Chapter 1
Introduction

1.1 : Introduction

During the last two decades, Multirate filter banks have found various applications in many different areas, such as speech coding, scrambling, adaptive signal processing, image compression, signal and image processing applications as well as transmission of several signals through the same channel. The main idea of using Multirate filter banks is the ability of the system to separate in the frequency domain the signal under consideration into two or more signals or to compose two or more different signals into a single signal.

When splitting the signal into two or more signals an analysis-synthesis system is used. The analysis-synthesis systems under consideration are critically sampled multichannel or *M*-channel uniform filter banks as shown in Figures 1(a) and 1(b) respectively.

In the analysis bank of the uniform bank, the signal is split with the aid of *M* filters $H_k(z)$ for $k = 0,1,\ldots,M-1$ into *M* bands of the same bandwidth and each sub-signal is decimated by a factor of *M*. In the case of octave filter banks, the overall signal is first split into two bands of the same bandwidth and both sub-signals are decimated by a factor of two[5]. After that, the decimated low pass filtered signal is split into two bands and so on. Doing this three times gives rise to a three-level octave filter bank corresponding to the structure shown in Figure1(b). In this case, $H0(z)$ is a high pass filter with bandwidth equal to half the baseband and the decimation factor is 2, $H1(z)$ and $H2(z)$ are band pass filters with bandwidths equal to one fourth

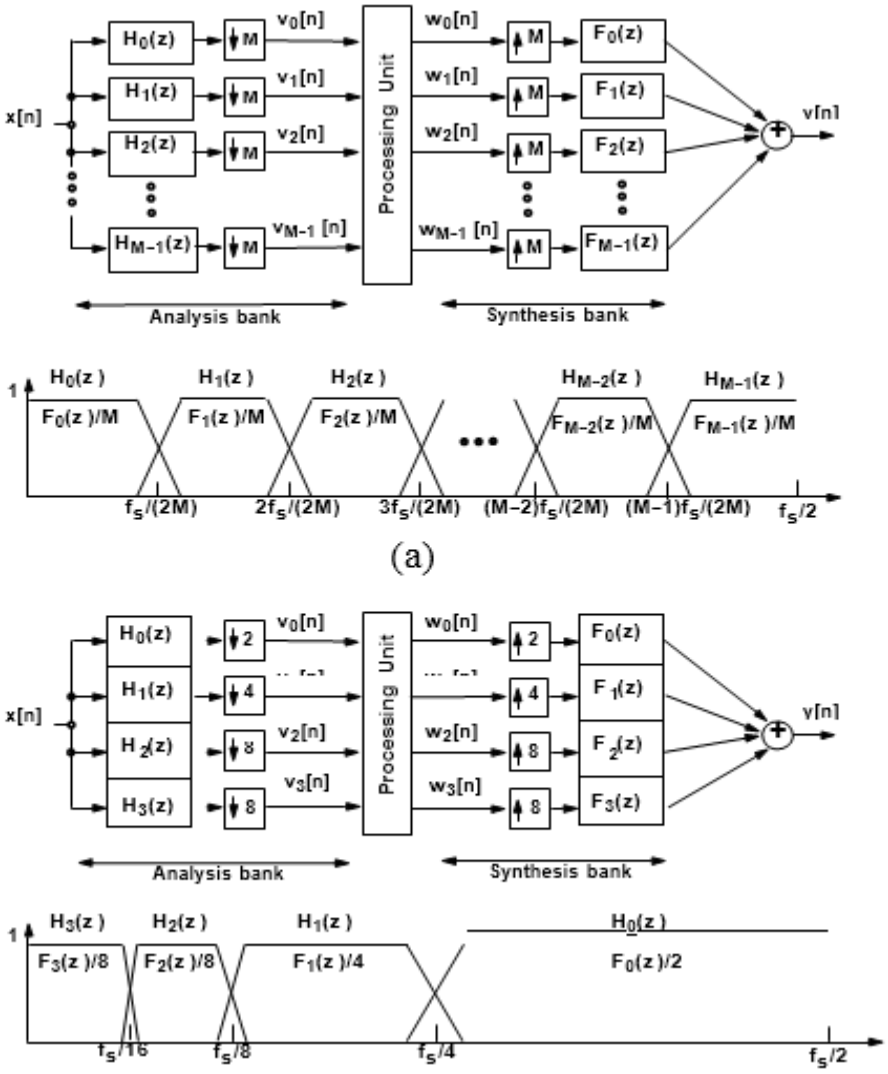


Figure 1. Analysis-synthesis filter bank. (a) *M*-channel uniform filter bank. (b) Three-level octave filter bank. Note that in the case of interpolation by a given factor, the corresponding filter should approximate this factor in the passband in order to preserve the signal energy.

Fig.1

and one eighth of the baseband, respectively, and the corresponding decimation factors are 4 and8, whereas *H*3(z) is a low pass filter with the same bandwidth and decimation factor as *H*2(*z*).

In many applications, the processing unit corresponds to storing the signal into the memory or transferring it through the channel. The main goal is to significantly reduce, with the aid of proper coding schemes, the number of bits representing the original signal for storing or transferring purposes.

When splitting the signal into various frequency bands with the aid of the analysis filter bank, the signal characteristics are different in each band and various numbers of bits can be used for coding and decoding the sub-signals. In some applications, the processing unit is used for treating the sub-signal in order to obtain the desired operation for the output signal of the overall system [5]. A typical example is the use of the overall system for making adaptive signal processing more efficient. The role of the filters in the synthesis part is to approximately reconstruct the original signal. This is performed in two steps. First, for the uniform filter bank, the M sub-signals at the output of the processing unit are interpolated by a factor of M and filtered by M synthesis filters $Fk(z)$ for k =0,1,…,M-1, whereas for the octave filter bank, the interpolation factors for the sub signals are the same as for the analysis part. Second, the outputs of these filters are added. In the transferring and storing applications, the ultimate goal is to design the overall system such that, despite of a significantly reduced number of bits used in the processing unit, the reconstructed signal is either a delayed version of the original signal or suffers from a negligible loss of information carried by the sub-signals. Another example is the de-noising of a signal performed with the aid of a special octave filter bank, called a discrete-time wavelet bankphase Filters of Multirate system proved to be one of the emerging and an excellent solution for the day to day problems of wireless communication

Specifically, Polyphase channel has been used in modern wireless system for performance improvement

1.2: Multirate filter in Wireless

A central problem within communication theory concerns how to efficiently transit it information –baring signals over unreliable channels. Depending on the application, the signals of interest may be continuous –time waveforms or discrete-time sequences, and they may be continuous in amplitude or discrete-valued. Examples include speech, image, and video signals, as well as various kinds of inherently digital data.

The communications problem is frequently partitioned into two problems – source coding and channel coding .Shannon's celebrated source – channel separation theorem ensures that for some important classes of channels these problems can be addressed independently without loss of performance. However, even when such partitioning cannot be justified in terms of the separation theorem, the approach is often popular for a variety of other reasons, among which are tractability of system design and robustness of the resulting system to errors in source and channel modeling.

1.2.1 Introduction to generation of Filter

In this section Base stations for cellular mobile communication systems offer an example of a radio receiver that must down-convert and demodulate multiple simultaneous narrow-band RF channels.

1.2.1a First generation of filter

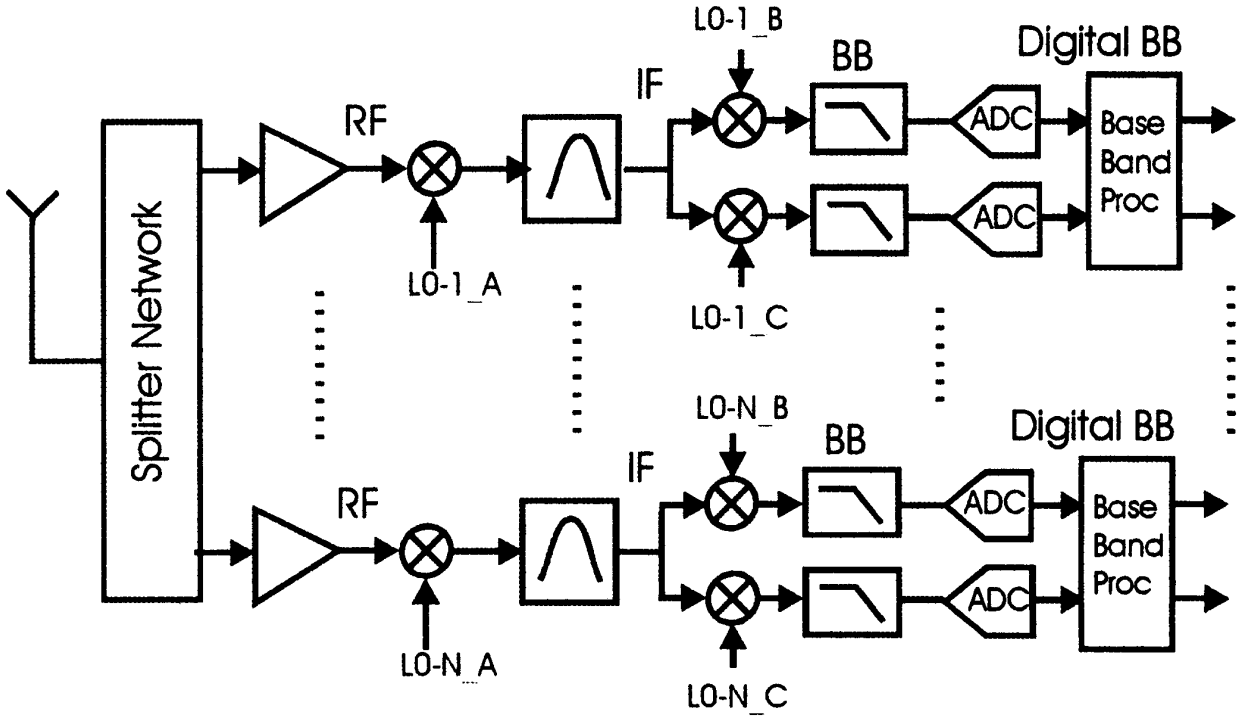


Fig. 2. First-generation RF architecture of N-channel receiver.[4]

This architecture contains sets of dual-conversion sub-receivers. Each receiver amplifies and down-converts a selected RF channel to an IF filter that performs initial bandwidth limiting. The output of each IF filter is again down converted to baseband by matched quadrature mixers that are followed by matched baseband filters that perform final bandwidth control.

Each quadrature down-converted signal is then converted to their digital representation by a pair of matched ADCs. The output of the ADCs is processed by DSP engines that perform the required synchronization, equalization, demodulation, detection, and channel decoding.

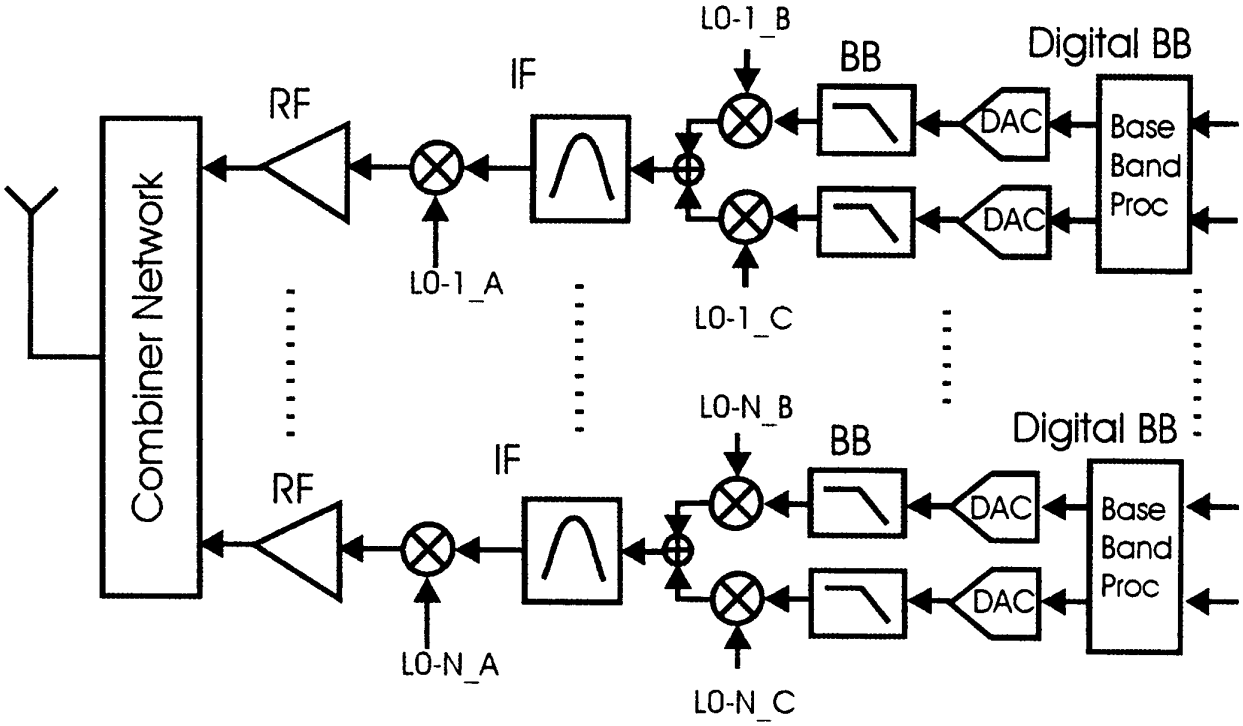


Fig.3 :First-generation RF architecture of N-channel transmitter [4]

The above figure shows a base-station companion radio transmitter formed by N sets of dual conversion sub-transmitters that modulate and up-convert multiple simultaneous narrow-band RF channels.

The signal flow for the transmitter chain is simply a reversal of the signal flow of the receiver chain.

Disadvantage of first generation of filter

Gain and phase imbalance between the two paths containing the Quadrature mixers, Analog baseband filters, and ADC in an N -channel receiver or N -

channel transmitter is the cause of crosstalk between the in-phase and quadrature (*I/Q*) components . This, in turn, results in coupling between the many narrow-band channels sometimes called ghosts or images.

1.2.1b Second generation of Filter

In the second-generation, multichannel receiver and transmitter in which the conversion from analog to digital (or digital to analog) occurs at IF rather than at baseband.

