DESIGN OF BANDPASS FIR DIGITAL FILTER

Oleh

Sheila Rita Sahamir

Disertasi ini dikemukakan kepada
UNIVERSITI SAINS MALAYSIA

Sebagai memenuhi sebahagian daripada syarat keperluan
untuk ijazah dengan kepujian

SARJANA MUDA KEJURUTERAAN (KEJURUTERAAN ELEKTRONIK)

Pusat Pengajian Kejuruteraan
Elektrik Dan Elektronik
Universiti Sains Malaysia

Mac 2005

ABSTRACT

This work is concerned with the design of digital bandpass FIR filter using three differents method; Window, Optimal and Frequency sampling. Besides, the filter will be designed using MATLAB Version 6.5.1 that available in the School of Electrical and Electronic, Engineering. The project then will compare the merit and de-merit of each method. The project also has a simple step-by-step guide for designing bandpass FIR digital filter. From the given specification, we will calculate the appropriate parameters that will be used in MATLAB programming. Basically, there are five steps involve in designing digital filter such as specification of the filter requirements, calculation of suitable filter coefficients, performs the suitable realization structure for the filter, analysis the effects of finite wordlength on the filter performance and implementation in software and/ or hardware. It has been found that the Optimal method of this design of FIR digital filter provides the best result as it produces filter which meet the specification with the least number of coefficient. The suitable filter realization will be used is transversal filter that is the most widely using in implementation of the digital filter.

ABSTRAK

Projek yang dijalankan ini melibatkan rekabentuk bagi penuras laluan jalur untuk penuras jenis FIR digital dengan menggunakan tiga kaedah yang berbeza iaitu; *Window*, *Optimal* dan *Frequency sampling*. Penuras akan direkabentuk dengan menggunakan aturcara MATLAB Versi 6.5.1 yang boleh didapati di dalam makmal Kejuruteraan Elektrik dan Elektronik. Projek yang dijalankan juga akan membandingkan kebaikan dan keburukan di antara ketiga-tiga kaedah yang digunakan. Projek yang dijalankan untuk merekabentuk penuras laluan jalur bagi penuras jenis FIR digital ini mempunyai panduan dari langkah demi langkah yang mudah untuk merekanya. Daripada spesifikasi yang diberikan, kita perlu mengira parameter-parameter berkaitan yang akan digunakan ketika penyelakuan menggunakan aturcara MATLAB. Secara asasnya, terdapat lima langkah yang terlibat dalam merekabentuk penuras digital seperti spesifikasi bagi keperluan sesuatu penuras, pengiraan pemalar penuras yang sesuai, struktur penuras sebelum diimplementasikan ke dalam pemproses isyarat digital (DSP), analisa kesan bagi panjang pemalar penuras yang dipilih ke atas pemproses yang digunakan dan langkah terakhir ialah mengimplementasikan rekabentuk ke dalam bentuk aturcara komputer dan/ atau alatan. Hasil daripada rekabentuk yang dijalankan menunjukkan kaedah *Optimal* memberikan keputusan yang terbaik dengan menghasilkan penuras digital yang memenuhi kehendak spesifikasi yang diberikan dengan bilangan pemalar yang

paling rendah. Struktur penuras yang sesuai digunakan untuk pelaksanaan ialah struktur *transversal* yang digunakan secara meluas untuk diimplementasikan ke dalam pemproses isyarat digital.

ACKNOWLEDGEMENT

Bismillahirrahmanirrahim…

By the name of Allah, The Most Merciful, who is still listening to all prayers from His humble servants.

I would like first and foremost to express our boundless gratitude to Allah, for without His will I could have never completed this dissertation in time.

I would also like to thank my dissertation lecturer coordinator, Professor Dr. Farid Ghani (also as my supervisor) for his compassion and dedication in his mission to guide me, with not just a basic knowledge of this final year project, but a solid and comprehensive understanding of it. I could not be more thankful than that.

I would also like to take this chance to extent my thankfulness to the second examiner who will co-operate with my supervisor in evaluating my final year project report and presentation session.

Not forgetting, my other School of Electrical and Electronic lecturers who are no less supportive and helpful in giving advices, particular information and guidance whenever I need them.

Last but never the least; I would like to thank my family and friends for their endless love and support, encouragement and helping hands. Thank you and May Allah bless you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

PREFACE

Much has happened from the very beginning this project running out. The main objective to present this final year project is to fulfill the condition of graduation for postgraduate student. Besides, this is a challenge for students to apply their knowledge during study at university for four years.

This dissertation include the understanding of digital signal processing which is applied the digital filter as a part of Digital Signal Processing applications.

A digital filter is built by converting the analog signals to digital values, and then implementing the filter using digital circuits, such as delay lines, accumulators, and feedback elements for the purpose of achieving a filtering objective.

The term digital filter refers to the specific hardware or software routine that performs the filtering algorithm. There are two types of digital filter, namely infinite impulse response (IIR) and finite impulse response (FIR).

Report for the final year project will cover up the design of bandpass FIR digital filter step by step. This report is divided into six chapters. Chapter 1, 2 and 3 are very important due to explanation of digital filter theoretical. The following chapter represents the design of FIR digital filter according to the given specification. It also was including the comparison of three methods that was used to design the filters by using MATLAB.

CHAPTER 1

INTRODUCTION

1.1 BACK GROUND

DSP, or Digital Signal Processing, as the term suggests, is the processing of signals by digital means. A signal in this context can mean a number of different things. Historically the origins of signal processing are in electrical engineering, and a signal here means an electrical signal carried by a wire or telephone line, or perhaps by a radio wave. More generally, however, a signal is a stream of information representing anything from stock prices to data from a remote-sensing satellite.

In many cases, the signal is initially in the form of an analog electrical voltage or current, produced for example by a microphone or some other type of transducer. In some situations the data is already in digital form- such as the output from the readout system of a CD (compact disc) player. An analog signal must be converted into digital (i.e. numerical) form before DSP techniques can be applied. An analog electrical voltage signal, for example, can be digitized using an integrated electronic circuit (IC) device called an analog-to-digital converter or ADC. This generates a digital output in the form of a binary number whose vale represents the electrical voltage input to the device.

