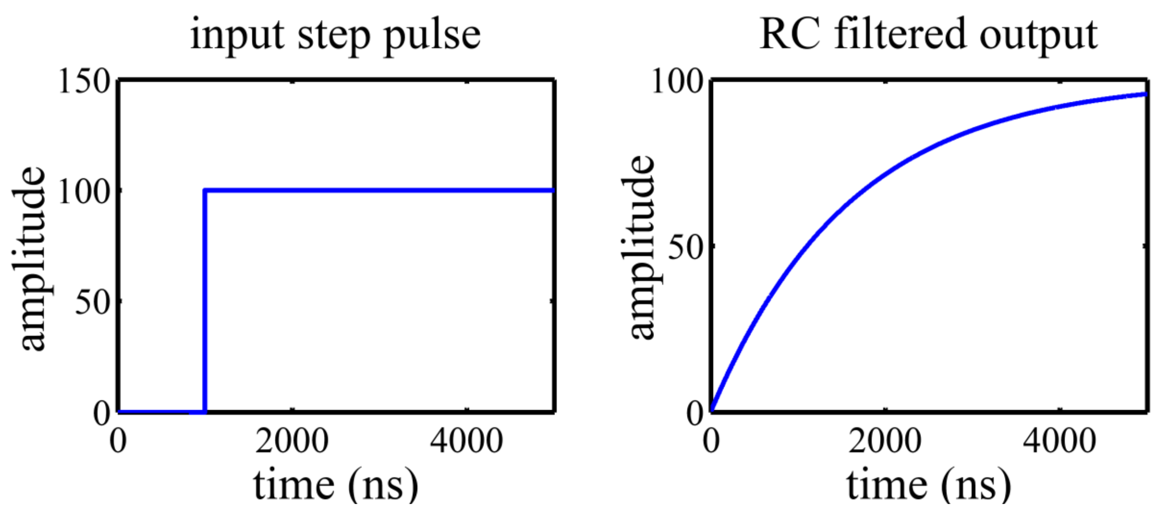


Figure - 4.3 RC filtered output with step input

Here we see that the output shape starts linearly as a ramp function but over substantially long time scale the time constant of the circuit is no longer large by comparison and eventually output shape tends to input step as limit.

Now we combine both the CR and RC filter to a single unit as CR-RC filter, in the next section.

4.1.3. CR-RC Filter

After implementing recursive algorithm shown in equation no (3.8) on the step function we have simulated CR-RC pulse which is shown below for time constant τ = 3.5ns. We get a semi Gaussian shape as shown below. The corresponding algorithm as implemented in MATLAB® is given in Appendix A-3.

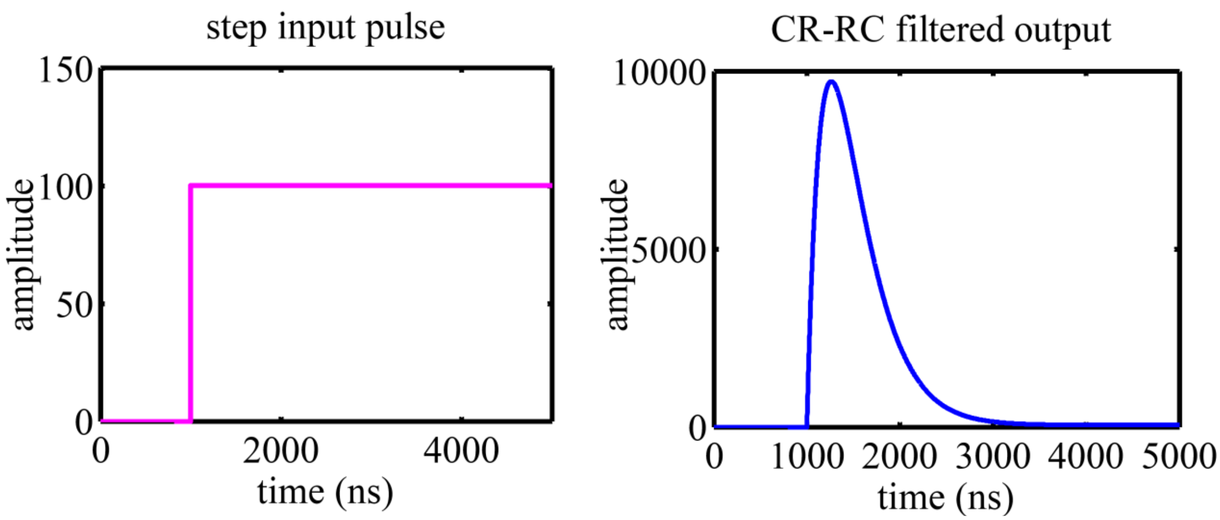


Figure - 4.4 CR-RC shaped output pulse.

4.1.4. Triangular Filter

Triangular pulse shape is desirable because it’s have good signal to noise (S/N) ratio and short duration which minimizes pulse pile up and suitable for high counting rates measurement. Symmetrical triangular shaping is not possible through analog processors however it’s possible in digital pulse processor using recursive algorithm shown in equation (3.9). a simulated triangular filters output shape with the rise time of τ = 2μs to an step input signal is shown below. The corresponding algorithm as implemented in MATLAB® is given in Appendix A-4.

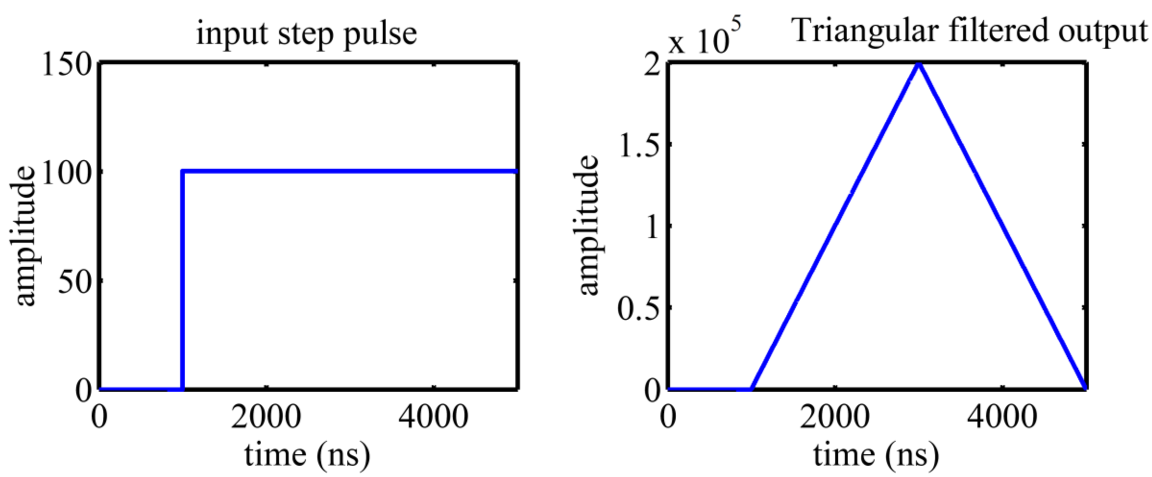


Figure - 4.5 Triangular output pulse to step input

4.1.5. Flat Top Cusp Filter

Here we simulate flattop cusp output shape with step pulse as an input by the recursive algorithm shown in equation (3.12) the figure below shows the same. Parameters are rise time of the output pulse = 1000 ns flattop width = 400 ns. The corresponding algorithm as implemented in MATLAB® is given in Appendix A-5.