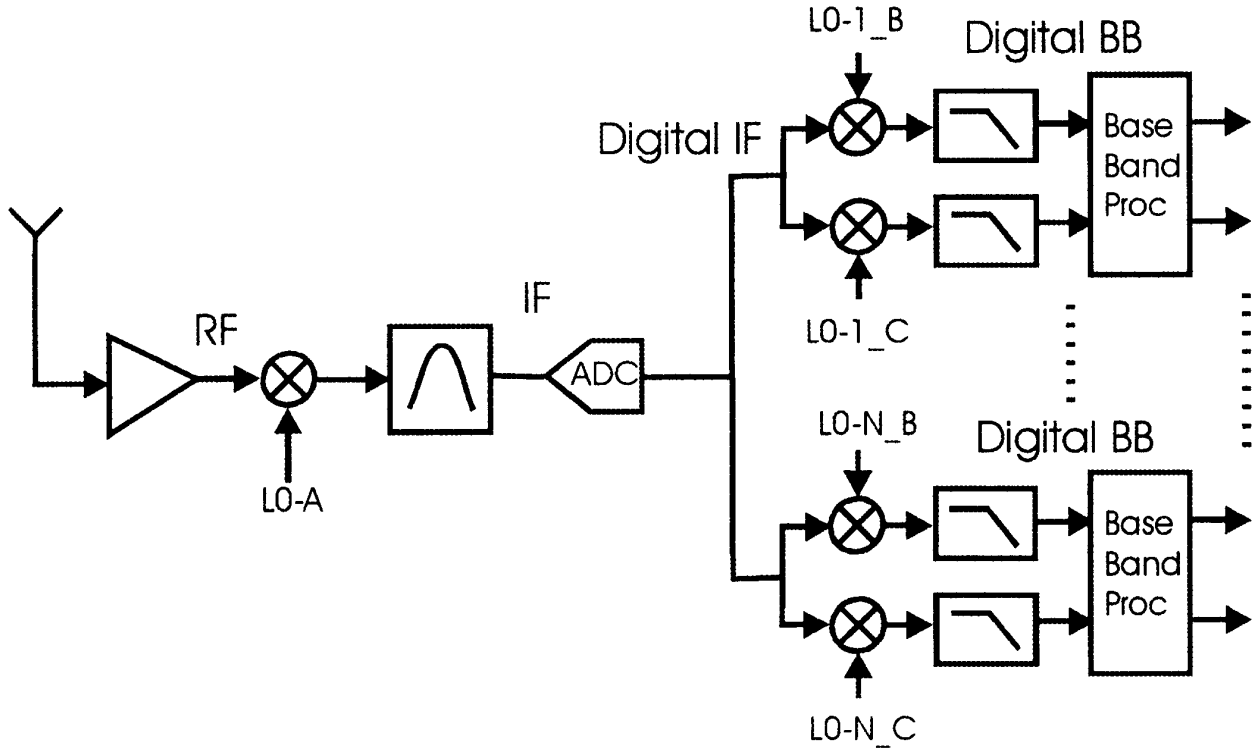


Fig.4.Second-generation RF architecture of N-channel receiver [4]

The receiver, we see that the down conversion of the separate channels is performed by a set of digital down converters and digital low-pass filters. The digital

process can realize arbitrarily small levels of imbalance by controlling the number of bits involved in the arithmetic operations. Precision of coefficients used in the filtering process sets an upper bound to spectral artefact levels at 5 dB/bit so that

12-bit arithmetic can achieve image levels below 60 dB. Thus, the DSP-based complex down conversion does not introduce significant imbalance-related spectral terms. Similar comments can be applying to the DSP based up-conversions in the digital transmitter [4].

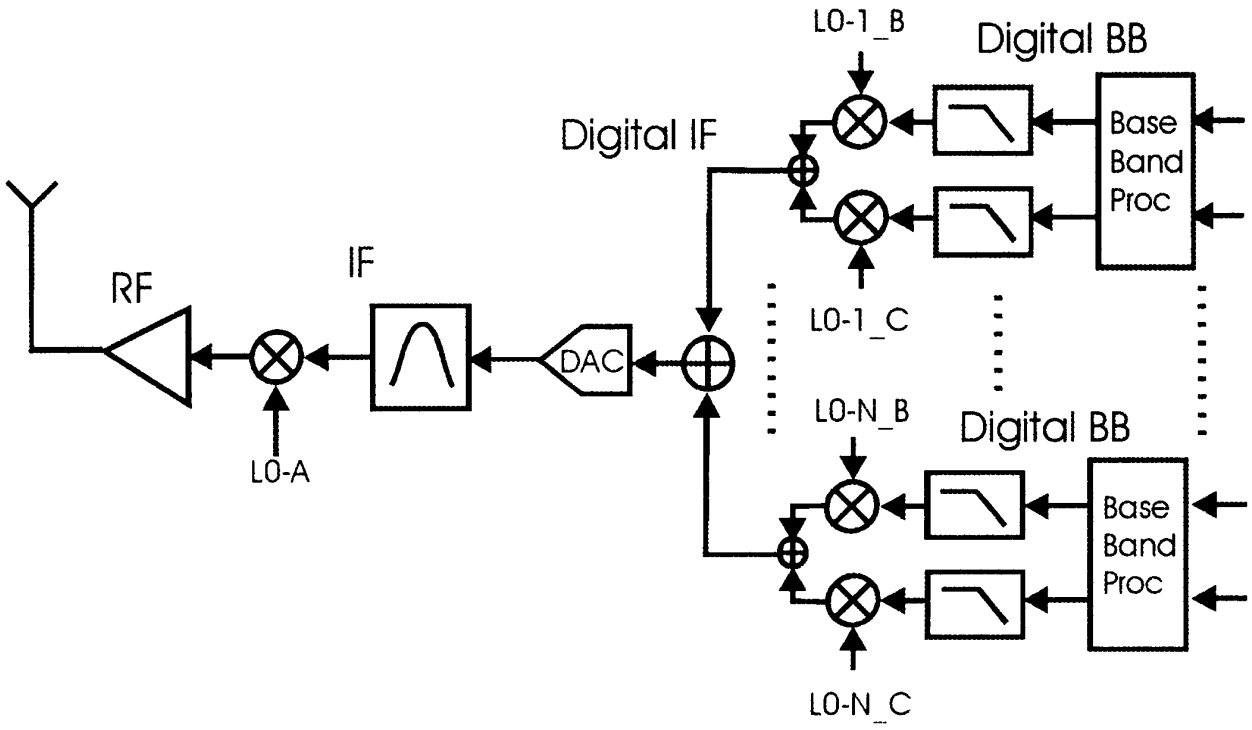


Fig.5:Second-generation RF architecture of N-channel transmitter [4]

The Similar comments can apply to the DSP based up-conversions in the second generation of digital transmitter.

Advantages of second generation

(a) The levels of spectral images are controlled to be below the quantizing noise floor of the ADC or DAC involved in the conversion process.

(b) The digital filters following or preceding the mixers are designed to have linear phase characteristics, a characteristic trivially simple to realize in digital Non recursive filters.

Disadvantage of second generation

The dynamic range and conversion speed of the ADC and DAC becomes the limiting factor in the application of the architectures of second generation shown above.

1.2.1b Third generation hybrid filter

To extend the application range of digital N -channel receivers and N digital - channels transmitters, we often use a hybrid scheme.

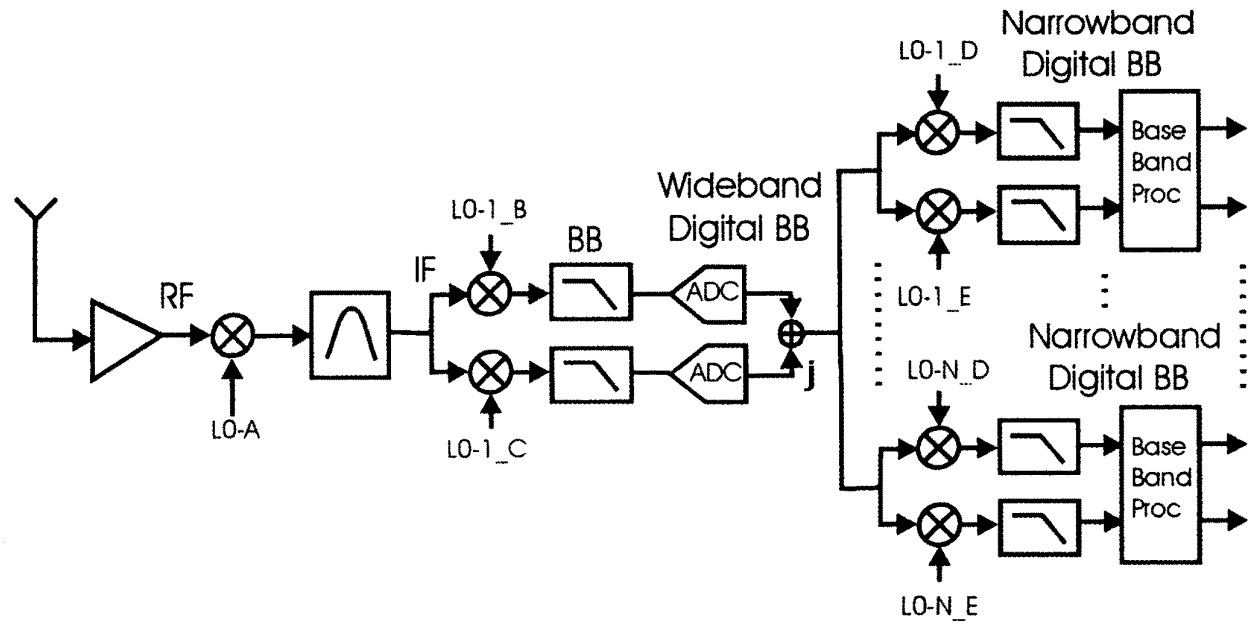


Fig.6: Second-generation hybrid RF digital N-channel receiver. [4]

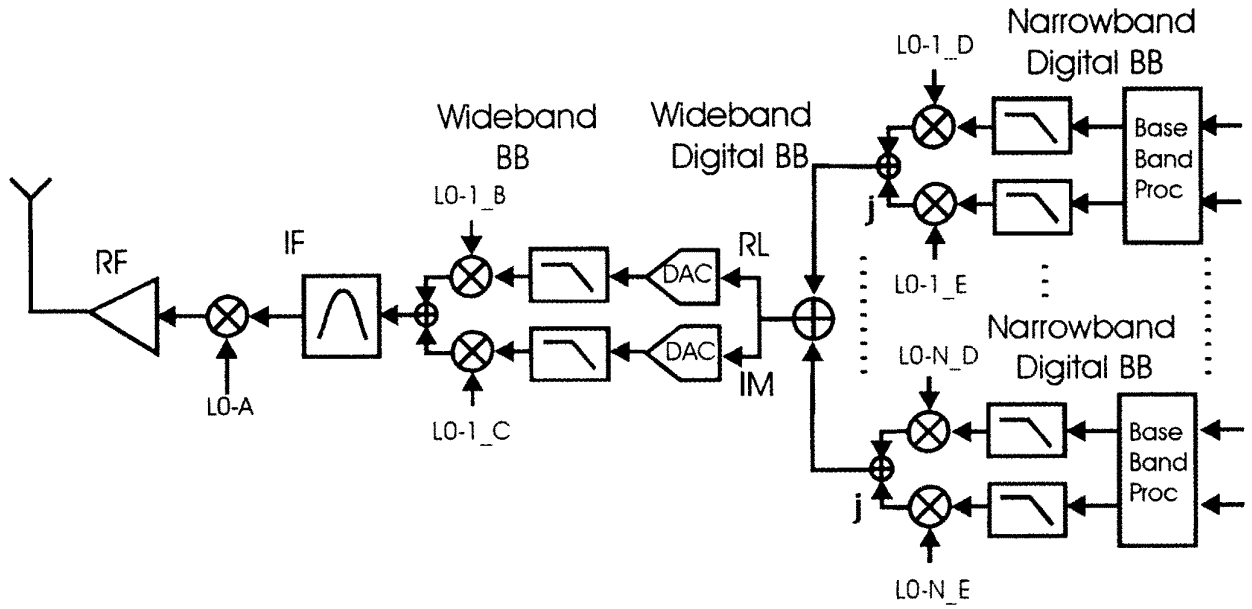


Fig.7: Second-generation hybrid RF digital N-channel transmitter. [4]

To extend the application range of digital N -channel receivers and digital N -channels transmitters, we often use a hybrid scheme in which the initial complex down conversion is performed with analog *I/Q* mixers and the channelization is performed digitally after the ADC [4]. The first conversion can be considered a block conversion to baseband that delivers the frequency band of interest to the DSP arena for subsequent channelization. The hybrid forms of the digital N -channel receiver and the digital N -channel transmitter are shown in Figs. 8 and 9, respectively. DSP techniques are applied to the digitized *I/Q* data to balance the gain and phase offsets in the analog ADC and DAC. DSP-based *I/Q* balance correction is a standard signal conditioning task in high-end, as well as consumer-based, receivers and transmitters .

1.2.2 Current scenario of filter

In previous generation, we described the process of sampling an analog IF signal or complex analog baseband signal containing the set of frequency-division-multiplexed (FDM) channels to be further processed or channelized by DSP techniques. We consider the input signal to be composed of many equal-bandwidth equally spaced FDM channels. These many channels are digitally down converted to baseband, bandwidth constrained by digital filters, and subjected to a sample rate reduction commensurate with the bandwidth reduction.