The development of digital signal processing dates from the 1960's with the use of mainframe digital computers for number-crunching applications such as the Fast Fourier Transform (FTT), which allows the frequency spectrum of a signal to be computed rapidly. These techniques were not widely used at that time, because suitable computing equipment was available only in universities and other scientific research institutions.

Introduction of microprocessors in 1970's and 80's resulted in a wide range of applications of DSP. With increasing the use of DSP techniques in industry and elsewhere, special purpose microprocessors with architectures designed specifically for applications in the field of DSP were introduced. Such microprocessors were termed as Digital Signal Processors (DSPs). DSPs are programmable devices and are capable of carrying out millions of instruction per second making it possible for use in real time applications.

DSP technology nowadays finds application in almost all spheres of life. Some of the more important areas are;

- Telephony (particularly mobile phones)
- IC Technology
- Medicine
- Multimedia (Image and Speech Processing)
- Data Compression
- Entertainment Electronics
- Instrumentation

Table 1.1 shows the typical applications of DSP.

General-purpose DSP	Graphics/ Imaging	Instrumentation
Digital filtering Convolution Correlation Hilbert transforma Adaptive filtering Windowing Waveform generation	3-D rotation Robot vision Image transmission/ compression Pattern rcognition Image enhancement Homomorphics processing Workstations Animation/ digital map	Spectrum analysis Function generation Pattern matching Seismic processing Transient analysis Digital filtering Phase-locked loops
Voice/ speech	Control	Military
Voice main Speech recording Speech recognition Speaker verification	Disk control Servo control Robot control Laser printer control	Secure communications Radar processing Sonar processing Image processing

Speech enhancement Speech synthesis Text to speech	Engine control Motor control	Navigation Misile guidance Radio frequency modems
Telecommunications		Automotive
Echo cancellation ADPCM transcoders Digital PBXs Line repeaters Channel multiplexing 1200 to 19,200-bps modems adaptive equalizers DTMF encoding/ decoding Data encryption	FAX Cellular telephone Speaker phones Digital speech Interpolation (DSI) X.25 packet switching Video conferencing Spread spectrum communications	Engine control Vibration analysis Antiskid brakes Adaptive ride control Global positioning Navigation Voice commands Digital radio Cellular telephones
Consumer	Industrial	Medical
Radar detectors Power tools Digital audio/ TV Music synthesizer Educational toys	Robotics Numeric control Security access Power line monitors	Hearing aids Patient monitoring Ultrasound equipment Diagnosis tools Prosthetics Fetal monitors

Table 1.1: Typical Applications of DSP

Signals commonly need to be processed in a variety of ways. For example, the output signal from a transducer may well be contaminated with unwanted electrical ‘noise’’. The electrodes attached to a patient’s chest when an ECG is taken measure tiny electrical voltage changes due to the activity of the heart and other muscles. The signal is often strongly affected by “mains pickup” due to electrical interference from the mains supply. Processing the signal using a filter circuit can remove or at least reduce the unwanted part of the signal. Increasingly nowadays the filtering of signals to improve signal quality or to extract important information is done by DSP techniques rather than by analog electronics.

Because of its great importance in communication, control, and practically all electronic systems, electronic filter design has been developed to a very advanced state. First, analog filters, both passive and active, were perfected. Then, their discrete-time equivalents were derived. Now design of digital filters is often carried out completely independently of analog system.

A digital filter, as has been explained in Chapter 2, is a mathematical algorithm implemented in hardware and/ or software that operates on a digital input signal to produce a digital output signal for the purpose of achieving a filtering objective. The term digital filter refers to the specific hardware or software routine that performs the filtering algorithm. Digital filters often operate to digitized analog signals or just numbers, representing some variable, stored in a computer memory.

A simplified block diagram of a real-time digital filter, with analog input and output signals is given in Figure 1.1. The bandlimited analog signal is sampled periodically and converted into a series of digital samples, x(n), n= 0, 1, … , N-1. The digital processor implements the filtering operation, mapping the sequences, x(n), into the output sequence, y(n), in accordance with a computational algorithm for the filter. The DAC converts the digitally filtered output into analog values which are then analog filtered to smooth and remove unwanted high frequency components.

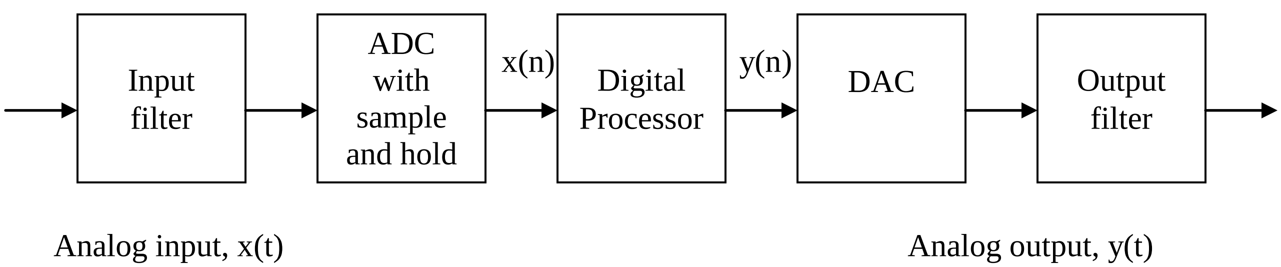

Figure 1.1: A simplified block diagram of a real-time digital filter with analog input and output signals.

Digital filters play very important roles in DSP. Compared with analog filters they are preferred in a number of applications. As an example in data compression, biomedical signal processing, speech processing, image processing, data transmission, digital audio, and telephone echo cancellation because their inherent advantages over the analog filters. However, their main disadvantage is due to the errors introduced because of the finite word length effects where the filters are subject to ADC noise resulting from quantizing a continuous signal, and to round-off noise incurred during computation. With higher order recursive filters, the accumulation of round-off noise could lead to instability.

The design of a digital filter involves the following five steps:

1) Specification of the filter requirement
2) Calculation of suitable filter coefficients
3) Representation of the filter by a suitable structure (realization)
4) Analysis of the effects of finite word length on filter performance
5) Implementation of filter in software and/ or hardware

The five steps are not necessarily independent: nor they are always performed in the order given. In fact, techniques are now available that combine the second and aspects of the third and fourth steps. However, the approach involving the above steps gives a simple step-by-step guide that will ensure a successful design. To arrive at an efficient filter, it may be necessary to iterate a few times between the steps, especially if the problem specification is not watertight, as is often the case, or if the designer wants to explore other possible designs.