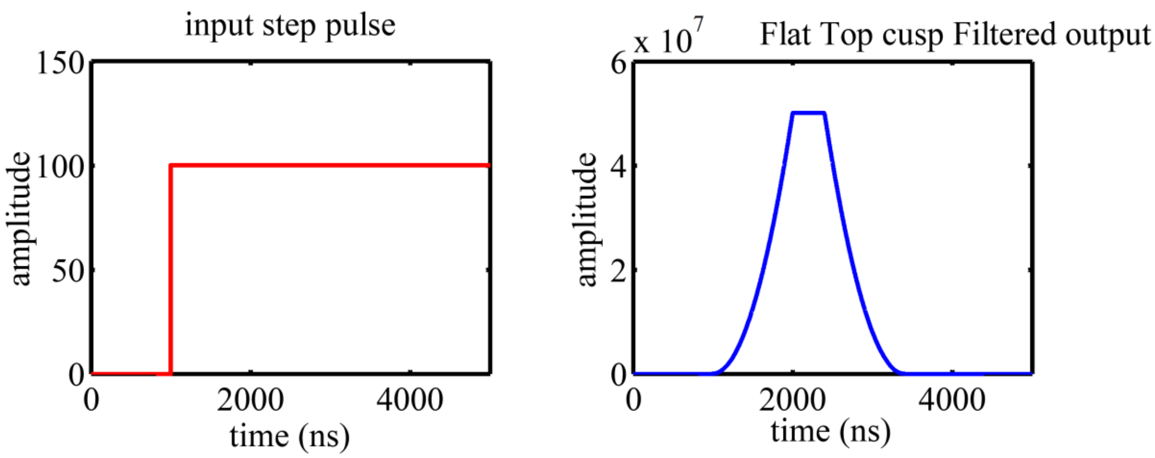


Figure - 4.6 Flat top cusp shape to step input.

4.1.6. Trapezoidal Filter

Trapezoidal filter is been discussed in chapter 3 now we will simulate trapezoidal filter on a step input parameters of trapezoidal filtering is given below. The corresponding algorithm as implemented in MATLAB® is given in Appendix A-6.

Rise time of output pulse= 1500 ns, flattop width of trapezoidal = 800 ns. Shape is shown below.

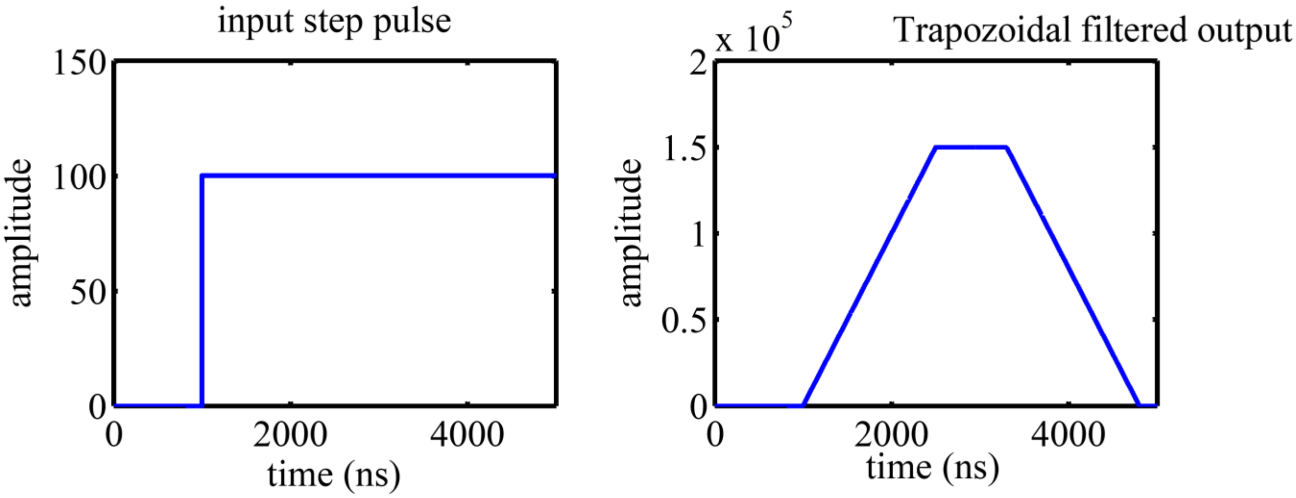


Figure - 4.7 Trapezoidal filtered output to a step input.

Till now we have discussed about idealistic pulse i.e. step input pulse. Next, we shall deal with realistic pulse produced by widely used detectors in nuclear spectroscopy scintillator detector and High Purity Germanium (HPGe) detector.

4.2. Scintillator Pulse

Scintillator detectors are widely in use in the contemporary nuclear spectroscopy. The six parameter function [22] that describes scintillator pulse shapes is given by the equation shown below [23].

$$x(t) = A[e^{(-(t-t_0)/\theta)} - e^{(-(t-t_0)/\lambda_s}] + Be^{(-(t-t_0)/\lambda_1)}$$

where A and B are the amplitudes of the short (fast) and long (slow) components at t=0, respectively λ_s and λ_1 is the decay time constant of short and long component respectively, θ s the third decay constant and t_0 is the reference time.

Table-4.1 Below shows the typical values for the parameters for gamma ray pulse [24].

Radiation type	A	B	Θ (ns)	λ_s (ns)	λ_1 (ns)	t_0 (ns)
Gamma ray	10.53	0.017	4.368	3.482	11.52	0.331

Simulated output of the scintillator pulse with above values of the six parameters is shown below. The function as implemented in MATLAB® is given in Appendix B-1.