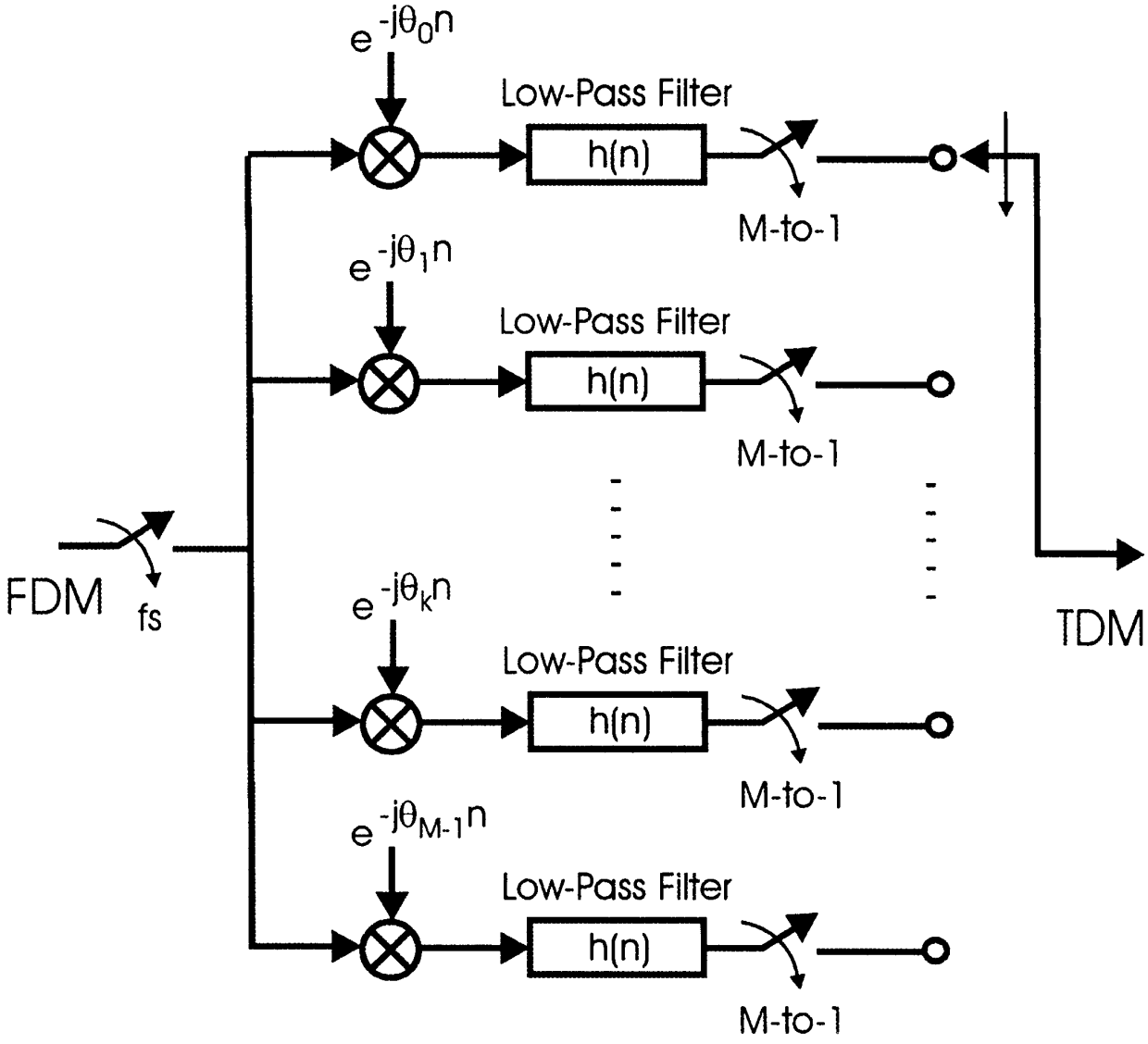
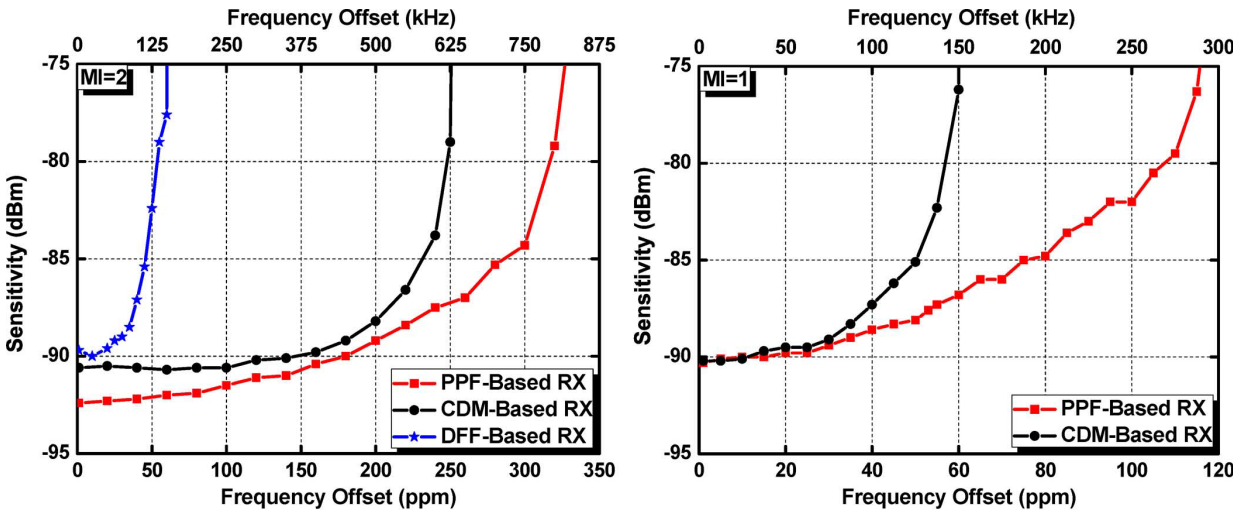


Fig.8: Conventional channelizer as a replica of analog prototype: down converters, baseband filters and resamplers [4].

This survey literature contains a detailed description of the ways in which the Polyphase Filters are implemented in various fields of wireless communication system.

The use of low power **hybrid Polyphase filter** in BFSK receiver to achieve a high frequency offset tolerance at a low MI is demonstrated [2]. The above is achieved by frequency to energy conversion of data samples using PPFs architecture. However, PPFs were originally utilized to reject image signals in low-IF receivers and also as a demodulator in a zero-IF BFSK receiver but extra filters were required for channel selection and interference rejection. The hybrid PPF based receiver architecture proposed in this work for medium data rate wireless network application simultaneously achieves frequency-to-energy conversion, channel selection, interference rejection and an improved ACR is obtained at reduced power consumption within the PPFs which also leads to a reduction in the hardware and power consumption.

.



(a) (b)

Fig.9:Comparison of frequency offset tolerance with conventional BFSK receivers. (a). For PPF-based RX, optimal PPF bandwidth=1.4MHz, order=3. For CDM/DFF based RX, optimal LPF bandwidth=2.05bMHz, order=6. (b)For PPF-based RX, optimal PPF bandwidth=1.4 MHz, order=3. For CDM-based RX, optimal LPF bandwidth=1.55MHz, order=5

In research papers [4] a digital transceiver wireless system using **Polyphase Filter Banks** is proposed. The detailed analytical model involving mathematical description and literature theorems for developing PPF channelizer along with the step by step transformation of existing pass band filters [4] into an approximate all pass M-path filters is outlined. The suggested description of PPF banks as a sequential rearrangement of operation of mixing, filtering, and re sampling lays ways to develop remarkably efficient signal processing structures. The paper involves a detailed discussion on the use of PPF channelizer in receiver architecture and simply provides an overview to the requirement PPF channelizer in the transmitter architecture.

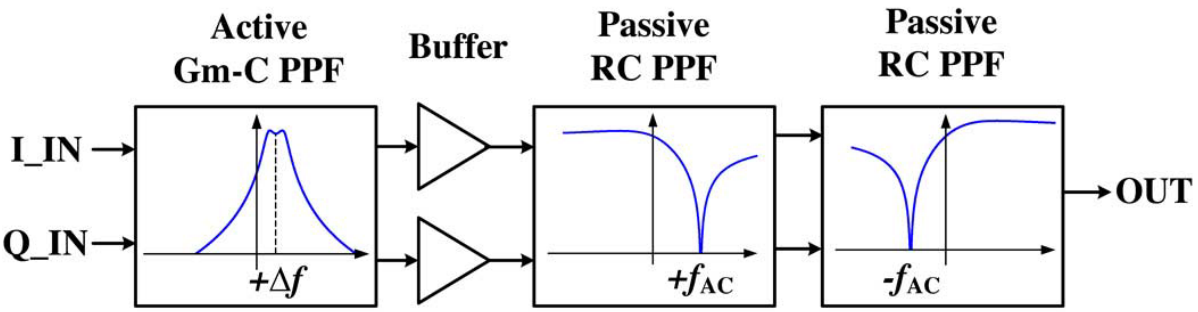
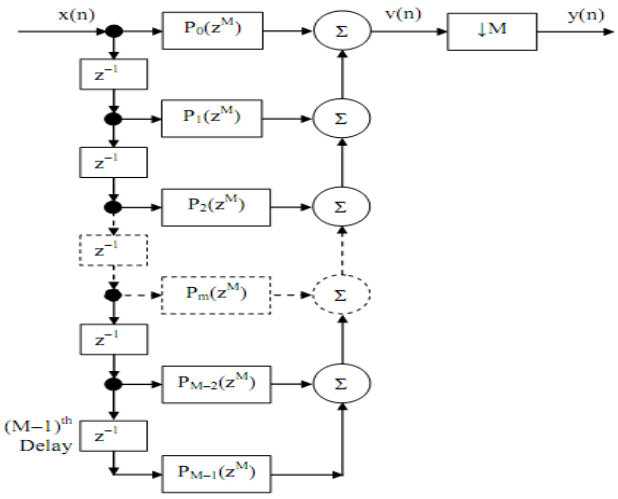


Fig.10.a: Block diagram of Hybrid Multirate Filter

b: Block Diagram of m-stage Polyphase Filter

.



3.1: Basics and Types of Multirate filters

Multirate simply means "multiple sampling rates". A Multirate DSP system uses multiple sampling rates within the system. Whenever a signal at one rate has to be used by a system that expects a different rate, the rate has to be increased or decreased, and some processing is required to do so. Therefore "Multirate DSP" really refers to the art or science of changing sampling rates [3].

Multirate signal processing is processing by Sampling Rate alteration based on two basic principles; namely, Decimation & Interpolation

- Decimation - Process to decrease the sampling rate (frequency) by a constant factor (integer)
- Interpolation- Process to increase the sampling rate (frequency) by a constant factor(integer)

Decimation:

- Decimation is carried with a anti-aliasing filter H(z)and down sampler M, where M is a positive integer,operation is implemented by keeping every M-th sample of x[n] and removing M-1 in-between samples to generate y[m] i.e. y[m] has (1/M)-th times samples of x[n]

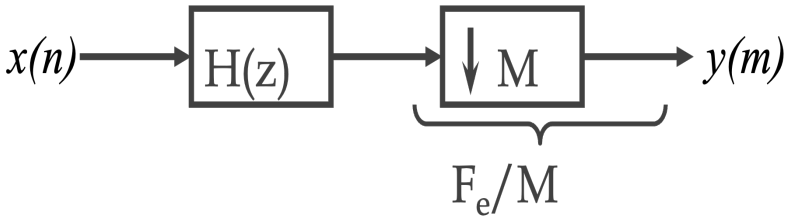


Fig.11: block diagram of decimation

- Digital LPF filter compresses the bandwidth before down sampling to prevent aliasing at reduced rate.

Interpolation:

- Interpolation is carried with a up sampler L and anti-imaging filter H(z), where L is a positive integer, operation is implemented by inserting L-1 equidistant zero-valued samples between two consecutive samples of x[m] i.e. *y*[n] has (L)-th times samples of *x*[m].

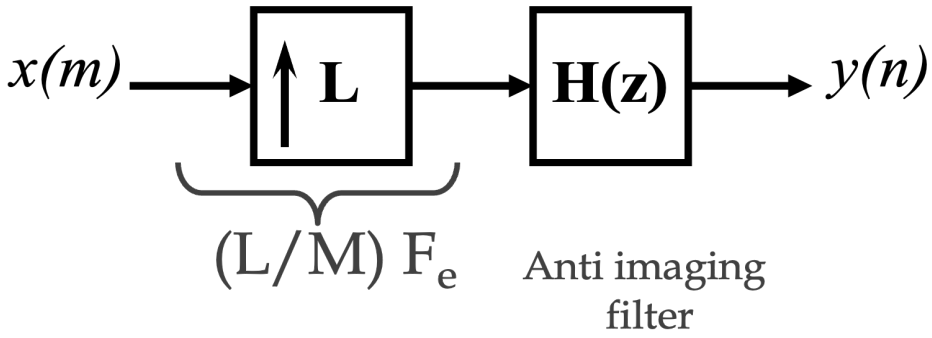


Fig.12: block diagram of interpolation

- Digital LPF filter expands the bandwidth after up sampling to prevent imaging at increased rate [3].

There are 2 types of Multirate filters:

1. Hybrid filters
2. Polyphase filters

3.1.1 Hybrid Filter

Introduction:

A Butterworth PPF with a bandwidth of 1.4 MHz and an order of 3 gives optimal sensitivity and FOT with a MI of 1 and 2. Furthermore, a hybrid topology is proposed for the PPFs which achieves an improved ACR at reduced power consumption. The block diagram of the hybrid PPF is shown in Fig. 8(a). It consists of an active PPF based on second-order Chebyshev LPFs, and two passive RC PPFs with nulls at $\pm f_{AC}$, the center frequency of the adjacent channel signal (f_{AC}=4 MHz and f_{AC} MHz fora MI of 2 and 1, respectively, referringtoFig.3).The system level simulations yield an optimal 1-dB bandwidth for the active Chebyshev filter as 1 MHz and 0.9 MHz for MI=2 and MI=1, respectively.

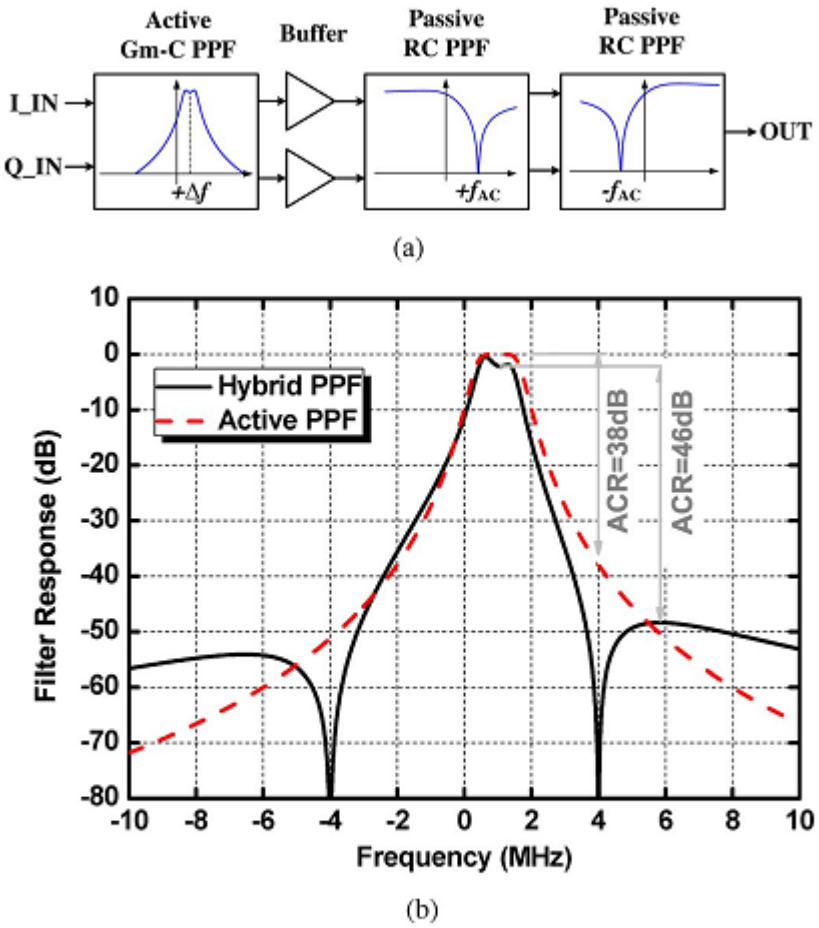


Fig. 8. Hybrid PPF. (a) Block diagram. (b) Frequency response compared with an active PPF with 8 dB improvement in ACR.