1.2 SCOPE OF THE PROJECT

This project is concerned with the design of a bandpass FIR digital filter suitable for use as a noise limiting filter for multirate system such as high quality data acquisition and compact disc player. The filter is to be designed using the following three methods:

i. The window method
ii. The frequency sampling method
iii. The optimal method

The filter is required to have the following specifications:

Pass band	8-12 kHz
Stop band ripple	0.001
Peak pass band ripple	0.01
Sampling frequency	44.14 kHz
Transition width	3 kHz

It is also required to compare these merits and de-merits of each design method. In the Chapter 2 we are going to discuss about the most significant component in digital filter. The representation of the difference equation, transfer function, impulse response, frequency response, step response and stability of digital filter will be discussed later.

CHAPTER 2

DIGITAL FILTERS

2.1 INTRODUCTION

A digital filter is just a filter that operates on digital signals. It is important to realize that a digital filter can do anything that a real-world filter can do. In signal processing, the function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range.

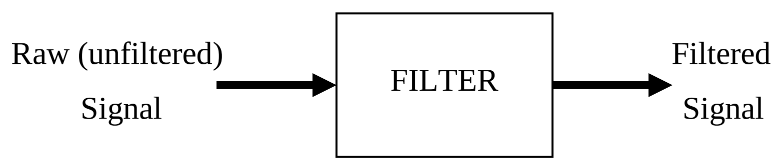

Figure 2.1: Block diagram illustrates the basic idea.

There are two main kinds of filter, analog and digital. They are quite different in their physical makeup and in how they work. An analog filter uses analog electronic circuits made up from components such as resistors, capacitors and op amps to produce the required filtering effect. Such filter circuits are widely used in such applications as noise reduction, video signal enhancement, graphic equalizers in hi-fi systems, and many other areas.

There are well-established standard techniques for designing an analog filter circuit for a given requirement. At all stages, the signal being filtered is an electrical voltage or current which is the direct analogue of the physical quantity (e.g. a sound or video signal or transducer output) involved.

A digital filter uses a digital processor to perform numerical calculations on sampled values of the signal. The processor may be a general-purpose computer such as a PC, or a specialized DSP (Digital Signal Processor) chip.

The analog input signal must first be sampled and digitized using an ADC (analog to digital converter). The resulting binary numbers, representing successive sampled values of the input signal, are transferred to the processor, which carries out numerical calculations on them. These calculations typically involve multiplying the input values by constants and adding the products together. If necessary, the results of these calculations, which now represent sampled values of the filtered signal, are output through a DAC (digital to analog converter) to convert the signal back to analog form.

Note that in a digital filter, the signal is represented by a sequence of numbers, rather than a voltage or current.

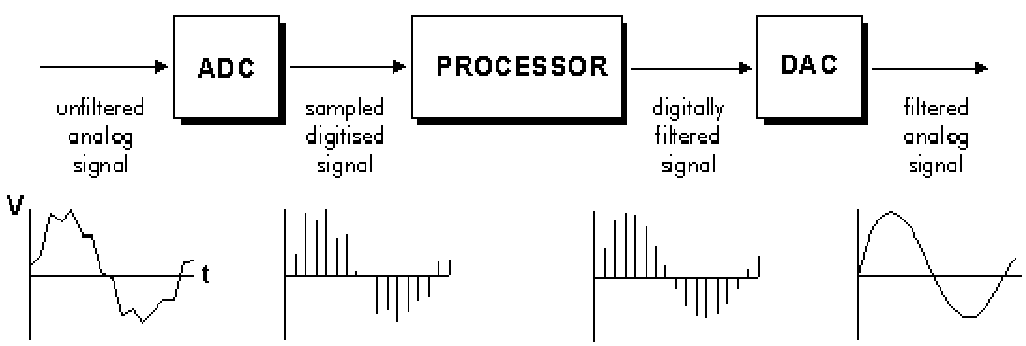

Figure 2.2: The diagram shows the basic setup of such a system.

There are many comparisons between digital filters and analog filters. Digital filter can have characteristics which are not possible with analog filters, such as a truly linear phase response. Unlike analog filters, the performance of digital filters does not vary with environment changes, for example thermal variations. This eliminates the need to calibrate periodically. The frequency response of a digital filter can be automatically adjusted if it is implemented using a programmable processor, which is why they are widely used in adaptive filters. Several input signals or channels can be filtered by one digital filter without the need to replicate the hardware.

Both filtered and unfiltered data can be saved for further use. Advantage can be readily taken of the tremendous advancements in VLSI technology to fabricate digital filters and to make them small in size, to consume low power, and to keep the cost down. In practice, the precision achievable with analog filters is restricted; for example, typically a maximum of only about 60 to 70 dB stopband attenuation is possible with active filters designed with off- the- shelf components. With digital filters the precision is limited only by the wordlength used. The performance of digital filters is repeatable from unit to unit. Digital filter can be used at very low frequencies, found in many biomedical applications for example, where the use of analog filters is impractical. Also, digital filters can be made to work over a wide range of frequencies by a mere change to the sampling frequency.

The disadvantages of digital filters compared with analog filters are speed limitation that is the maximum bandwidth of signals that digital filters can handle, in real-time, is much lower than for analog filters. The speed of operation of a digital filter depends on the speed of the digital processor used and on the number of arithmetic operations that must be performed for the filtering algorithm, which increases as the filter response is made tighter. The other advantage is finite wordlength effects which the digital filters are subject to ADC noise resulting from quantizing a continuous signal, and to roundoff noise incurred during computation. With higher order recursive filters, the accumulation of roundoff noise could lead to instability. The long design and development times for digital filters, especially hardware development, can be much longer than for the analog filters. However, once developed the hardware or software can be used for others filtering or DSP tasks with little or no modifications.