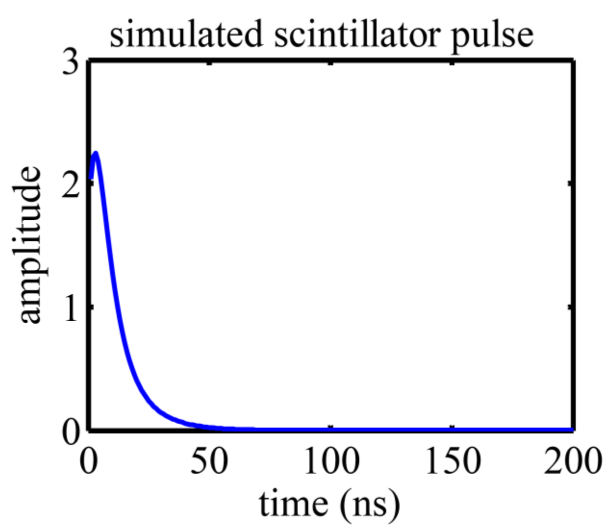


Figure - 4.8 Simulated scintillator input pulse

Now we apply the different filters and see the corresponding output pulse.

4.2.1. Trapezoidal Filter Applied to Scintillator Pulse

Here we apply the recursive algorithm shown in equation (3.11) to the scintillator pulse. The parameters are rise time of the output pulse = 70 (ns), flat top width = 35 (ns), M = 1. The corresponding algorithm as implemented in MATLAB® is given in Appendix B-2.

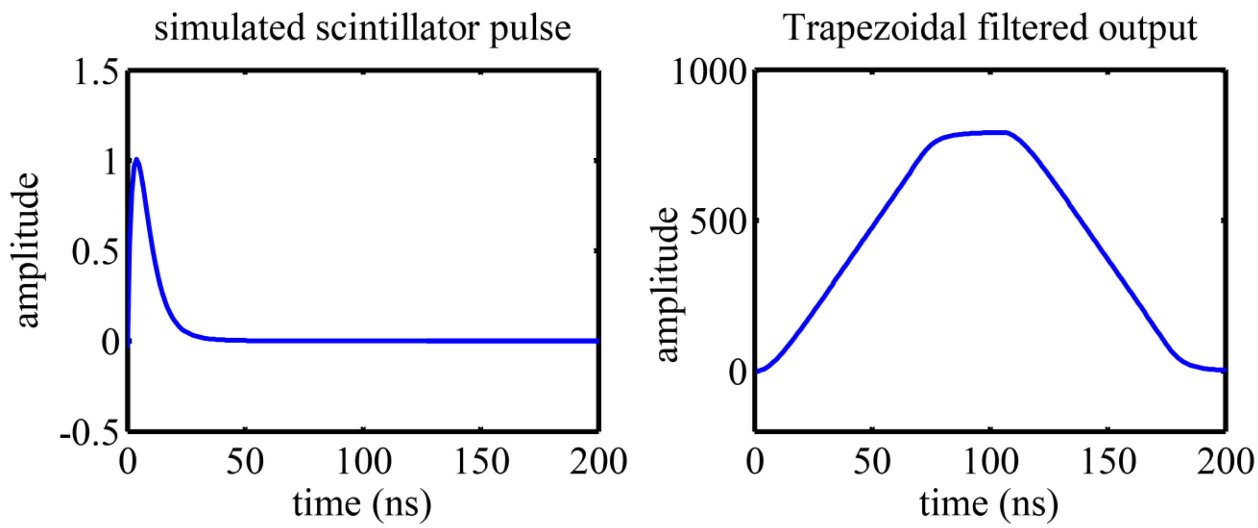


Figure - 4.9 Trapezoidal output shape with scintillator input pulse.

4.2.2. CR-RC Filter Applied to Scintillator Pulse

Here we apply CR-RC filter algorithm shown in equation (3.8) to the scintillator pulse. Time constant is taken as 1μs. The output pulse is shown below. The corresponding algorithm as implemented in MATLAB® is given in Appendix B-3.

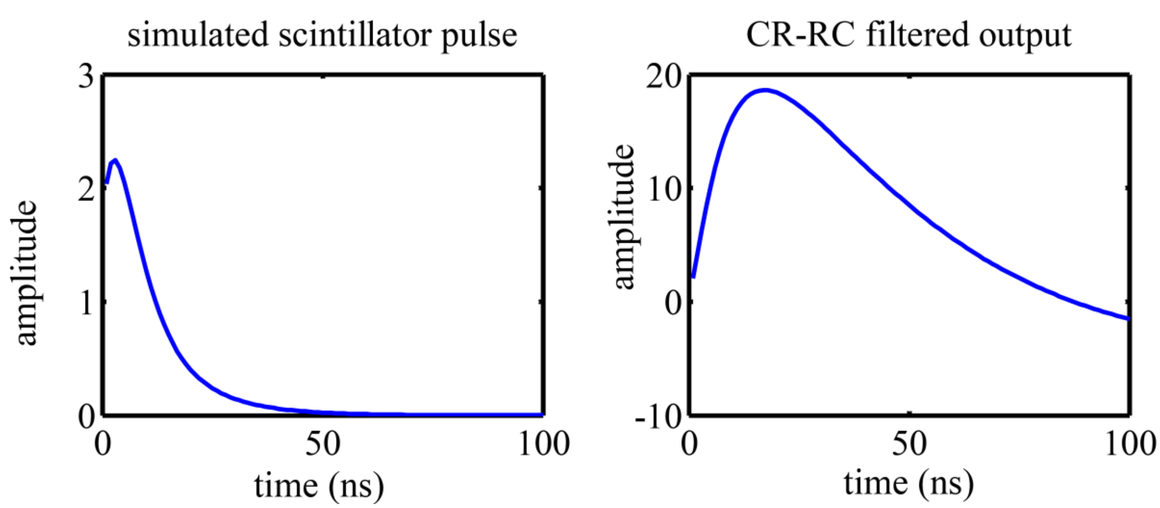


Figure - 4.10 CR-RC filtered output with scintillator pulse as input

Now we will vary the time constant of the CR-RC filter and will see the output pulse. Figure below shows the CR-RC filtered output with scintillator input. Here the time constant of the output pulse in colour violet, green, red is 0.5μs, 0.9μs and 1.2μs respectively. The corresponding algorithm as implemented in MATLAB® is given in Appendix B-4.