Fig.13 (ref. Fig8[2])

The hybrid topology has two advantages over an active one. First, power is reduced by using an active filter of a lower order. One reason is that the active PPF requires less out-of-band attenuation because extra attenuation is provided by the RC PPFs. The other reason is that a Chebyshev implementation is feasible because the PPF-based energy detection is less sensitive to phase distortion compared with the conventional zero-crossing based detection. The Chebyshev implementation requires a lower order and, therefore, a lower power consumption for the same amount of stop band attenuation. The noise penalty from the passive PPFs is mitigated by the high gain in the RF front-end. Second, as shown in Fig. 8(b), due to the nulls placed at the centre frequency of the

adjacent channels, the hybrid PPF achieves 46 dB ACR, an improvement of 8 Db over the active Butterworth PPF.

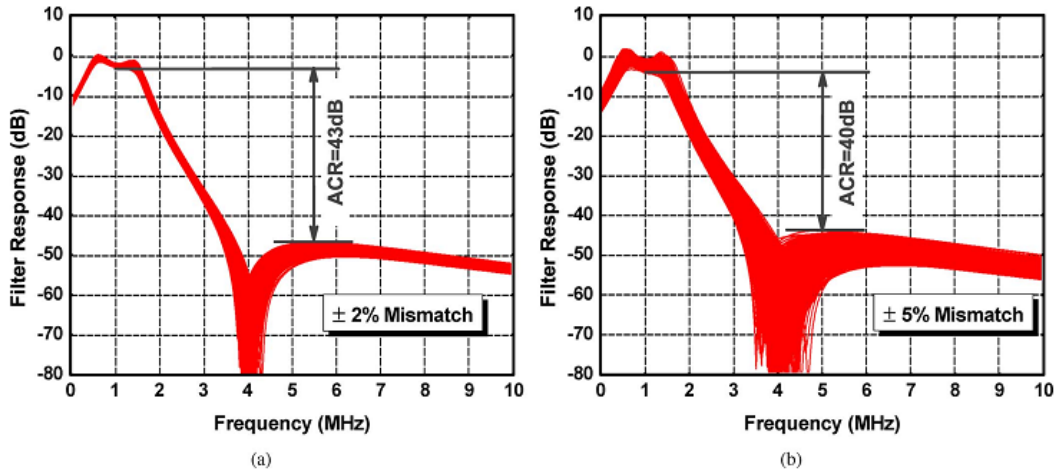


Fig. 9. Monte Carlo simulations of the hybrid PPF frequency response. (a) With ±2% mismatch. (b) With ±5% mismatch.

Fig.14 (ref. Fig 9[2])

Component mismatches affect the transfer function of the filter and, therefore, the ACR. By choosing a Gm-C implementation for the active PPF to save power consumption, the sources of mismatch include the transconductance (Gm) mismatch between the Gm cells, and R and C mismatches. Monte Carlo simulations have been performed to show the effect of mismatch on the hybrid PPF and the results are shown in Fig. 9. The worst case ACR is 43 dB and 40 dB, respectively, for 2% and 5%mismatch in Gm, R and C. Layout techniques were utilized to improve the component matching. The measured ACR of 46 dB verifies the excellent matching achieved after fabrication.

3.1.2 Polyphase Filter

Introduction:

Polyphase is a way of doing sampling-rate conversion that leads to very efficient implementation. But more than that, it leads to very general viewpoints that are useful in building filter banks. Polyphase filtering is a computationally efficient structure for applying resampling and filtering to a signal. Most digital filters can be applied in a polyphase format, and it is also possible to create efficient resampling filter banks using the same theories.

Transforming the channelizer:

The current configuration of the single channel down converter involves a band pass filtering operation followed by a down sampling of the filtered data to alias the output spectrum to baseband. Following the idea developed in the previous section that led us to down-convert only those samples retained by the down sampler; we similarly conclude that there is no need to compute the output samples from the pass band filter that will be discarded by the down sampler. We now interchange the operations of filter and down sample with the operations of down sample and filter. The process that accomplishes this interchange is known as the Noble Identity[4].

The noble identity is compactly presented in figure 20 which we describe with similar conciseness by “The output from a filter H (zm) followed by an M-to-1 down sampler is identical to an M-to-1 down sampler followed by the filter H(Z)”. The Z Min the filter impulse response tell us that the coefficients in the filter are separated M samples rather than the more conventional one sample delay between coefficients in the filter H(Z). We must take care to properly interpret the operation of the M-to-1 down sampler. The interpretation is that the M-to1 down sampled time series from a filter processing every Mth input sample presents the same output by first down sampling the input by M-to-1 to

discard the samples not used by the filter to compute the retained output samples and then operating the filter on the retained input samples. The noble identity works because M-samples of delay at the input clock rate is the same interval as one sample delay at the output clock rate [4].

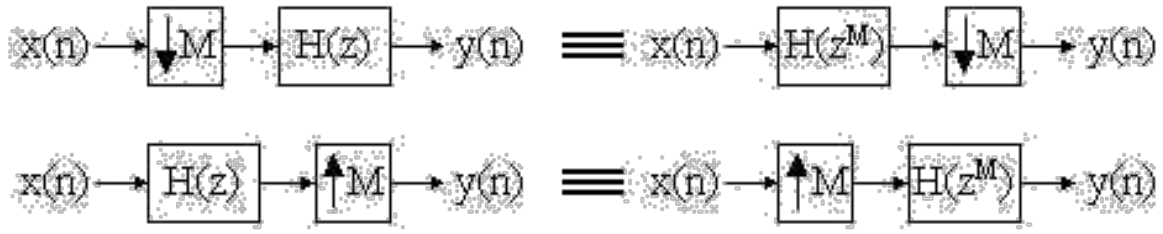


Fig.15: Noble Identity: A Filter Processing Every M-th Input Sample Followed by an Output M-to-1 Down/up Sampler is the same as an Input M-to-1 Down/up Sampler Followed by a Filter Processing Every M-th Input Sample.

"Under what condition does a filter manage to operate on every M-th input sample?" We answer this query by rearranging the description of the filter to establish this condition so that we can invoke the noble identity. This rearrangement starts with an initial partition of the filter into M-parallel filter paths. The Z transform description of this partition is presented in equations 11 through 14, which we interpret in figures 21 through 23. For ease of notation, we first examine the base-band version of the noble identity and then trivially extend it to the pass band version.

$$H(Z) = \sum_{n=0}^{N-1} h(n)\, Z^{-n}$$
$$= h(0) + h(1)\, Z^{-1} + h(2)\, Z^{-2} +$$
$$h(3)\, Z^{-3} + \cdots + h(N-1)\, Z^{-(N-1)} \qquad (1)$$

Anticipating the M-to-1resampling, we partition the sum shown in eq-1 to a sum of sums as shown in eq-2. This partition maps a one-dimensional array of weights (and index markers Z^{-n}) to a two dimensional array. This mapping is sometimes called lexicographic, for natural order, a mapping that occurs in the CooleyTukey fast Fourier transform. In this mapping we load an array by columns but process the array by rows. In our example, the partition forms columns of length M containing M successive terms in the original sum, and continues to form adjacent M-length columns till we account for all the elements of the original one-dimensional array.

$$
\begin{aligned}
H(Z) = h(0) \quad &+ h(M+0)Z^{-M} + H(2M+0)Z^{-(2M+0)} + \ldots\ldots \\
&h(1)Z^{-1} + h(M+1)Z^{-(M+1)} + h(2M+1)Z^{-(2M+1)} + \ldots\ldots \\
&h(2)Z^{-2} + h(M+2)Z^{-(M+2)} + h(2M+2)Z^{-(2M+2)} + \ldots\ldots \\
&h(3)Z^{-3} + h(M+3)Z^{-(M+3)} + h(2M+3)Z^{-(2M+3)} + \ldots\ldots \\
&\quad . \quad . \qquad . \quad . \qquad . \quad . \\
&\quad . \quad . \qquad . \quad . \qquad . \quad . \\
&h(M-1)Z^{-(M-1)} + h(2M-1)Z^{-(2M-1)} + h(3M-1)Z^{-(3M-1)} + \ldots\ldots \qquad (2)
\end{aligned}
$$

We note that the first row of the two dimensional array is a polynomial in Z^M, which we will denote $H_0(Z^M)$ a notation to be interpreted as an addressing scheme to start at index 0 and increment in stride of length M. The second row of the same array, while not a polynomial in Z^M, is made into one by factoring the common Z-term and then identifying this row as $Z^{-1}\,H_1(Z^M)$. It is easy to see that each row of above equation can be described as $Z^{-r}\,H_r(Z^M)$ so that above equation can be re-written in a compact form :

$$H(Z) = H_0(Z^M) + Z^{-1}H_1(Z^M) + Z^{-2}H_2(Z^M) + \ldots\ldots\ldots + Z^{-(M-1)}H_{(M-1)}(Z^M) \qquad (3)$$

We rewrite above equation in the traditional summation form, which describes the original polynomial as a sum of delayed polynomials in Z^M.

$$H(Z) = \sum_{r=0}^{M-1} Z^{-r} H_r(Z^M)$$
$$= \sum_{r=0}^{M-1} Z^{-r} \sum_{n=0}^{(N/M)-1} h(r+nM) Z^{-Mn} \qquad (4)$$

The block diagram reflecting this M-path partition of a resampled digital filter is shown in figure 16. The output of the filter is the resampled sum of the output of the M separate filter stages along the M-paths. We pull the resample through the output summation element and down sample the separate outputs, only performing the output sum for the retained output sample points. With the resamplers at the output of each filter, which operates on every M-th input sample, we are prepared to invoke the noble identity and pull the resampler to the input side of each filter stage. This is shown in figure 17. The input resamplers operate synchronously, all closing at the same clock cycle. When the switches are close, the signal delivered to the filter on the top path is the current input sample. The signal delivered to the filter one path down is the content of the one stage delay line, which of course is the previous input sample. Similarly, as we traverse the successive paths of the M-path partition, we find upon switch closure, that the k-th path receives a data sample delivered k-samples ago. We conclude that the interaction of the delay lines in each path with the set of synchronous switches can be likened to an input commutator that delivers successive samples to successive legs of the M-path filter. This interpretation is shown in figure 18.

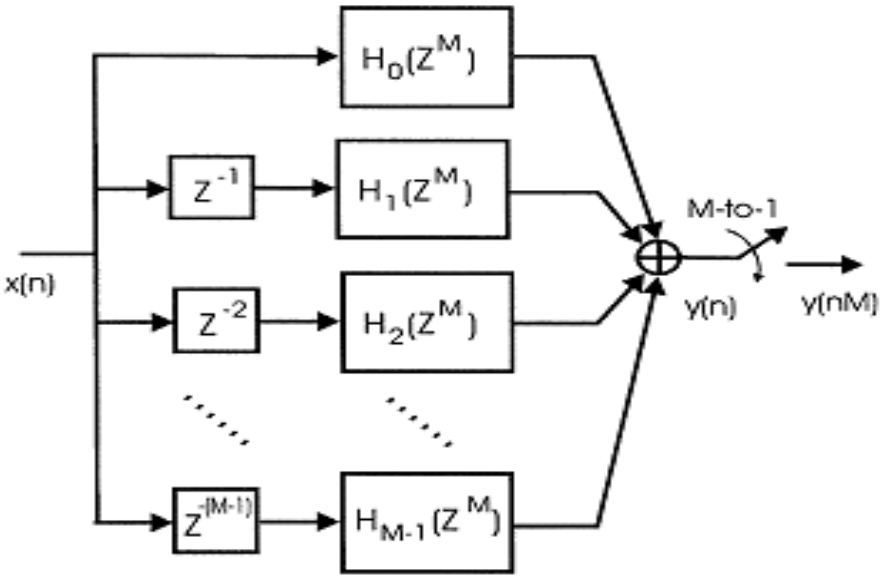


Fig. 21. M-path partition of prototype low-pass filter with output resampler.

Fig.16 (ref. Fig 21 [4])

We now complete the final steps of the transform that changes a standard mixer down converter to a resampling M-Path down converter. We note and apply the frequency translation property of the Z-Transform.

Interpreting the relationship presented in eq-5, we note that if h(n), the impulse response of a base band filter, has a Z transform H(Z), then the sequence $h(n)e^{+j\theta n}$, the impulse response of a pass band filter, has a Z-transform H(Z ejθn). Simply stated, we can convert a low pass filter to a band pass filter by associating the complex heterodyne terms of the modulation process either with the filter weights or with the delay elements storing the filter weights.

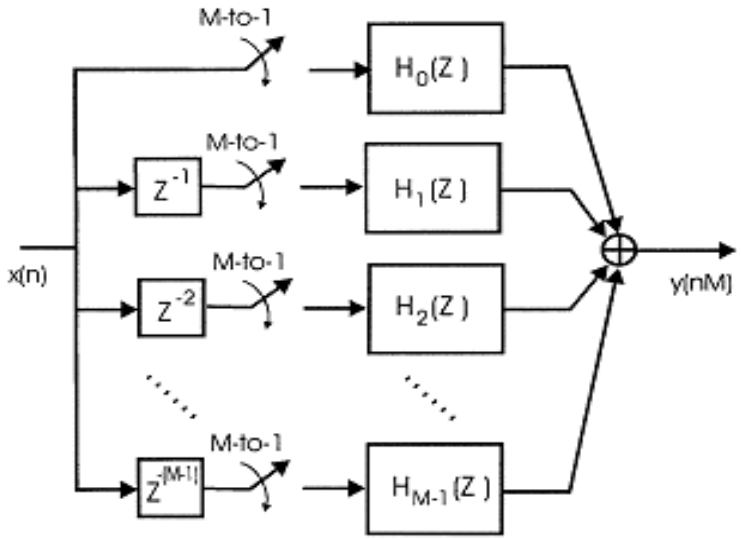


Fig. 22. M-path partition of prototype low-pass filter with input resamplers.