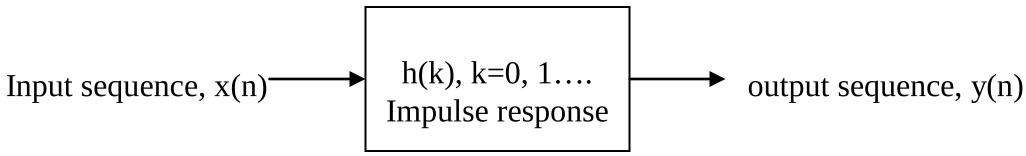

Figure 2.3: A conceptual representation of a digital filter

Digital filters are broadly divided into two classes, namely infinite impulse response (IIR) and finite impulse response (FIR) filters. Either type of filter, in its basic form, can be represented by its impulse response sequence, h(k), k= 0, 1, … as shown in Figure 2.3

2.2 **MATHEMATICAL MODELLING OF DIGITAL FILTERS**

The design process begins with the filter specifications, which may include constraints on the magnitude and/or phase of the frequency response, constraints on the impulse response or step response of the filter, specifications of the type of filter, as an example FIR or IIR, and the filter order.

2.2.1 DIFFERENCE EQUATION REPRESENTATION

The different equation specifies the actual operations that must be performed by the discrete-time system on the input data, in the time domain, in order to generate the desired output. Two types of digital filters consist of finite impulse response, FIR and infinite impulse response, IIR. The input and output signals to the filter are related by the convolution sum, which is given by the following equations 2.2.1 for IIR and 2.2.2 for FIR:

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \qquad (2.2.1)$$

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \qquad (2.2.2)$$

It is evident from these equation that, for IIR filters, the impulse response is of infinite duration whereas for FIR it is of finite duration, since h(k) for the FIR has only N values. In practice, it is not feasible to compute the output of the IIR filter using equation 2.2.1 because the length of its impulse response is too long (infinite in theory). Instead, the IIR filtering equation is expressed in a recursive form:

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^{N} b_k x(n-k) - \sum_{k=1}^{M} a_k y(n-k) \qquad (2.2.3)$$

Where the a_k and b_k are the coefficients of the filter. Thus, equation 2.2.2 and 2.2.3 are the different equations for the FIR and IIR filters respectively. These equations, and in particular the values of h(k), for FIR, or a_k and b_k for IIR are often very important objectives of most filter design problems. We note that, in equation 2.2.3, the current output sample, y(n), is a function only of past and present values of the input. Note, however, that when the b_k are set to zero, equation 2.2.3 reduces to the FIR equation 2.2.2

2.2.2 TRANSFER FUNCTION REPRESENTATION

Alternative representation for FIR and IIR filters are shown by the following equations that has a transfer function with a polynomial in z^{-k}. These are the transfer functions for these filters and are very useful in evaluating their frequency responses.

$$H(z) = \sum_{k=0}^{N-1} h(k) z^{-k} \qquad (2.2.2.1)$$

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}} \qquad (2.2.2.2)$$

2.2.3 IMPULSE RESPONSE

In the design digital systems the need often arises to obtain values of impulse responses. For example, in FIR system design the impulse response is required to implement the system, and in IIR system design the values are required for stability analysis. The impulse response may also used to evaluate the frequency response of the system.

The impulse response of a digital system may be defined as the inverse z-transform of the system's transfer function, H(z):

$h(k) = Z^{-1}[\ H(z)\ ],\ k=0,1,\ldots$ (2.2.3.1)

If the z-transform, H(z), is available as a power series, that is

$$H(z) = \sum_{k=0}^{\infty} h(k) z^{-k}$$

$$= h(0) + h(1)z^{-1} + h(2)z^{-2} + \ldots \qquad (2.2.3.2)$$

The coefficient of the z-transform gives directly the impulse response.

For IIR digital filters, H(z) is often expressed as a ratio of polynomials.

The impulse response may also be viewed as the response of a digital system to a unit impulse, u(n), which has a value of 1 at n=0 and a value of 0 at all other values of n. This view arises from the fact that if we make the input to the system equal to h(n), the system's impulse response:

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^{\infty} h(k)u(n-k)$$

$$= h(0)u(n) + h(1)u(n-1) + h(2)u(n-2) + \ldots$$

$$= h(n),\ n=0,\ 1,\ \ldots \qquad (2.2.3.3)$$

This provide a simple alternative method of computing h(k) (indeed it provides another method of obtaining the inverse z-transform).

2.2.4 FREQUENCY RESPONSE

There are many instances when it is necessary to evaluate the frequency response of digital filters. For example, in the design of digital filters, it is often necessary to examine the spectrum of the filter to ensure that the desired specifications are satisfied. The frequency response of a system can be readily obtained from its z-transform.

If we set z=$e^{j\omega T}$, that is evaluate the z-transform around the unit circle, we apply the Fourier Transform in the equation 2.2.2.1:

$$H(z) = \sum_{k=0}^{N-1} h(k)\, z^{-k} \quad |_{z = e^{-jnwT}} \qquad (2.2.4.1)$$

$$H(e^{j\omega T}) = \sum_{k=0}^{N-1} h(k)\, e^{-jn\omega T} \qquad (2.2.4.2)$$

$H(e^{j\omega T})$ is referred to as the frequency response of the system. We have used the symbol T to emphasize the dependence of the frequency response of digital filter on the sampling frequency. In general, $H(e^{j\omega T})$ is complex. Its modulus gives the magnitude response and its phase the phase response of the system.

2.3 POLE-ZERO PLOT AND STABILITY OF DIGITAL FILTER

In most practical discrete-time systems the z-transform, that is the system transfer function, H(z), can be expressed in terms of its poles and zeros. Consider, for example, the following z-transform representing a general, Nth-order discrete-time filter (where N=M):

$$H(z) = \frac{N(z)}{D(z)} \qquad (2.3.1)$$

Where

$$N(z) = b_o z^N + b_1 z^{N-1} + b_2 z^{N-2} + \ldots + b_N$$

$$D(z) = a_o z^N + a_1 z^{N-1} + a_2 z^{N-2} + \ldots + a_N$$

The a_k and b_k are the coefficients of the filter. If H(z) has poles at $z=p_1, p_2, \ldots, p_N$ and zeros at $z=z_1, z_2, \ldots, z_N$, then H(z) can be factored and represented as

$$H(z) = \frac{K(z-z_1)(z-z_2)\ldots(z-z_N)}{(z-p_1)(z-p_2)\ldots(z-p_N)} \tag{2.3.2}$$

Where z_i is the ith zero, p_i is the ith pole and K is the gain factor. The poles of a z-transform such as H(z) are the values of z for which H(z) becomes infinity. The values of z for which H(z) becomes zero are referred to as zeros. The poles and zeros of H(z) may be real or complex. When they are complex, they occur in conjugate pairs, to ensure that the coefficients, a_k and b_k are real. It should be clear from equation 2.3.2 that if the locations of poles and zeros of H(z) are known, then H(z) itself can be readily reconstructed to within a constant.