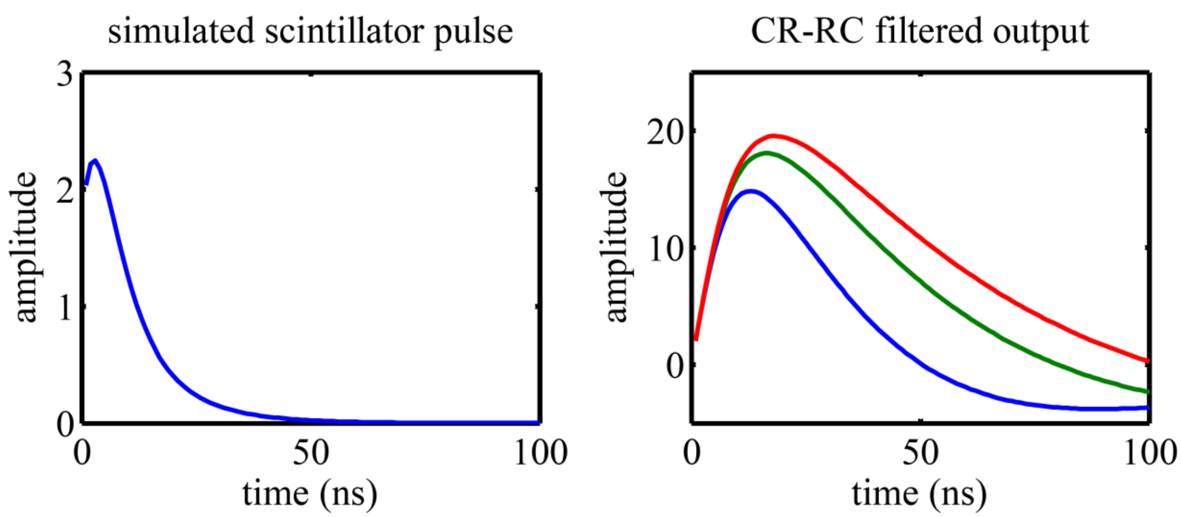


Figure - 4.11 CR-RC filtered shape of varying time constant to the scintillator input.

Here we see that the one whose decay time constant is large have greater amplitude.

4.3. HPGe Pulse

The HPGe detectors are extensively used in the modern nuclear spectroscopy, especially for their superior energy resolution. Mathematical expression that gives a HPGe (high purity germanium) pulse is shown below.

$$x(t) = \frac{l}{\tau - \theta}(e^{-t/\tau} - e^{-t/\theta})$$

Here θ and τ is finite rise time and decay time constant of the input pulseFigure below shows the simulated shape of the HPGE input pulse. With decay time constant τ of 1200 (ns) and finite rise time θ = 80 (ns). The corresponding algorithm as implemented in MATLAB® is given in Appendix C-1.

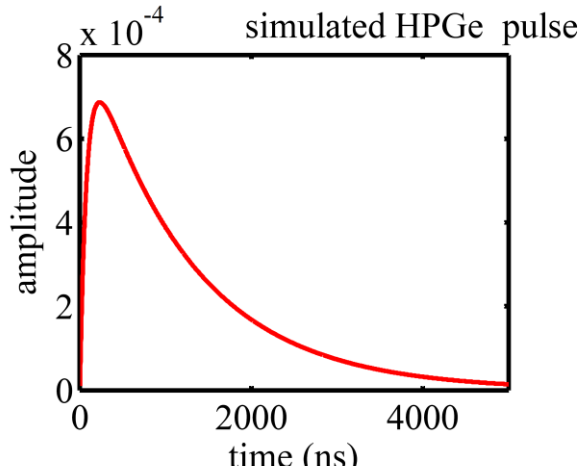


Figure - 4.12 HPGe input pulse.

4.3.1. CR-RC Filter applied to HPGe Pulse

CR-RC algorithm shown in equation (3.8) is applied on the HPGe pulse and the simulated output pulse is shown in the figure below parameters that we have used is decay time constant τ = 1200 (ns), finite rise time of the input pulse = 80 (ns) and the time constants are 25 (ns), 35 (ns), 45 (ns) for the violet, green and red coloured output pulse shown below. The corresponding algorithm as implemented in MATLAB® is given in Appendix C-2.

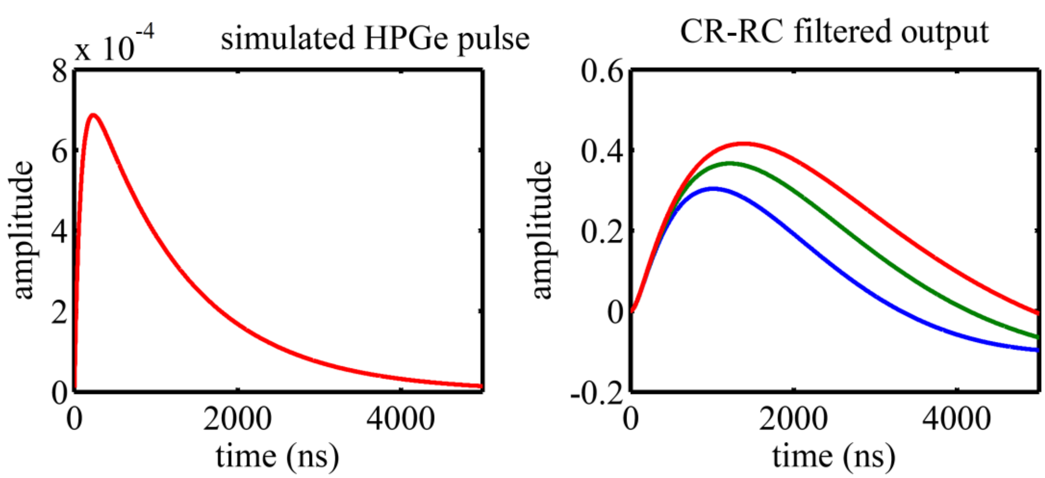


Figure - 4.13 CR-RC filtered output pulse with HPGe input pulse.