Fig.17 (ref. Fig 22 [4])

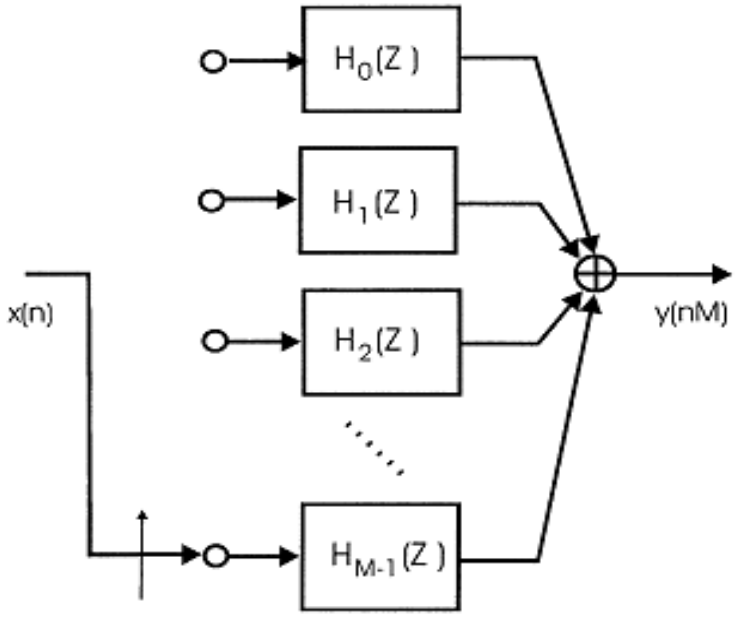


Fig. 23. M-path partition of prototype low-pass filter with input path delays and M-to-1 resamplers replaced by input commutator.

Fig.18 (ref. Fig. 23[4])

$$\text{if} \quad H(Z) = h(0) + h(1)Z^{-1} + h(2)Z^{-2} + \cdots + n(N-1)Z^{-(N-1)} = \sum_{n=0}^{N-1} h(n)Z^{-n}$$

and

$$G(Z) = h(0) + h(1)e^{j\theta}Z^{-1} + h(2)e^{j2\theta}Z^{-2} + \cdots + h(N-1)e^{j(N-1)\theta}Z^{-(N-1)} = h(0) + h(1)[e^{-j\theta}Z]^{-1} + h(2)[e^{-j\theta}Z]^{-2} + \cdots + h(N-1)[e^{-j\theta}Z]^{-(N-1)} = \sum_{n=0}^{N-1} h(n)[e^{-j\theta}Z]^{-n}$$

then

$$G(Z) = H(Z)\Big|_{Z=e^{-j\theta}Z} = H(e^{-j\theta}Z) \qquad (5)$$

We now apply this relationship to figure 18 by replacing each Z with Z $e^{-j\theta,}$ or perhaps more clearly, replacing each Z-1with Z-1ejθ, with the phase term θ satisfying the congruency constraint of the previous section, that θ=k(2π/M). Thus Z^{-1} is replaced with Z^{-1} $e^{jk(2\pi/M),}$ and Z^{-M} is replaced with Z{MejkM(2π/M)}. By design, the kM-th multiple of 2π/M is a multiple of 2πfor which the complex phase rotator term defaults to unity, or in our interpretation, aliases to base band (DC). The default to unity of the complex phase rotator occurs in each path of the M-path filter shown in figure 24. The non-default complex phase angles are attached to the delay elements on each of the M paths. For these delays, the terms Z-rare replaced by the terms Z^{-r} $e^{jkr(2\pi/M).}$ The complex scalar $e^{jkr(2\pi/M)}$ attached to each path of the M-path filter can be placed anywhere along the path, and in anticipation of the next step, we choose to place the complex scalar after the down sampled path filter segments $H_r(Z)$. This is shown in figure 19.

The modification to the original partitioned Z -Transform of eq to reflect the added phase rotators of figure 19 is shown in eq.

$$H(Ze^{-j\frac{2\pi}{M}k}) = \sum_{r=0}^{M-1} Z^{-r} e^{j\frac{2\pi}{M}rk} H_r(Z) \tag{6}$$

The computation of the time series obtained from the output summation in figure 24 is shown in eq-17. Here the argument nM reflects the down sampling operation which increments through the time index in stride of length M, delivering every M-th sample of the original output series. The variable yr (nM) is the nM-th sample from the filter segment in the r-th path, and y(nM,k) is the nM-th time sample of the time series from the k-th centre frequency. Remember that the down converted centre frequencies located at integer multiples of the output sample frequency are the frequencies that alias to zero frequency under the resampling operation. Note the output y (nM,k) is computed as a phase coherent summation of the M output series yr (nM). This phase coherent sum is in fact, a DFT of the M-path outputs, which can be likened to beam-forming the output of the path filters.

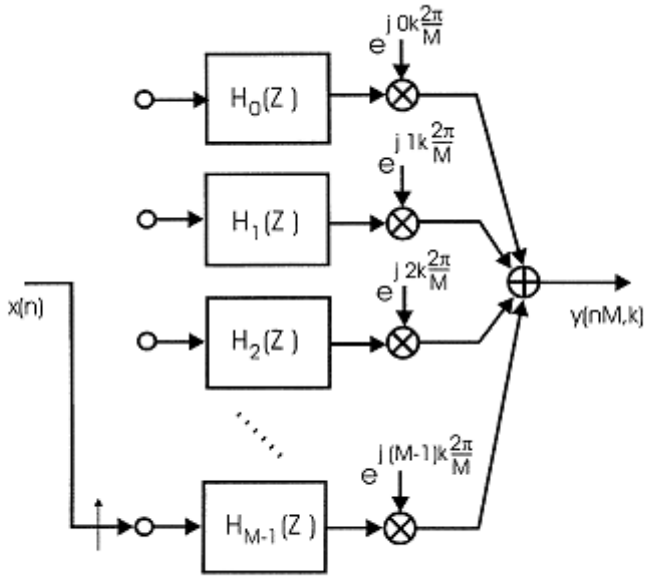


Fig. 24. Resampling M-path down converter.

Fig.19 (ref. Fig 24[4])

$$y(nM,k) = \sum_{r=0}^{M-1} y_r(nM) e^{j\frac{2\pi}{M}rk} \tag{7}$$

The beam-forming perspective offers interesting insight to the operation of the resampled down-converter system we have just examined. The reasoning proceeds as follows: the commutator delivering consecutive samples to the M input ports of the M-path filter performs a down sampling operation. Each port of the M-path filter receives data at one-Mth of the input rate. The down sampling causes the M-to-1 spectral folding, effectively translating the M-multiples of the output sample rate to base band. The alias terms in each path of the M-path filter exhibit unique phase profiles due to their distinct centre frequencies and the time offsets of the different down sampled time series delivered to each port. Each of the aliased centre frequency experiences a phase shift shown in eq-18, equal to the product of its centre frequency and the path time delay.

$$\begin{aligned} \phi(r,k) &= \omega_k \, \Delta T_r \\ &= 2\pi \frac{f_s}{M} k \; rT_s \\ &= 2\pi \frac{f_s}{M} k \; r \frac{1}{f_s} = \frac{2\pi}{M} kr \end{aligned} \tag{8}$$

The phase shifters of the DFT perform phase coherent summation, very much like that performed in narrow band beam forming, extracting from the myriad of aliased time series, the alias with the particular matching phase profile. This phase sensitive summation aligns contributions from the desired alias to realize the processing gain of the coherent sum while the remaining alias terms, which exhibit rotation rates corresponding to the M roots of unity, are destructively cancelled in the summation.

The inputs to the M-path filter are not narrow band, and phase shift alone is insufficient to effect the destructive cancellation over the full bandwidth of the undesired spectral contributions. Continuing with our beam-forming perspective, to successfully separate wideband signals with unique phase profiles due to the input commutator delays, we must perform the equivalent of time-delay beam forming. The M-path filters, obtained by M-to-1 down sampling of the prototype low-pass filter supply the required time delays. The M-path filters are approximations to all-pass filters, exhibiting, over the channel bandwidth, equal ripple approximation to unity gain and the set of linear phase shifts that provide the time delays required for the time delay beam forming task.

Chapter 4

Software Section

4.1 MATLAB: A Brief Review

MATLAB (matrix laboratory) is a multi-paradigm (more than one programming language styles, capabilities, structure and elements) numerical computing environment and fourth-generation programming language. Developed by Math Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++ and Java .

MATLAB Basics:

✓ Variables :

Variables are assigned numerical values by typing the expression directly, example,

Typing: a= 1+2

Creates: a=3

The answer will not be displayed when a semicolon is put at the end on expression a=1+2;

✓ Arithmetic Operators in MATLAB :

+	Addition of input number(s)	c=a+b
-	Subtraction of input number(s)	c=a-b
*	Multiplication of input number(s)	c=a*b
/	Division of input number(s)	c=a/b
^	Power of a number	c=a^b
'	Transpose matrix of a number	c=a'

✓ Predefined Variables and Functions generally used in MATLAB:

i	sqrt (-1)
j	sqrt (-1)

Pi		3.1416
abs	magnitude of a number	c= abs(a)
angle	angle of a complex number, in radian	c=angle(a)
cos	cosine function, assumes argument is in radian	c=cos(a)
sin	sine function, assumes argument is in radian	c=sin(a)
exp	exponential function	c=exp(a)

✓ Matrices:

MATLAB is based on matrix and vector algebra; even scalar are treated as 1x1 matrices. Therefore, vector and matrix operations are as simple as common calculator operations.

Vectors can be defined in two ways:

- The FIRST METHOD is used for arbitrary elements:

  v= [1 3 5 7];

  Creates a 1x4 vector with elements 1, 3, 5 and 7. Note that comas could have been used in place of spaces to separate the elements. Additional elements can be added to the vector.

  v (5) = 8;

  Creates the vector v = [1 3 5 7 8].

  Previously defined vectors can be used to define a new vector.

  Example, with v defined above

  a= [9 10]; b = [v a];

  Creates the vector b = [1 3 5 7 8 9 10]

- The SECOND METHOD is used for creating vectors with equally spaced elements:

  T = 0: 1:10;

Creates a of 1x10 vector with the elements 0, 1, 2, 3…….10. Note that the middle number defines the increments. If only two number are given, then the increment is set to default of 1

K = 0:10;

Creates 1x11 vectors with the elements 1, 1, 2……. 10

There are a number of special matrices that can be defined:

Null matrix m = [ ];

nxm matrix of zeros: m = zeros (n, m);

nxm identity matrix: m = eye (n);

MATLAB Function used in simulation of Polyphase Channelizer [4]

S. no	Function Name	Description	Syntax
1	Rectwin	Designs a FIR rectangular window	b = rectwin (L)
2	Remez	Design of equiripple filters	b = remez (n ,f, a) n-filter order f-frequency at band edges a-amplitude at frequency edges
3	Reshape	Used to reshape the dimension of the vector/matrices	b=reshape(A,m,n) m is no. of row. n is no. of column A is the matrix whose elements are taken column wise
4	Unwrap	Corrects the phase angle of the elements of a vector by adding +/- 2π if the default phase of the elements are greater than or equal to π.	b = unwrap(P)
5	Fft	Calculates DFT of a sequence through fast fourier transform method.	X = fft(x) Calculates DFT of discrete sequence x
6	Fftshift	Shift zero-frequency component to centre of spectrum	Y = fftshift(X)

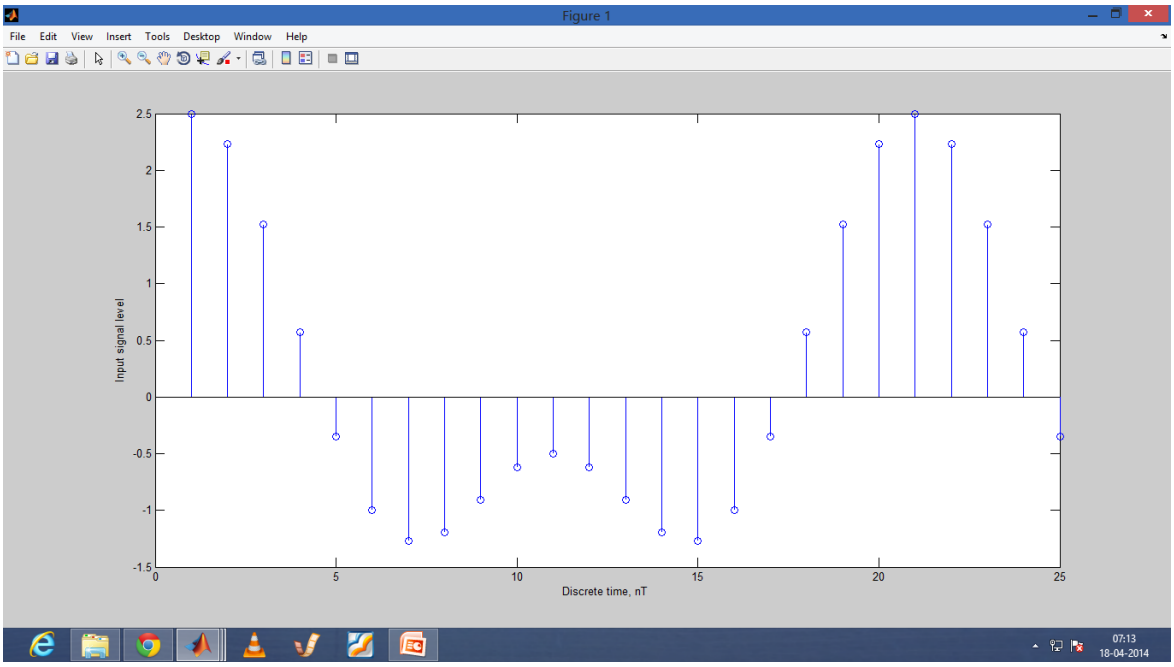
7	Plot	Generates linear plot of vectors and matrices	plot (a , b) Plots a on x-axis and b on y-axis
8	Xlabel,Ylabel,Title	Used to label the plots	xlabel(‘’) ylabel(‘’) title(‘my plot’)
9	Subplot	Plot more than one graph on the screen through partition of screen into an mXn grid	subplot(m, n, p) m-no. of row n-no. of column p-determines the position of the particular graph counting the upper left corner as p=1.
10	Pause	Used to Halt execution temporarily	Pause pause(on) pause(off)
11	Hold	Used to Retain current graph when adding new graphs	Hold hold (on) hold(off)
12	Grid	Used to make your plot easier to read	Grid grid (on)