The information contained in the z-transform can be conveniently displayed as a pole-zero diagrams.

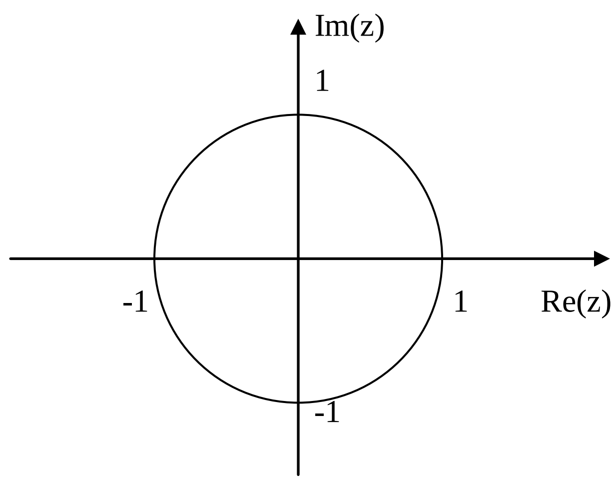

Figure 2.4: z-transform in the form of a pole-zero diagrams

The pole-zero diagram provides an insight into the properties of a given discrete-time system. For example, from the locations of poles and zeros can infer the frequency response of the system as well as its degree of stability. For a stable system, all poles must lie inside the unit circle (or be coincident with zeros on the unit circle) like in Figure 2.3

2.4 FIR DIGITAL FILTER

Finite digital filter, FIR, is the operation of which is governed by linear constant-coefficient different equations of a nonrecursive nature. FIR filters exhibit three important properties:

i. They have finite memory and, therefore, any transient start-up is of limited duration
ii. They are always BIBO stable
iii. They can realize a desired magnitude response with exactly linear phase response such as no phase distortion.

The basic FIR filter is characterized by the following two equations:

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad (2.4.1)$$

$$H(z) = \sum_{k=0}^{N-1} h(k)z^{-k} \quad (2.4.2)$$

Where h(k), k=0, 1, …, N-1, are the impulse response coefficients of the filter, H(z) is the transfer function of the filter and N is the filter length, that is the number of filter coefficients. y(n) is the FIR different equation. It is a time domain equation and describes the FIR filter in its non-recursive form. When FIR filters are implemented in this form that is by direct evaluation of y(n), they are always stable. FIR filters are very simple to implement. All DSP processors available have architectures that are suited to FIR filtering. Non-recursive FIR filters suffer less from the effects of finite wordlength than IIR filters. Recursive FIR filters also exist and may offer significant computational advantages.

2.4.1 LINEAR PHASE FILTER

The ability to have an exactly linear phase response is one of the most important properties of FIR filters. When a signal passes through a filter, it is modified in amplitude and phase. The nature and extent of the modification of the signal is dependent on the amplitude and phase characteristics of the filter. The phase delay or group delay of the filter provides a useful measure of how the filter modifies the phase characteristics of the signal.

If we consider a signal that consists of several frequency components such as a speech waveform or a modulated signal, the phase delay of the filter is amount of time delay each frequency component of the signal suffers in going through the filter. The group delay on the other hand is the average time delay the composite signal suffers at each frequency. Mathematically, the phase delay is the negative of the phase angle divided by frequency whereas the group delay is the negative of the derivative of the phase with respect to frequency:

$$T_p = -\theta(\omega)/\omega \qquad (2.4.1.1)$$

$$T_g = -d\theta(\omega)/d\omega \qquad (2.4.1.2)$$

A filter with a nonlinear phase characteristic will cause a phase distortion in the signal that passes through it. This is because the frequency components in the signal will each be delayed by an amount not proportional to frequency thereby altering their harmonic relationships. Such a distortion is undesirable in many applications, for example music, data transmission, video, and biomedicine, and can be avoided by using filters with linear phase characteristics over the frequency bands of interest.

A filter is said to have a linear phase response if its phase response satisfies one of the following relationships:

$$\theta(\omega) = -\alpha\omega \tag{2.4.1.3}$$

$$\theta(\omega) = \beta - \alpha\omega \tag{2.4.1.4}$$

Where α and β are constant. If a filter satisfies the condition given in the above equation, it will have both constant group and constant phase delay responses.

2.4.2 FILTER STRUCTURE

A causal FIR filter has a system function that is a polynomial in z^{-1}:

$$H(z) = \sum_{k=0}^{N-1} h(k) z^{-k} \tag{2.4.2.1}$$

For an input x(n), the output is

$$y(n) = \sum_{k=0}^{N-1} h(k) x(n-k) \tag{2.4.2.2}$$

Realization involves converting a given transfer function, H(z), into a suitable filter structure. Block or flow diagrams are often used to depict filter structures and they show the computational procedure for implementing the digital filter. The structure used depends on whether the filter is FIR or IIR.

The most widely used structure for FIR is the direct form as shown in Figure 2.4. This is because it is particularly simple to implement. In this form, the FIR is sometimes called a tapped delay line because it resembles a tapped delay line or transversal filter. Two other FIR structures that are also used are the frequency sampling structure and the fast convolution technique.