4.3.2. Trapezoidal Filter Applied to HPGe Pulse.

Using the algorithm shown in equation (3.11) we simulate a trapezoidal filter for a HPGe pulse. Figure below shows the trapezoidal filter when applied to HPGE input pulse the parameters are decay time constant of the input pulse τ = 1000 (ns), finite rise time of the input pulse θ = 100 (ns), rise time of the output trapezoidal pulse = 1000 (ns), flat top width = 300 (ns) and M = 1. The corresponding algorithm as implemented in MATLAB® is given in Appendix C-3

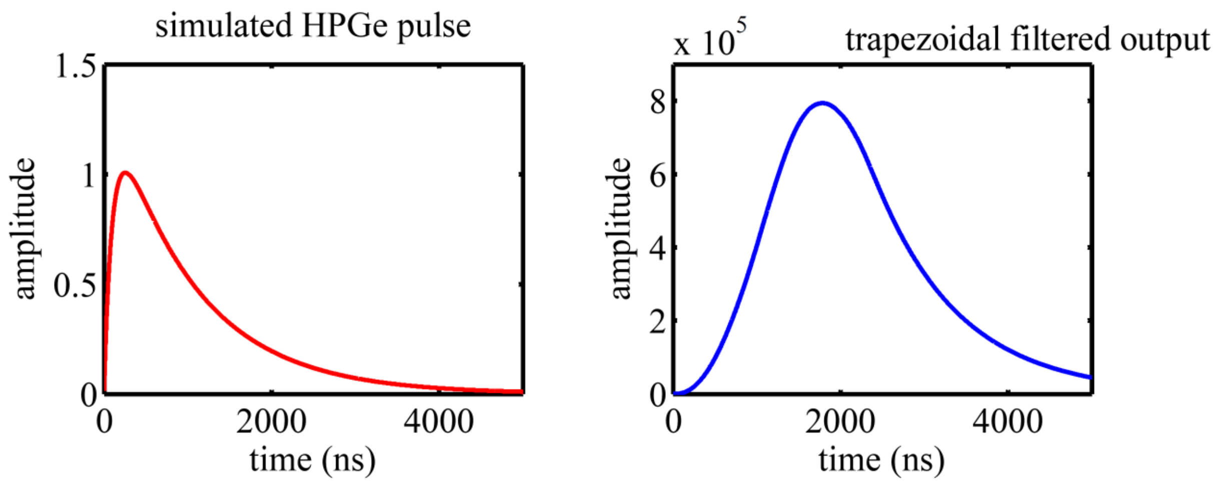


Figure - 4.14 Trapezoidal output to a HPGe input pulse.

Now we shall see how trapezoidal shaped output greatly reduces ballistic deficit (loss of signal if charge collection time of the detector > rise time of the shaper). Due to ballistic effect amplitude of the signal is reduced. Let us consider exponential signals with different rise time are coming out of the detector. Figure below shows the shape of the exponential input signal of different rise time. The corresponding algorithm as implemented in MATLAB® is given in Appendix C-4.

Parameters for the plots are decay time constant τ = 1200 (ns), and the varying finite rise time of the input pulse θ = 80 (ns), 140 (ns), 200 (ns), rise time and flattop of output trapezoidal pulse is taken as, 1200 (ns) and 700 (ns) respectively.

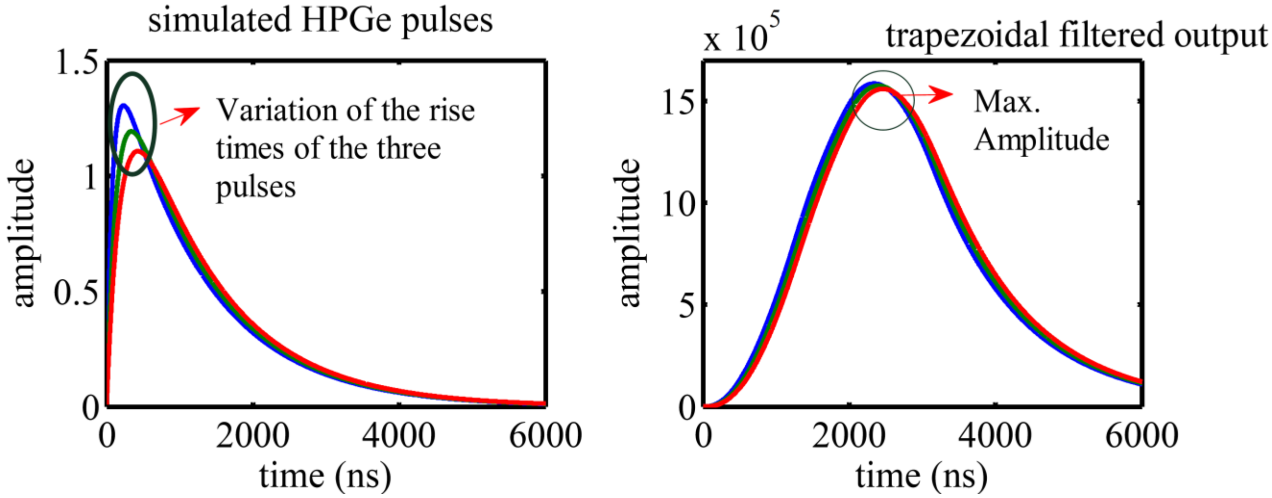


Figure - 4.15 Trapezoidal shaping for removal of ballistic deficit.

From the above figure we can see that the maximum amplitude of the shaped output is approximately same for all the three input pulses with different rise time. That means the output is free of the ballistic deficit over the prescribed range of variation of input rise time. Provided that time duration of flat top is sufficiently long to incorporate the rise time variation of input. Thus, trapezoidal shaping is thus preferred to compensate for the ballistic deficit and thus prevent the resulting loss of resolution.

We have thus investigated the effect of various types of pulse shaping techniques, realized through recursive algorithms, on pulses, both ideal (step) as well as realistic (scintillator and HPGe). The choice of the values of different parameters was based on the consideration of best representation of the input pulse. However, actual test of the parameter set can only be carried out through the algorithms implemented in hardware and being applied to the preamplifier pulse outputs from the detectors, which is beyond the scope of the current thesis.

Chapter 5

Conclusions

The current thesis was aimed at investigating the different aspects of the digital signal processing (DSP) techniques being widely applied to the contemporary endeavours in the domain of nuclear physics experiments. To this purpose, various facets of the DSP technology has been introduced and discussed in the first chapter. The second chapter has been devoted to the discussion of a conventional nuclear spectroscopy experimental setup and associated instrumentation as well as replacing the same with the DSP systems. The merits and disadvantages of both the approaches have been elaborated. Different types of pulse shaping and the corresponding recursive algorithms have been discussed in chapter 3. This provides the theoretical background to the application of the digital pulse processing techniques on different types of pulses, as illustrated in chapter 4. The different pulses considered here included ideal step function as well as realistic pulses, simulated through appropriate analytical expression, expected to be obtained from scintillator and HPGe based detectors widely applied in nuclear spectroscopy. The effect of different choice of parameters in the recursive algorithm, on the resulting pulse shape was also tested for specific instances. The programming of the input pulses and implementation of the different types of shaping algorithms were carried out using MATLAB®. The programs have been detailed in the appendix.

5.1. Outlook.

In the light of the successful application of recursive algorithms to shaping of various pulses, in the current thesis, several projects in continuation to the present one can be planned. The actual test of the aforesaid recursive algorithms and choice of parameters would be implementing the different types of shaping in the hardware and applying the same on actual pulses from the detectors. The effect of various choices of parameters of the recursive algorithms on the energy and timing resolution of the detector can be probed in a subsequent endeavour. The current thesis provides a background to the development of an embedded system for pulse processing applications in nuclear spectroscopy experiments.