4.2MATLAB Simulation of Multirate Filters

4.2a Simulation result of basic Multirate Filters

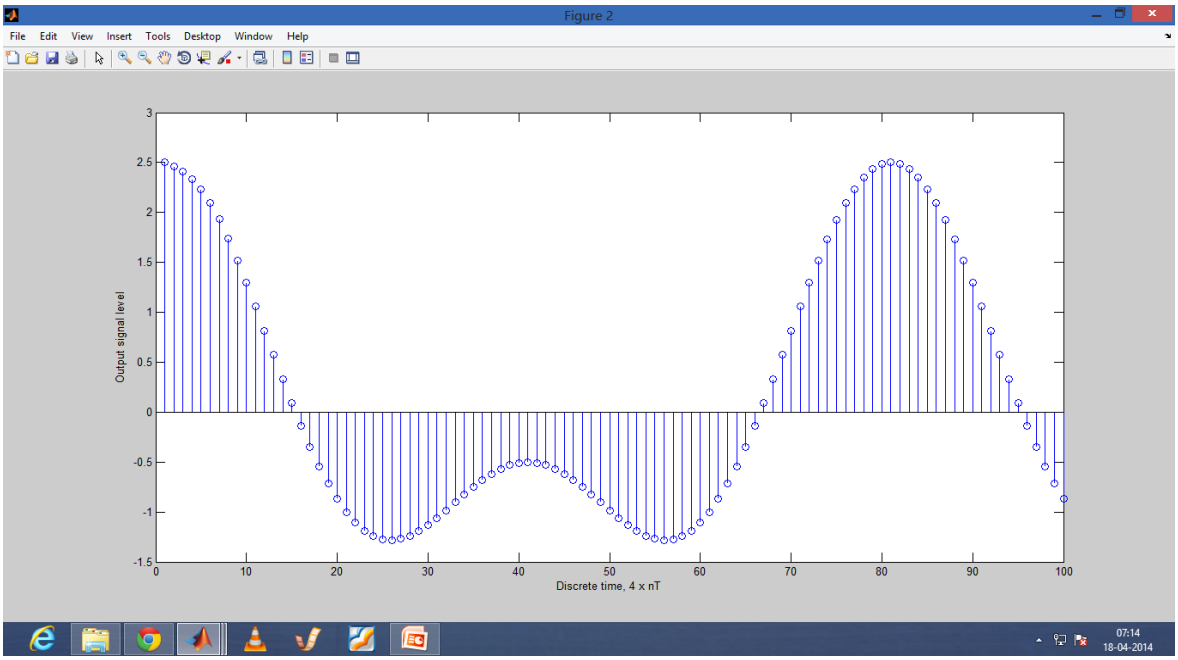
In this section the description of the MATLAB (R2011a) simulation for basic Multirate Filter is demonstrated

Plot 1



. Fig. 19: Discrete Input signal

Plot 2



. Fig. 20:Input Signal Interpolated by a factor of x4

Plot3

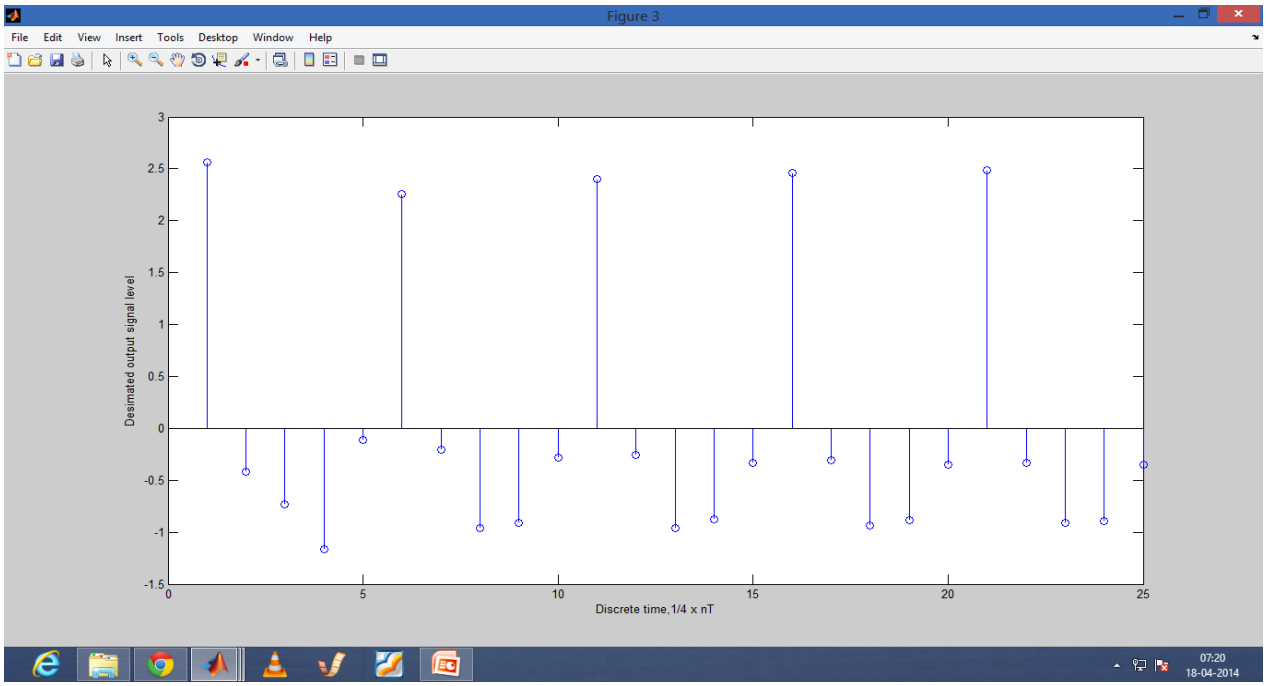


Fig.21:Input discrete Signal decimated by a factor of ¼

Plot 4

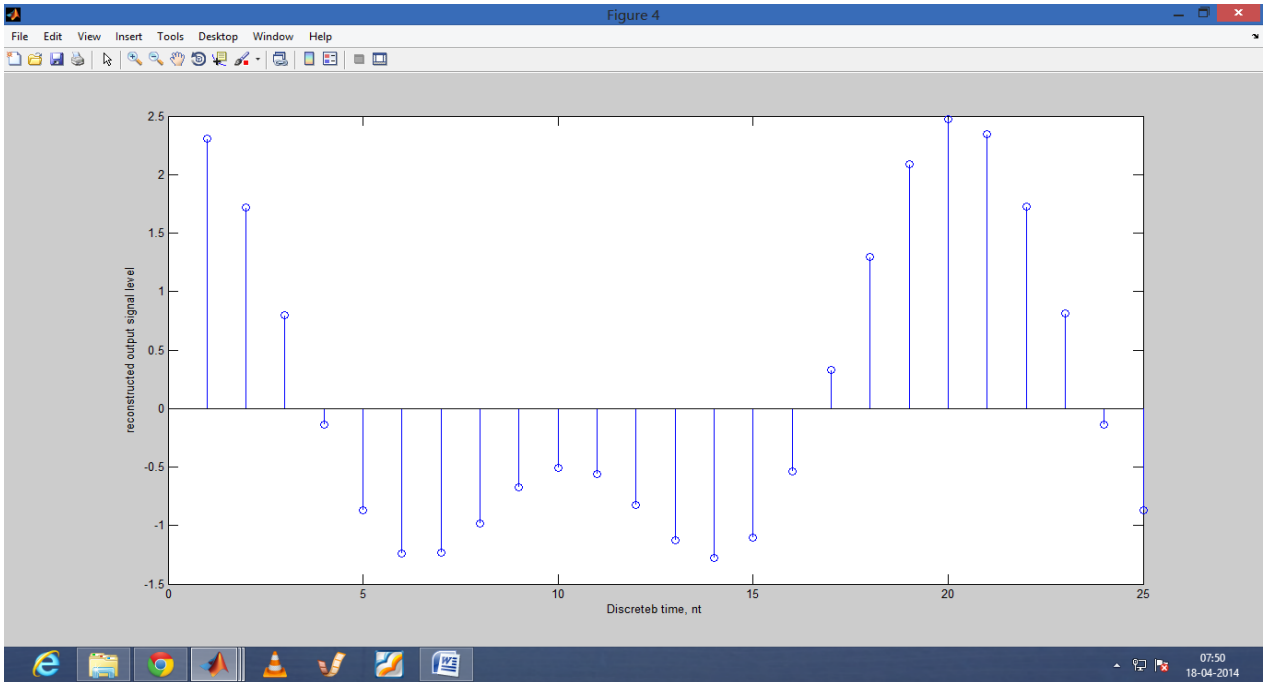


Fig.22:Reconstructed output Signal

4.3MATLAB Simulation of 10-Stage Polyphase Channelizers

4.3b Simulation result of 10-Stage Polyphase Channelizers

In this section the description of the MATLAB (R2011a) simulation result for 10-stage Polyphase Channelizers is demonstrated