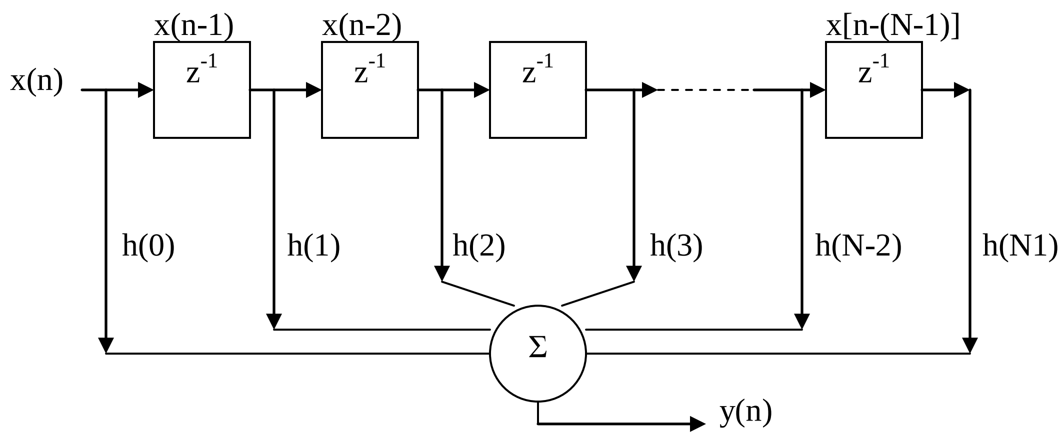

Figure 2.5: Transversal filter

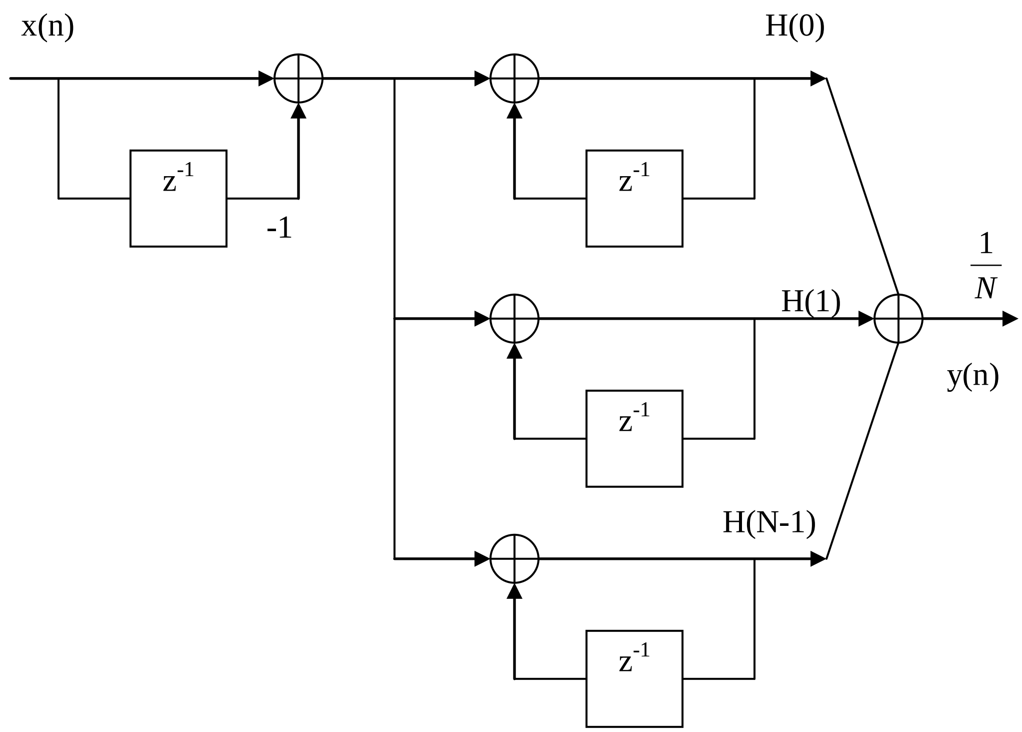

Figure 2.6: Frequency sampling structure

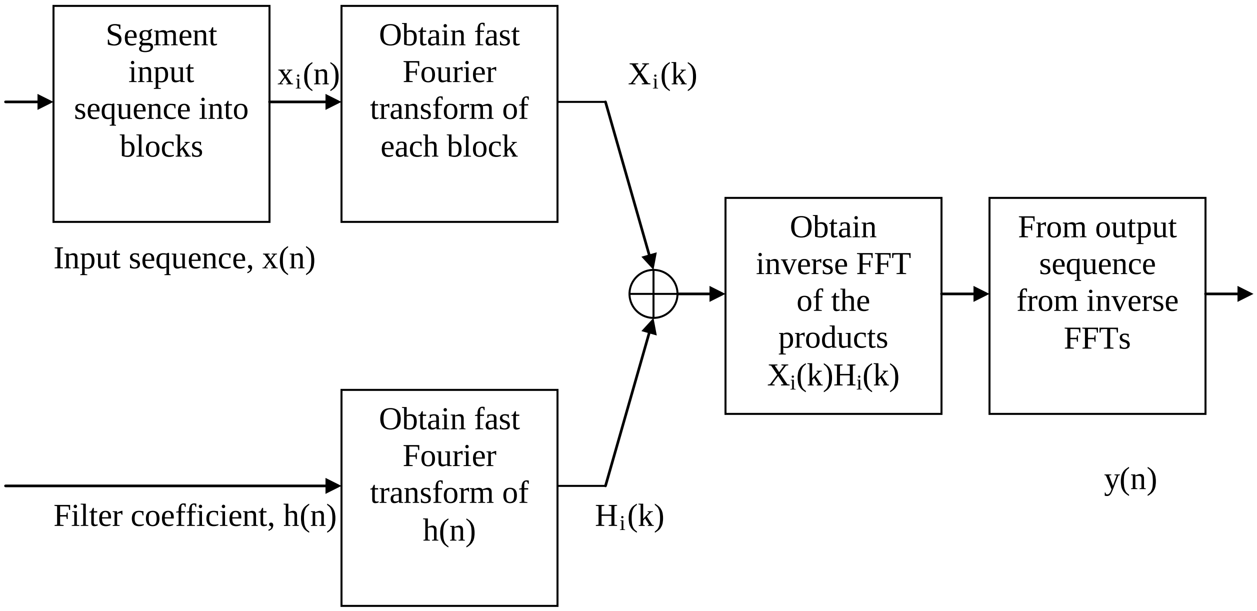

Figure 2.7: Fast convolution technique

2.5 IIR DIGITAL FILTER

Infinite impulse response, IIR, has input-output characteristic are governed by linear constant-coefficient different equations of a recursive nature. The transfer function of an IIR digital filter is a rational function in z^{-1}.