APPENDIX A-1

Application of CR filter on square wave pulse in MATLAB®.

```
clear all;
t=1:5000;
for n=1:5000

       if n>1000
        x(n)=100;
    else
           x(n)=0;
       end
       if n<1000
           x(n)=0;
       end
end
subplot(2,2,1);
plot(t,x);
tau=input('enter the time constant');
y=x(n)*exp(-t/tau);
subplot(2,2,2);
plot(t,y);
```

APPENDIX A-2

Application of RC filter on square wave pulse in MATLAB®.

```
clear all;
t=1:5000;
for n=1:5000

       if n>=1000
        x(n)=100;
    else
            x(n)=0;
       end
       if n<1000
            x(n)=0;
       end
end
subplot(2,2,1);
      plot(t,x);
      tau1=input('enter the time constant for rc circuit');
y=x(n)*(1-exp(-t/tau1));
subplot(2,2,2);
plot(t,y);
```

APPENDIX A-3

Application of CR-RC filter on square wave pulse in MATLAB®.

```
clear all;
for n=1:5000
            if n>1000
        x(n)=100;
            else
                x(n)=0;
    end
    if n<1000;
        x(n)=0;
    end
end
subplot(2,2,1);
plot(x,'m');
tau=input('please enter the value of time constant');
T=1/(75);%sampling time
k=exp(-T/tau);
z=k*(1+(T/tau));
for n=1:5000
    if n>0
        c1=x(n);
    else
        c1=0;
    end
    if n-1>0
        c2=2*k*y(n-1);
        c3=x(n-1);
    else
        c2=0;
        c3=0;
    end
    if n-2>0
        c4=((k)^2)*y(n-2);
    else
        c4=0;
    end
    y(n)=c1+c2-z*c3-c4;
end
subplot(2,2,2);
 plot(y);
```

APPENDIX A-4

Application of triangular filter on square wave pulse in MATLAB®.

```
clear all;
t=1:5000;
for n=1:5000
      if n>1000
    x(n)=100;
      else
        x(n)=0;
  end
  if n<1000;
    x(n)=0;
  end
end
subplot(2,2,1);
plot(t,x);
k=input('enter the value of rise time of output pulse');
for n=1:5000

  if (n-1>0)
    t1=y(n-1);

    t2=x(n);
  else
    t1=0;
    t2=0;
  end
  if (n-k>0)
    t3=2*x(n-k);
  else
    t3=0;
  end

  if (n-2*k>0)
    t4=x(n-2*k);
  else
    t4=0;
  end

    y(n)=t1+t2-t3-t4;

end

subplot(2,2,2)
plot(t,y)
```

APPENDIX A-5

Application of cusp filter on square wave pulse in MATLAB®.

```
clear all
t=1:5000;
for n=1:5000
   if n>1000
      x(n)=100;
   else
      x(n)=0;
   end
   if n<1000
      x(n)=0;
   end
end
subplot(2,2,1)
plot(t,x,'r')
k=input('enter the rise time of the output pulse');
m=input(' cusp lenght');
l=k+m;
for n=1:5000
   if n-1>0
      p1=p(n-1);
   else
      p1=0;
   end
   if n>0
      p2=x(n);
   else
      p2=0;
   end
   if n-k>0
      p3=x(n-k);
   else
      p3=0;
   end
   if n-l>0
      p4=x(n-l);
   else
      p4=0;
   end
   if n-l-k>0
      p5=x(n-l-k);
   else
      p5=0;
   end
   p(n)=p1+p2-p3+p4-p5;
end
for n=1:5000
   if n-1>0
      g1=g(n-1);
```

```
    else
        g1=0;
    end
    if n>0
        g2=p(n);
    else
        g2=0;
    end
    if n-k>0
        g3=x(n-k)*k;
    else
        g3=0;
    end
    if n-l>0
        g4=x(n-l)*k;
        g5=x(n-l);
    else
        g4=0;
        g5=0;
    end
    if n-l-k>0
        g6=x(n-l-k);
    else
        g6=0;
    end
    g(n)=g1+g2-g3-g4-g5+g6;
end
subplot(2,2,2)
plot(t,g);
```

APPENDIX A-6

Application of trapezoidal filter on square wave pulse in MATLAB®.

```
clear all;
t=1:5000;
for n=1:5000
            if n>1000
        x(n)=100;
            else
                x(n)=0;
    end
    if n<1000;
        x(n)=0;
    end
end
subplot(2,2,1)
plot(t,x);
title('step function of magnitude 100');
k=input('please input the rise time of the output pulse');
m=input('please input the value of flat top trapozoidal');
l=k+m;
%hold all;
for n=1:5000

    if (n-1>0)
        t1=y(n-1);
        t2=x(n);
    else
        t1=0;
        t2=0;
    end
    if (n-k>0)
        t3=x(n-k);
    else
        t3=0;
    end

   if (n-l>0)
       t4=x(n-l);
   else
       t4=0;

   end
   if (n-l-k>0)

       t5=x(n-l-k);
   else
       t5=0;
```

```
   end

       y(n)=t1+t2-t3-t4+t5;

end

subplot(2,2,2)
plot(t,y);
```

APPENDIX B-1

Generation of scintillator pulse in MATLAB®.

```
t=1:200;
A=10.530;
B=0.170;
c=0.331;
d=4.368;
e=3.482;
f=11.520;
x=(A*exp(-(t-c)/d)-A*exp(-(t-c)/e)+A*B*exp(-(t-c)/f));
subplot(2,2,1);
plot(t,x);
```

APPENDIX B-2

Application of trapezoidal filter on scintillator pulse in MATLAB®.

```
clear all
t=0:200;
A=10.530;
B=0.0170;
c=0.331;
d=4.368;
e=3.482;
f=11.520;
x=(A*exp(-(t-c)/d)-A*exp(-(t-c)/e)+A*B*exp(-(t-c)/f));
subplot(2,2,1);
plot(t,x);
k=input('enter rise time of the output pulse');
m=input('enter flat top duration of trapozoidal');
l=k+m;
M=input('please enter the desired value of M');
for n=1:201
    if n>0
        d1=x(n);
    else
        d1=0;
    end
        if n-k>0;
            d2=x(n-k);
        else
            d2=0;
        end
        if n-l>0
            d3=x(n-l);
        else
            d3=0;
        end
        if n-k-l>0
            d4=x(n-k-l);
        else
            d4=0;
        end
        D(n)=d1-d2-d3+d4;
end
for n=1:201
    if n-1>0
        p1=p(n-1);
```

```
    else
        p1=0;
    end
        if n>0
            p2=D(n);
        else
            p2=0;
        end
        p(n)=p1+p2;
end
for n=1:201
    if n>0
        r1=p(n);
        r2=M*D(n);
    else
        r1=0;
        r2=0;
    end
    r(n)=r1+r2;
end
for n=1:201
    if n>0
        s1=r(n);
    else
        s1=0;
    end
    if n-1>0
        s2=s(n-1);
    else
        s2=0;
    end
    s(n)=s1+s2;
end
subplot(2,2,2);
plot(s);
```