Plot 1

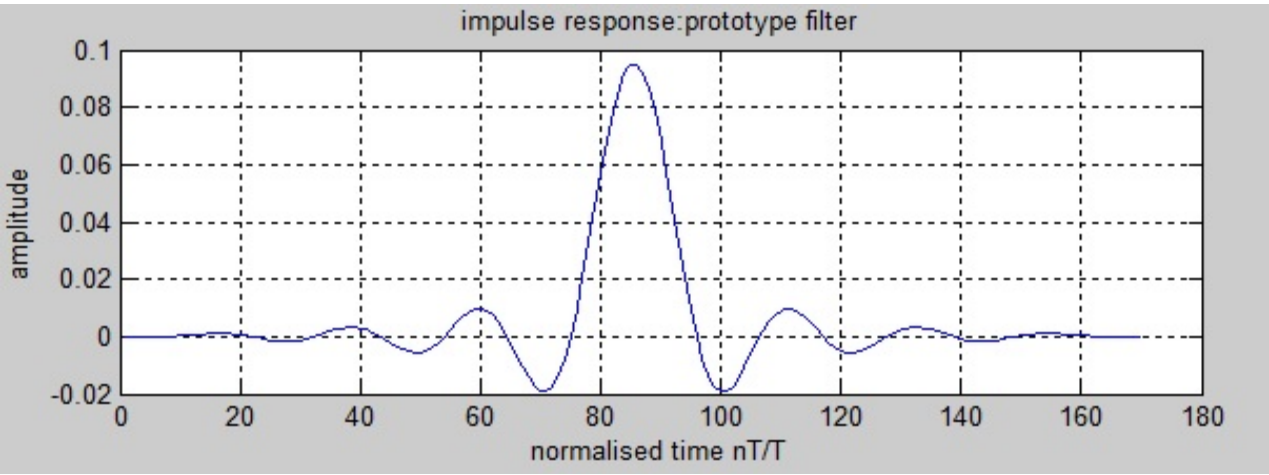


Fig.23: Impulse response of Fir Equiripple Filter

Plot 2

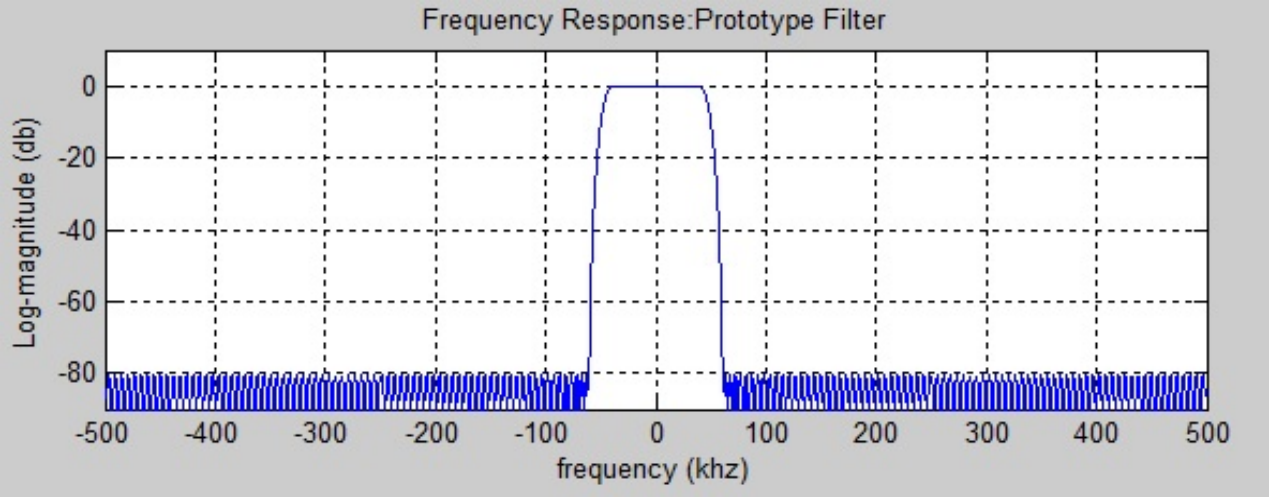


Fig.24:Fourier Transform of Fir Equiripple Filter

Plot 3

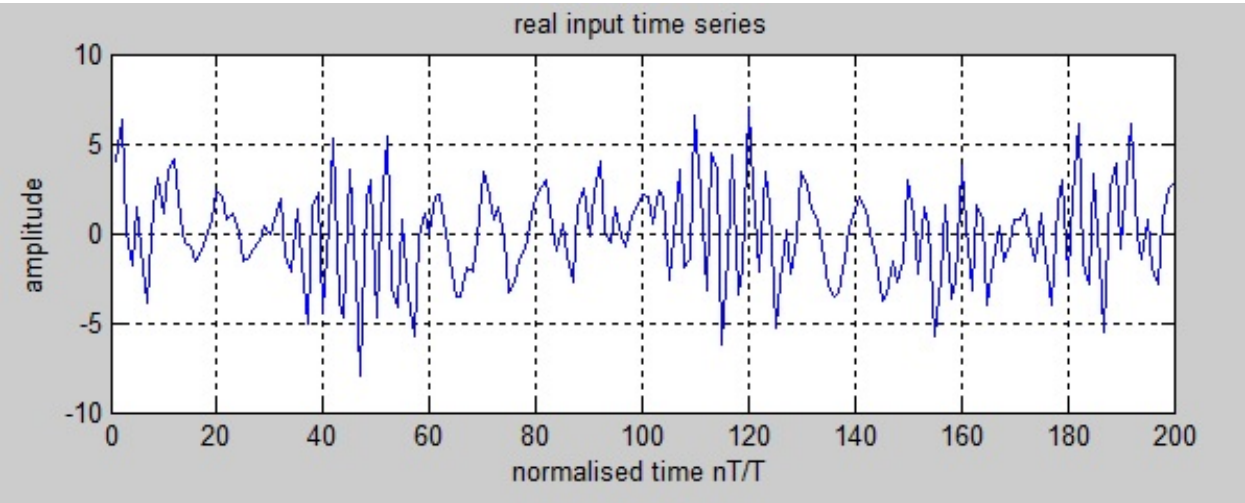


Fig.25:Plot of Input Signal

Plot 4

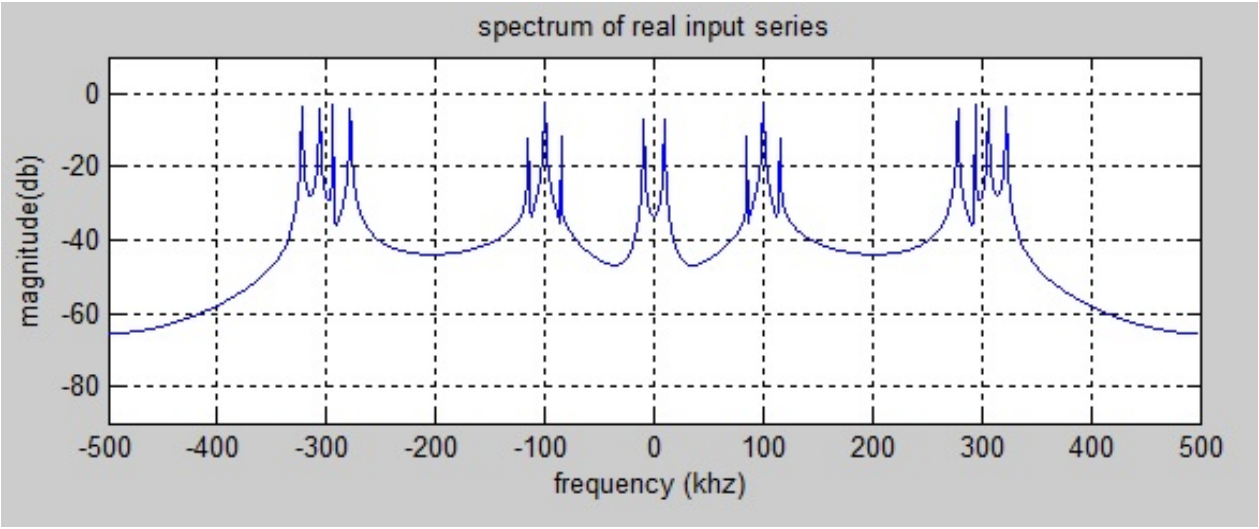


Fig.26:Fourier Transform of Input Signal

Plot 5

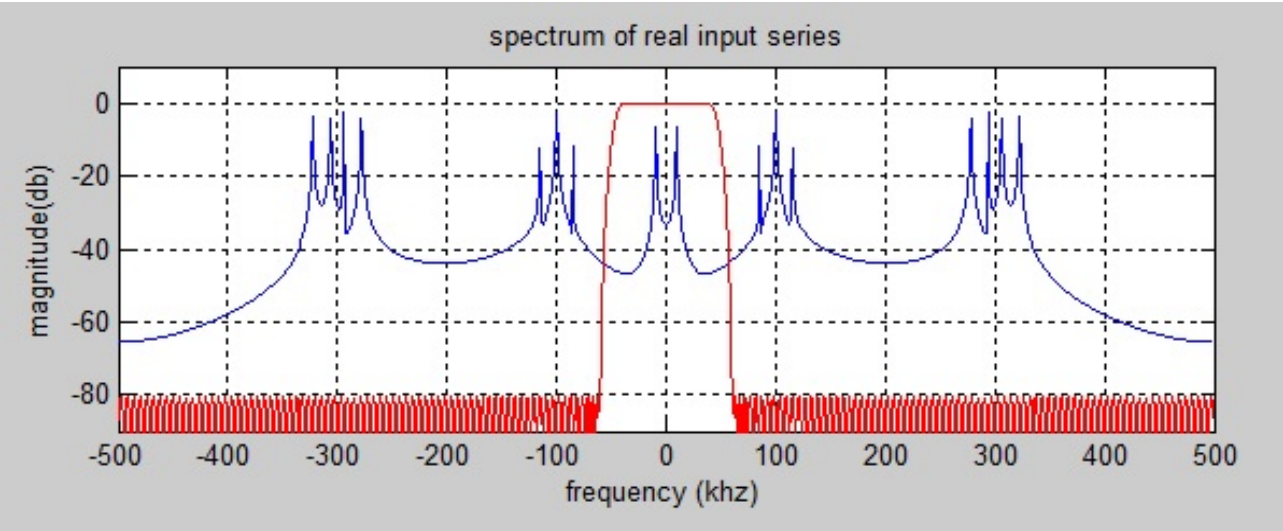


Fig.27:Plot showing Input Signal applied to Equiripple Filter in frequency domain

Plot 6

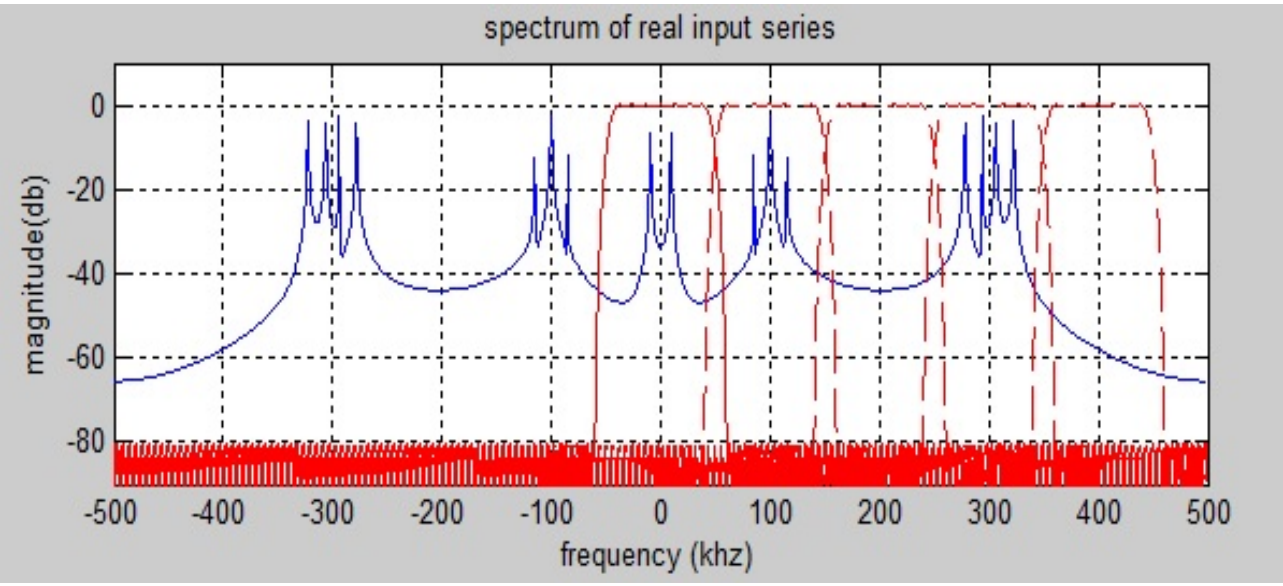


Fig.28:Plot showing Input Signal applied to Quadrature Phase Equiripple Filter in frequency domain (layman view of Polyphase Filter)