Realizable IIR digital filter are characterized by the following recursive equation:

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^{N} b_k x(n-k) - \sum_{k=1}^{M} a_k y(n-k) \qquad (2.5.1)$$

Where h(k) is the impulse response of the filter which is theoretically infinite in duration b_k and a_k are the coefficients of the filter, and x(n) and y(n) are the input and output to the filter. The transfer function for the IIR filter is given by

$$H(z) = \frac{b_o + b_1 z^{-1} + \ldots + b_N z^{-N}}{1 + a_1 z^{-1} + \ldots + a_M z^{-M}} \qquad (2.5.2)$$

$$= \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}} \qquad (2.5.3)$$

An important part of the IIR filter design process is to find suitable values for the coefficient b_k and a_k such that some aspect of the filter characteristic, such as frequency response, behaves in a desired manner.

Consequently, for a prescribed frequency response the use of an IIR digital filter usually results in a shorter filter length than the corresponding FIR digital filter. However, this improvement is achieved at the expense of phase distortion and a transient start-up that is not limited to a finite-time interval.

FIR filters offer several advantages over IIR filters:

i. Completely constant group delay throughout the frequency spectrum.

ii. Complete stability at all frequencies regardless of the size of the filter.

FIR filters also come with some disadvantages as well:

i. The frequency response is not as easily defined as it is with IIR filters

ii. The number of states required to meet a frequency specification may be far larger than that required for IIR filters.

2.5.1 STABILITY

From the equation 2.5.1, the current output sample, y(n) is a function of past output, y(n-k), as well as present and past input samples, x(n-k), that is the IIR filter is a feedback system of some sort. The strength of IIR filters comes from the flexibility the feedback arrangement provides. For example, an IIR filter normally requires fewer coefficients than an FIR for the same set of specification, which is why IIR filters are used when sharp cutoff and high throughput are the important requirements. The price for this is that the IIR filter can be unstable or its performance significantly degraded if adequate care is not taken in its design.

2.5.2 FILTER STRUCTURE

The input,x(n) and output,y(n) of a causal IIR filter with a rational system function:

$$H(z)= \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b_k z^{-k}}{1+\sum_{k=1}^{p} a_k z^{-k}} \qquad (2.5.2.1)$$

In this section, we will discuss several different implementation of IIR systems are presented, including the direct form filter structures, the cascade and the parallel.

There are two direct form filter structures, referred to as direct form I and direct form II. The direct form I structure is an implementation that results the above equation is written as a pair of different equation as follows:

$$w(n) = \sum_{k=0}^{q} b(k)x(n-k) \qquad (2.5.2.2)$$

$$y(n) = w(n) - \sum_{k=1}^{p} a(k)y(n-k) \qquad (2.5.2.3)$$

The first equation corresponds to an FIR filter with input, x(n) and output,w(n) and the second equation corresponds to an all-pole filter with input,w(n) and output,y(n). Therefore, this pair of equations represents a cascade of two systems. The direct form II structure is obtained by reversing the order of the cascade of B(z) and 1/A(z). With this implementation,x(n) is first filtered with all-pole filter 1/A(z) and then with B(z). The direct form II structure is said to be canonic because it uses the minimum number of delays for a given H(z).

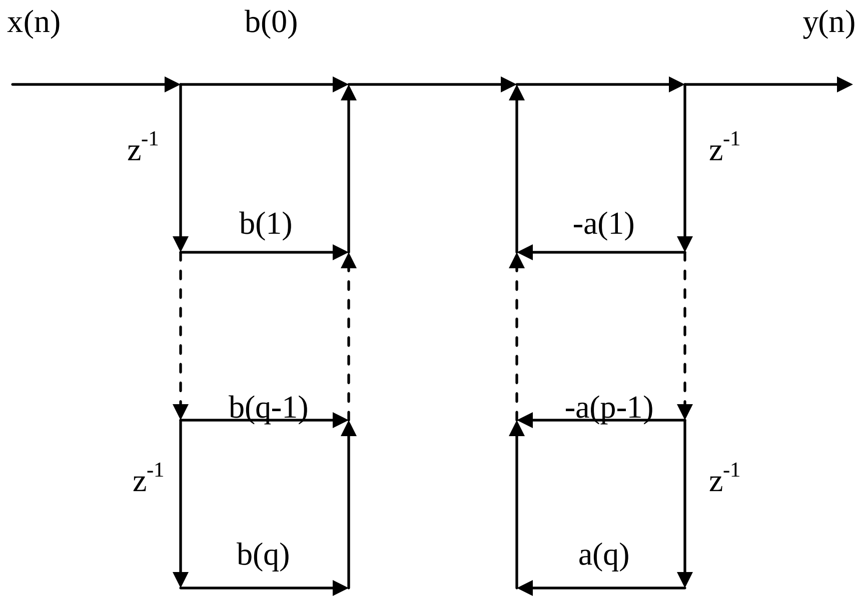

Figure 2.8: Direct form I realization of an IIR filter

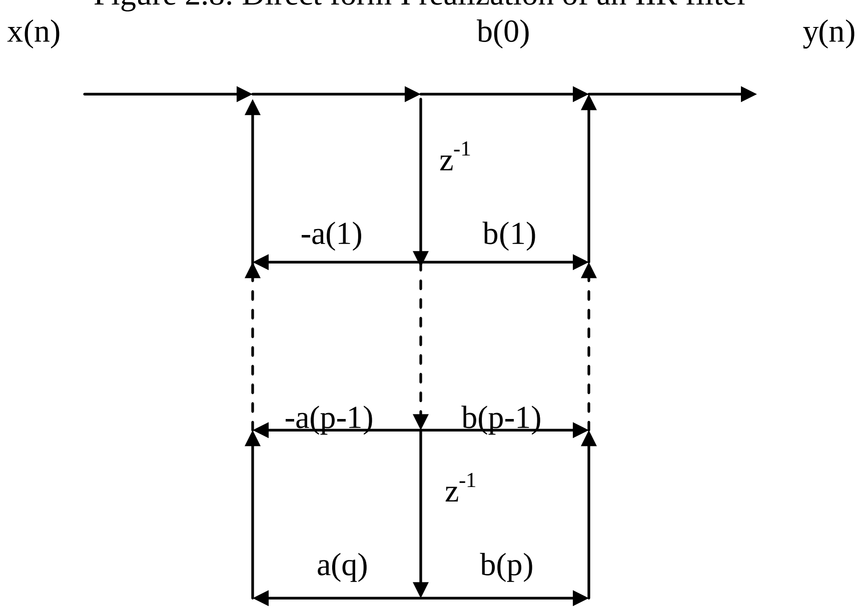

Figure 2.9: Direct form II or canonic realization

The cascade structure is derived by factoring the numerator and denominator polynomials of H(z):

$$H(z) = \frac{\sum_{k=0}^{q} b_k z^{-k}}{1 + \sum_{k=1}^{p} a_k z^{-k}} \qquad (2.5.2.4)$$