APPENDIX B-3

Application of CR-RC filter on the scintillator pulse.

```
t=1:100;
A=10.530;
B=0.170;
c=0.331;
d=4.368;
e=3.482;
f=11.520;
x=(A*exp(-(t-c)/d)-A*exp(-(t-c)/e)+A*B*exp(-(t-c)/f));
subplot(2,2,1);
plot(t,x);
tau=input('please enter the value of time constant');
T=1/(75);%sampling time
k=exp(-T/tau);
z=k*(1+(T/tau));
for n=1:100
    if n>0
        c1=x(n);
    else
        c1=0;
    end
    if n-1>0
        c2=2*k*y(n-1);
        c3=x(n-1);
    else
        c2=0;
        c3=0;
    end
    if n-2>0
        c4=((k)^2)*y(n-2);
    else
        c4=0;
    end
    y(n)=c1+c2-z*c3-c4;
end
subplot(2,2,2);
 plot(y);
```

APPENDIX C-1

Generation of HPGe pulse in MATLAB®.

```
clear all
t=1:5000;
TAU=input('enter the value of tau');
THETA=input('enter the value of theta');%finite rise time of the pulse
A=exp((-t)/TAU);
B=exp((-t)/THETA);
C=(TAU-THETA);
x=(1/C)*(A-B);
subplot(2,2,1);
plot(t,x,'r');
```

APPENDIX C-2

Application of CR-RC filter on HPGe pulse in MATLAB®.

```
clear all;
t=1:5000;
TAU=input('enter the value of tau');
THETA=input('enter the value of theta');%finite rise time of the pulse
A=exp((-t)/TAU);
B=exp((-t)/THETA);
C=(TAU-THETA);
x=(1/C)*(A-B);
subplot(2,2,1)
plot(t,x,'r')
tau1=input('please enter the value of time constant for plot 1');
T1=1/(75);%sampling time
k1=exp(-T1/tau1);
z1=k1*(1+(T1/tau1));
for n=1:5000
    if n>0
        c1=x(n);
    else
        c1=0;
    end
    if n-1>0
        c2=2*k1*y(n-1);
        c3=x(n-1);
    else
        c2=0;
        c3=0;
    end
    if n-2>0
        c4=((k1)^2)*y(n-2);
    else
        c4=0;
    end
    y(n)=c1+c2-z1*c3-c4;
end
 tau2=input('please enter the value of time constant for plot2');
```

```
T2=1/(75);%sampling time
k2=exp(-T2/tau2);
z2=k2*(1+(T2/tau2));
for n=1:5000
    if n>0
        d1=x(n);
    else
        d1=0;
    end
    if n-1>0
        d2=2*k2*y1(n-1);
        d3=x(n-1);
    else
        d2=0;
        d3=0;
    end
    if n-2>0
        d4=((k2)^2)*y1(n-2);
    else
        d4=0;
    end
    y1(n)=d1+d2-z2*d3-d4;
end
 tau3=input('please enter the value of time constant for plot 3');
T3=1/(75);%sampling time
k3=exp(-T3/tau3);
z3=k3*(1+(T3/tau3));
for n=1:5000
    if n>0
        e1=x(n);
    else
        e1=0;
    end
    if n-1>0
        e2=2*k3*y2(n-1);
        e3=x(n-1);
    else
        e2=0;
        e3=0;
    end
    if n-2>0
        e4=((k3)^2)*y2(n-2);
    else
        e4=0;
    end
    y2(n)=e1+e2-z3*e3-e4;
end
 subplot(2,2,2)
 plot(t,y,t,y1,t,y2);
```

APPENDIX C-3

Application of trapezoidal filter on HPGe pulse in MATLAB®.

```
clear all
t=1:5000;
TAU=input('enter the value of tau');
THETA=input('enter the value of theta');%finite rise time of the pulse
k=input('enter rise time of the output pulse');
m=input('enter flat top duration of trapezoidal');
l=k+m;
A=exp((-t)/TAU);
B=exp((-t)/THETA);
C=(TAU-THETA);
x=(1/C)*(A-B);
subplot(2,2,1);
plot(t,x,'r');
M=input('please enter the desired value of M');
for n=1:5000
    if n>0
        d1=x(n);
    else
        d1=0;
    end
        if n-k>0;
            d2=x(n-k);
        else
            d2=0;
        end
        if n-l>0
            d3=x(n-l);
        else
            d3=0;
        end
        if n-k-l>0
            d4=x(n-k-l);
        else
```

```
            d4=0;
        end
        d(n)=d1-d2-d3+d4;
end
for n=1:5000
    if n-1>0
        p1=p(n-1);
    else
        p1=0;
    end
        if n>0
            p2=d(n);
        else
            p2=0;
        end
        p(n)=p1+p2;
end
for n=1:5000
    if n>0
        r1=p(n);
        r2=M*d(n);
    else
        r1=0;
        r2=0;
    end
    r(n)=r1+r2;
end
for n=1:5000
    if n>0
        s1=r(n);
    else
        s1=0;
    end
    if n-1>0
        s2=s(n-1);
    else
        s2=0;
    end
    s(n)=s1+s2;
end
subplot(2,2,2);
plot(t,s);
```