This factorization corresponds to a cascade of first-order filters, each having one pole and one zero. In general, the coefficient α_k and β_k will be complex. However, if h(n) is real, the roots of H(z) will occur in complex conjugate pairs, and these complex conjugate factors may be combined to form second-order factors with real coefficients:

$$H(z) = \frac{1 + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \qquad (2.5.2.5)$$

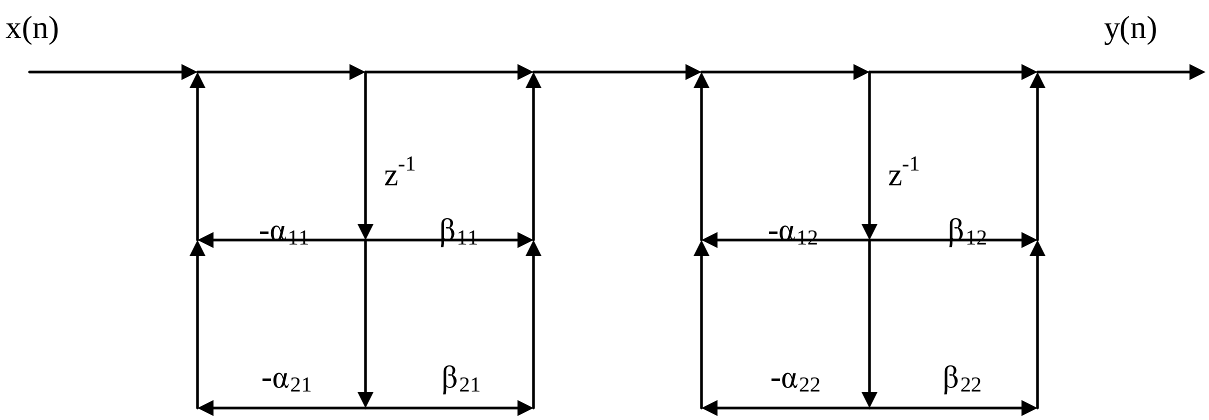

Figure 2.10: Cascade realization

An alternative to factoring H(z) is to expand the system using a partial fraction expansion, then we will have the following structure

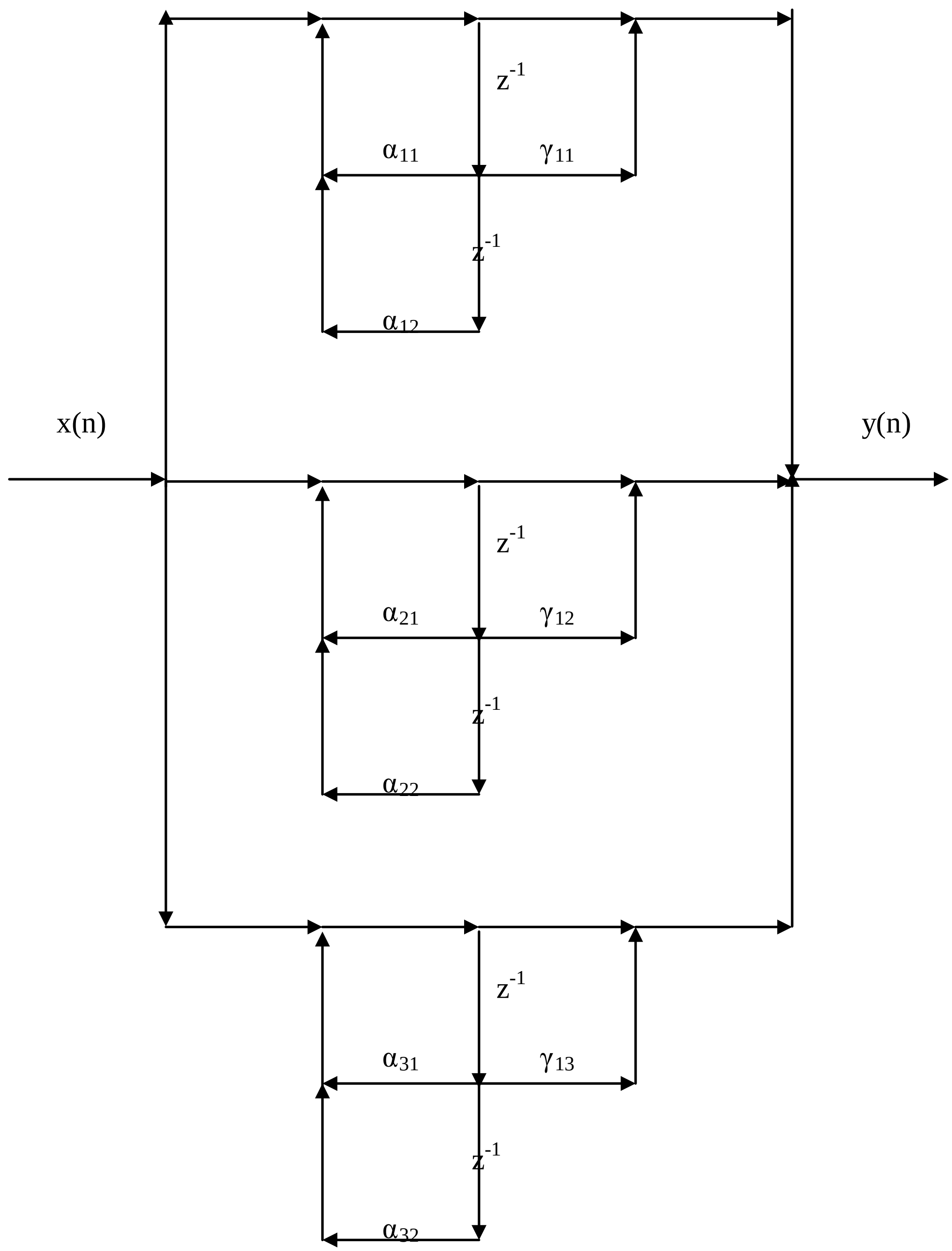

Figure 2.11: Parallel connection of three second-order direct form II structures