APPENDIX C-4

```
clear all;
t=1:6000;
TAU=input('enter the value of tau');
THETA=input('enter the value of theta');%finite rise time of the pulse
k=input('enter rise time for the output pulse 1');
m=input('enter flat top duration of trapozoidal 1');
l=k+m;
A=exp((-t)/TAU);
B=exp((-t)/THETA);
C=(TAU-THETA);
x=(l/C)*(A-B);

TAU1=input('enter the value of tau2');
THETA1=input('enter the value of theta2');%finite rise time of the pulse
k1=input('enter rise time for the output pulse 2');
m1=input('enter flat top duration of trapozoidal 2');
l1=k1+m1;
A1=exp((-t)/TAU1);
B1=exp((-t)/THETA1);
C1=(TAU1-THETA1);
x1=(l/C1)*(A1-B1);

TAU2=input('enter the value of tau');
THETA2=input('enter the value of theta');%finite rise time of the pulse
k2=input('enter rise time for the output pulse 3');
m2=input('enter flat top duration of trapozoidal 3');
l2=k2+m2;
A2=exp((-t)/TAU2);
B2=exp((-t)/THETA2);
C2=(TAU2-THETA2);
x2=(l/C2)*(A2-B2);
```

```
M=input('please enter the desired value of M');
for n=1:6000
    if n>0
        d1=x(n);
    else
        d1=0;
    end
        if n-k>0;
            d2=x(n-k);
        else
            d2=0;
        end
        if n-l>0
            d3=x(n-l);
        else
            d3=0;
        end
        if n-k-l>0
            d4=x(n-k-l);
        else
            d4=0;
        end
        d(n)=d1-d2-d3+d4;
end
for n=1:6000
    if n-1>0
        p1=p(n-1);
    else
        p1=0;
    end
        if n>0
            p2=d(n);
        else
            p2=0;
        end
        p(n)=p1+p2;
end
for n=1:6000
    if n>0
        r1=p(n);
        r2=M*d(n);
    else
        r1=0;
        r2=0;
    end
    r(n)=r1+r2;
end
for n=1:6000
    if n>0
        s1=r(n);
    else
        s1=0;
    end
    if n-1>0
        s2=s(n-1);
```

```
    else
        s2=0;
    end
    s(n)=s1+s2;
end

for n=1:6000
    if n>0
        v1=x1(n);
    else
        v1=0;
    end
        if n-k1>0;
            v2=x1(n-k1);
        else
            v2=0;
        end
        if n-l1>0
            v3=x1(n-l1);
        else
            v3=0;
        end
        if n-k1-l1>0
            v4=x1(n-k1-l1);
        else
            v4=0;
        end
        v(n)=v1-v2-v3+v4;
end
for n=1:6000
    if n-1>0
        q1=q(n-1);
    else
        q1=0;
    end
        if n>0
            q2=v(n);
        else
            q2=0;
        end
        q(n)=q1+q2;
end
for n=1:6000
    if n>0
        b1=q(n);
        b2=M*v(n);
    else
        b1=0;
        b2=0;
    end
    b(n)=b1+b2;
end
for n=1:6000
    if n>0
        u1=b(n);
```

```
    else
        u1=0;
    end
    if n-1>0
        u2=u(n-1);
    else
        u2=0;
    end
    u(n)=u1+u2;
end

for n=1:6000
    if n>0
        a1=x2(n);
    else
        a1=0;
    end
        if n-k2>0;
            a2=x2(n-k2);
        else
            a2=0;
        end
        if n-l2>0
            a3=x2(n-l2);
        else
            a3=0;
        end
        if n-k2-l2>0
            a4=x2(n-k2-l2);
        else
            a4=0;
        end
        a(n)=a1-a2-a3+a4;
end
for n=1:6000
    if n-1>0
        Y1=Y(n-1);
    else
        Y1=0;
    end
        if n>0
            Y2=a(n);
        else
            Y2=0;
        end
        Y(n)=Y1+Y2;
end
for n=1:6000
    if n>0
        N1=Y(n);
        N2=M*a(n);
    else
        N1=0;
        N2=0;
    end
```

```
    N(n)=N1+N2;
end
for n=1:6000
    if n>0
        S1=N(n);
    else
        S1=0;
    end
    if n-1>0
        S2=S(n-1);
    else
        S2=0;
    end
    S(n)=S1+S2;
end
subplot(2,2,2);
plot(t,s,t,u,t,S);
```

References

[1] P. Ramesh Babu, “*Digital Signal Processing*” fourth edition, SciTech Publications (India) Pvt. Ltd. (2008).

[2] The Ohio State University Gateway Engineering Education Coalition.

[3] S.W. Smith, “The Scientist and Engineer Guide to Digital Signal Processing”, 2nd Edition, *California Technical Publishing*, San Diego, CA, 1999.

[4] B.Paillard, ”*An Introduction to Digital Signal Processors*” *Génie électrique et informatique Université de Sherbrooke*, January 27 2002

[5] J. F. Loude, , “*Energy Resolution in Nuclear Spectroscopy*”, *LPHE notes and Conferences*, IPHE 2000-22, Institute de Physique des Hautes Energies, Lou sane, Switzerland, (2000).

[6] IAEA-TECDOC-1634

[7] Jordanov V. T. et al., “Digital techniques for real-time pulse shaping in radiation Measurements”, *Nucl. Instr. Meth*. A353 (1994) 261-264.

[8] http://radware.phy.ornl.gov/dsp_work

[9] http://docs.nscl.msu.edu/daq/samples/Standards.pdf

[10] T. Hoagland, “A brief discussion of NIM”,*CAMAC*,VME standards, December 2,2004

[11] COLMEK,VME Chassis

[12] WIENER, PLEIN&BAUS, Corporation

[13] NIM 8301 7U 12 slot smart fan unit 300/600 W crate, CAEN document

[14] P.A.B Scoullar & R.J Evans, "High throughput digital pulse processing hardware", *European conference on X-RAY spectrometry*, 16th-20th June 2008, Cavtat, Dubrovnik, Croatia.

[15] A. Fallu-Labruyer *et al.*, Time resolution studies using digital constant Fraction discrimination, *Nucl. Instr. and Meth. in Phys. Res A579* (2007) 247.

[16] "Digital Pulse Processing in Nuclear Physics", CAEN, revision 1.6.

[17] Jordanov *et al.*, *Nuclear Instruments and Methods in Physics Research A353 (1994)261-264.*

[18] G.F. Knoll, "Radiation detection and measurement", *third edition*, Wiley, new York 2000.

[19] http://www.dnp.fmph.uniba.sk/~kollar/je_w/el2.htm

[20] M. Nakhostin, "Recursive Algorithms for Real-time Digital CR-$(RC)^n$ Pulse Shaping", *IEEE TRANSACTIONS ON NUCLEAR SCIENCE*, VOL 58, NO. 5, OCTOBER 2011.

[21] V.T Jordanov, G.F Knoll, *Nucl. Instr. and Meth. in Phys. Res A345 (1994)337-345*

[22] S. Marrone *et al*, *Nucl. Instr. and Meth. in Phys. Res A 490 (2002)299.*

[23] M.D. Aspinall, R.O. Mackin *et al.*, *Nucl. Instr. and Meth. in Phys. Res A 578 (2007)261*.

[24] G. Liu et al., *Nucl. Instr. and Meth. in Phys. Res A 607 (2009)620-628.*