Plot 7

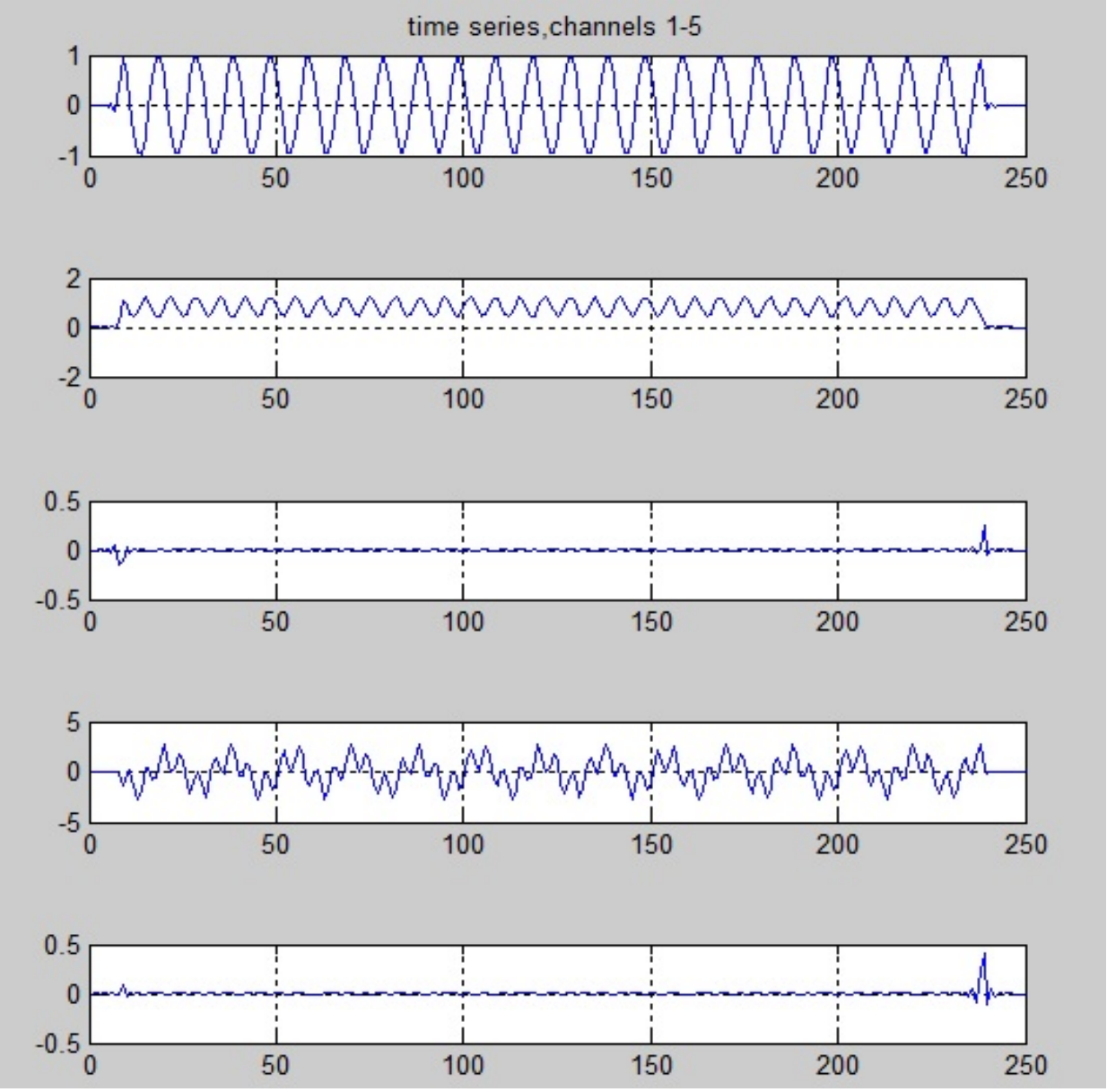


Fig.29:5-Real time Channels of the Polyphase Channelizer

Plot 8

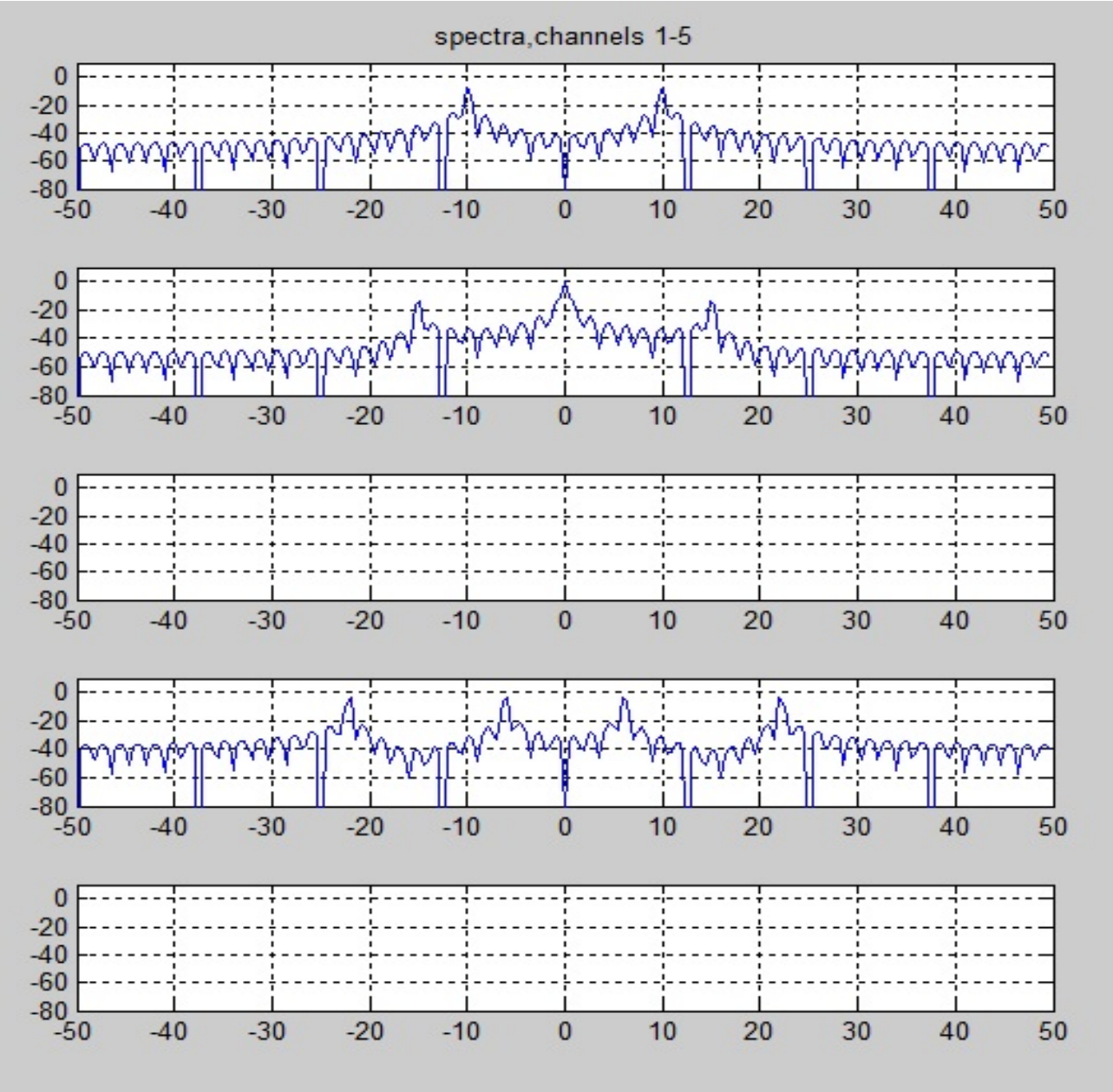


Fig.30:Frequency Series Channels of the Polyphase Channelizer

Plot 9

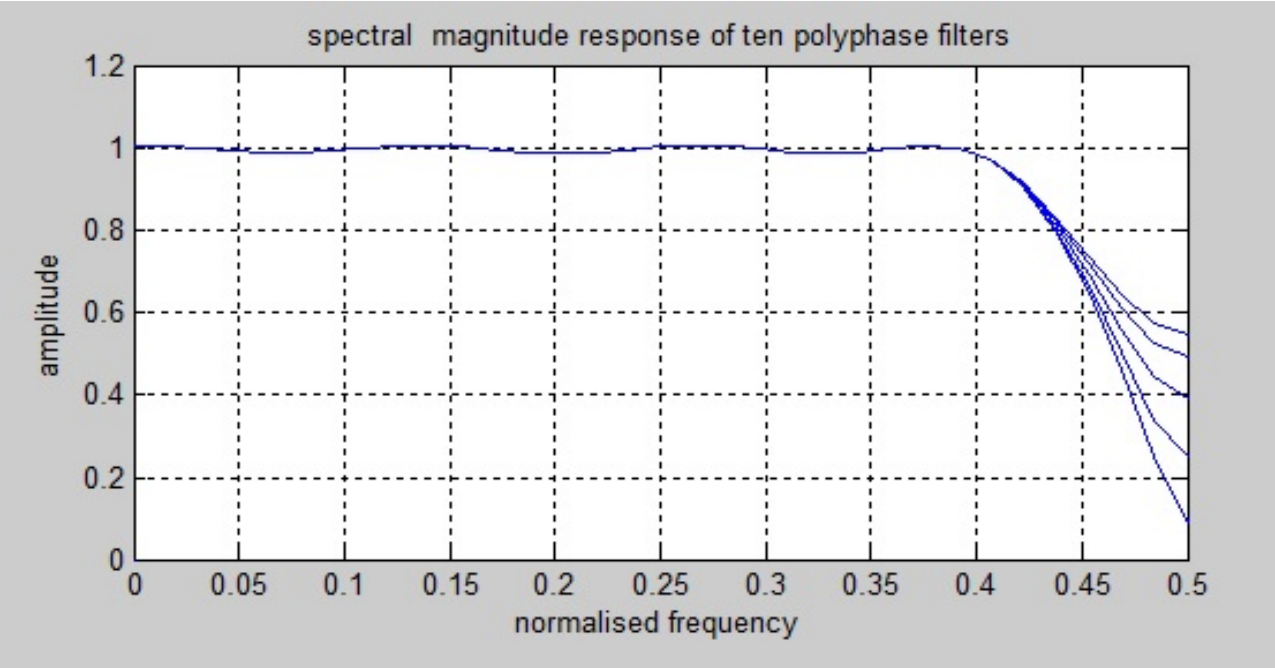


Fig.31:Frequency response of 10 Polyphase Filters

Plot 10

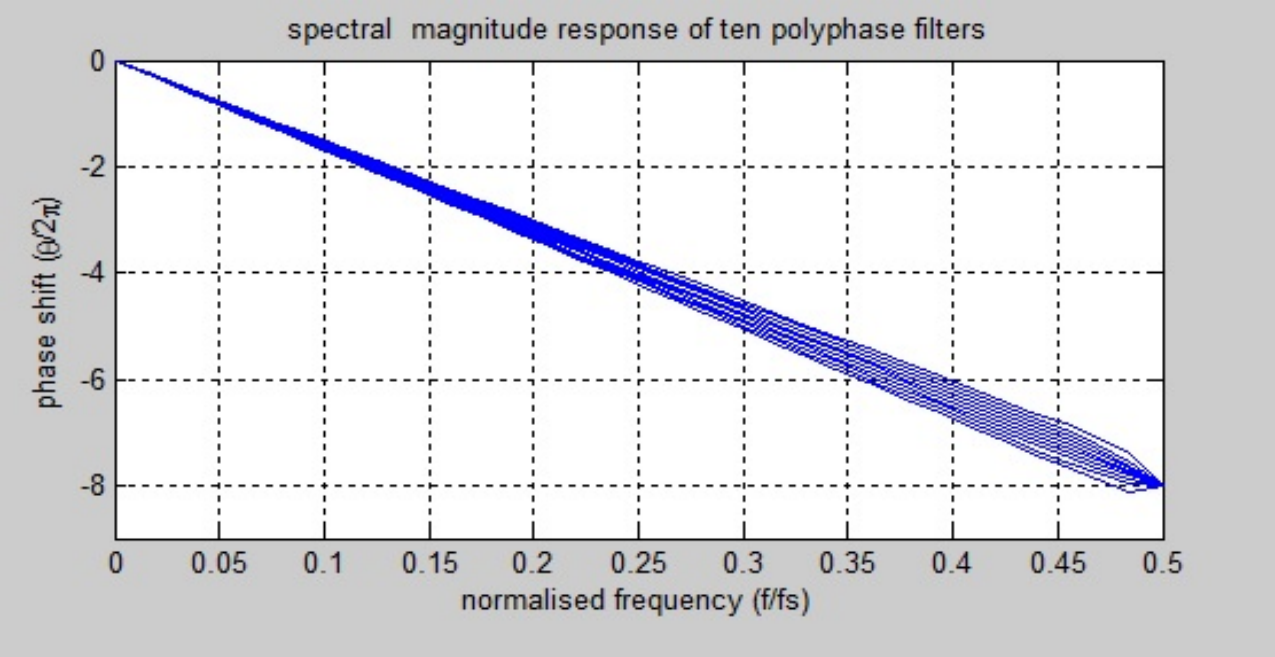


Fig.32:Phase response of 10 Polyphase Filters

Plot 11

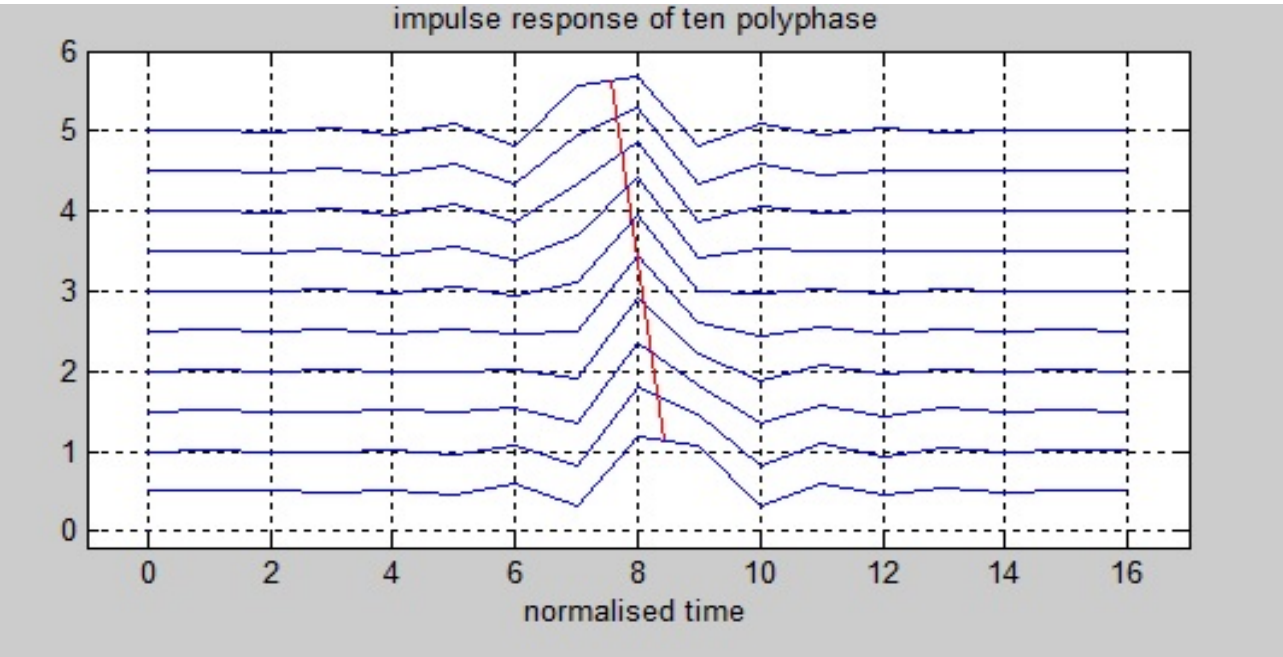


Fig.33:Impulse Response of ten Polyphase Filters

Plot 12

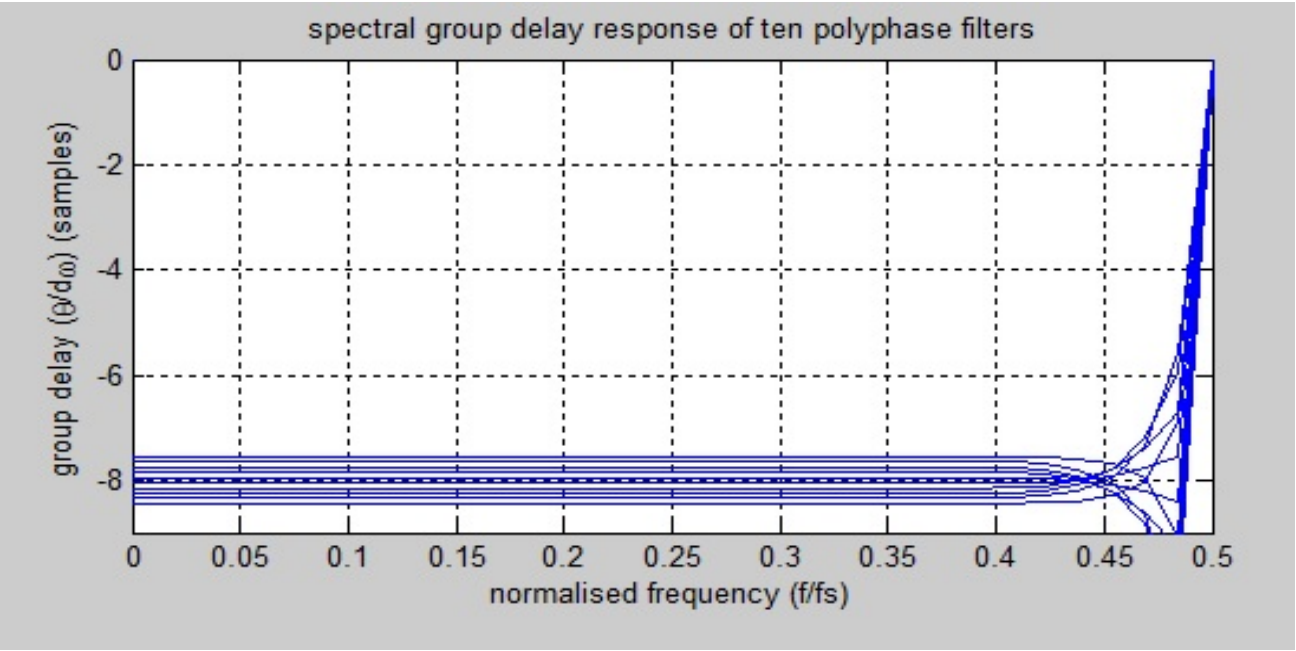


Fig.34: Group delay of ten Polyphase Filters

CHAPTER 5
RESULT AND DISCUSSION

5.1a Result based discussion of basic Multirate filters according to Plot obtained

- Plot 1: Discrete input signal of frequency f=50 hz.
- Plot 2: Discrete input signal interpolated by a factor of 4 by padding zero value samples in between the input samples.
- Plot 3:Discrete input signal decimated by a factor of 4 by removing 3 samples in between the input samples
- Plot 4: Reconstructed discrete input signal by interpolation of the decimated signal.

5.1b Result based discussion on 10 –stage Polyphase channelizer according to Plot obtained

- Plot 1:Impulse response of Fir Equiripple Filter using remez function.
- Plot 2:Frequency response of Fir Equiripple Filter
- Plot 3: Analog input signal of frequency 5400 hz with considering first 200 samples only.
- Plot 4: Frequency response of the input signal
- Plot 5:Input Signal applied to Equiripple Filter in frequency domain
- Plot 6:Input Signal applied to Quadrature Phase Equiripple Filter in frequency domain (layman view of Polyphase Filter)
- Plot 7:Real time Channels of the Polyphase Channelizer
- Plot 8:Frequency Series Channels of the Polyphase Channelizer
- Plot 9:Frequency response of 10 Polyphase Filters
- Plot 10: Phase response of 10 stage Polyphase channelizer

  Phase Delay is the ratio of phase function of input signal to frequency

Ƭp = -ϕ/GƆ ; GƆ :-angular frequency

- Plot 11:Impulse profiles for 10 stage Polyphase channelizer
- Plot 12:Group delay profiles for 10 stage Polyphase channelizer

Group delay Ƭg = -dϕ / dGƆ

Result comparison between existing window and proposed window

Channel no.	Axis point	Amplitude recorded with existing window	Amplitude recorded with proposed window
2.	-10,10	-6,-7	-12,-11
4.	-15,0,15	-18,0,-12	-14,-1,-13
6.	-50,50,	0,0	0,0
8.	-22.5,-7.45,7.45,22.5	-3,-2,-2,2	-2,-4,-4,-3,
10.	-50,50	0,0	0,0

The table represents that the use of rectangular window as FIR filter has better peak amplitude filtering as compared with the used Kaiser FIR window.

CHAPTER 6

FUTURE SCOPE

The descriptive and tutorial derivation of the Polyphase filter has been developed. The filter alone performs the signal processing tasks (down conversion, sample rate change etc.),hence reduces the transceiver manufacturing cost.

Also the demand of Polyphase processing arises in many fields of digital signal processing:

- Digital audio: sampling frequency conversion (32 kHz, 44.1kHz, 48kHz), sharp cut-off of FIR filter.

- Signal processing for digital communications: symbol rate processing, bit rate processing, sample rate processing.

- Speech processing: 3G speech codec (Adaptive Multi Rate), fractional pitch estimation.

- Software Radio telescopes are a new radio astronomy using polyphase filter. GPUs and Multicore processors are promising devices to provide the required processing power.

- Polyphase Filters have been used extensively in high quality audio-compression.

- Gregory W. Wornell "Emerging application of Multirate signal processing and wavelets in digital communication," in IEEE vol.84,no.4 pp. April 1996.

- Ronghua NI, KartikeyMayaram and Terri S. Fiez"A 2.4 GHz hybrid polyphase filter based BFSK receiver with high frequency offset tolerance for wireless sensor networks ," IEEE journal of solid-state circuits,vol.48,no.5 pp. May 2013.

- E. Ifeachor B. Jervis , Digital Signal Processing: A Practical Approach ,ch.-9,"Multirate digital signal processing,"

- Fred Harris, Chris Dick"Digital Receivers and Transmitters Using Polyphase Filter Banks for Wireless Communications"IEEE transactions on microwave theory and techniques,vol.51,no.4

- Ricardo A. Losada "Practical FIR filter Design in MATLAB",March 31,2003

- T. Saramäki and R. Bregovi, "Multirate Systems and Filter Banks," Chapter 2 in *Multirate Systems: Design and Applications* edited by G. Jovanovic-Dolecek. Hershey PA: Idea Group Publishing.

In this section the description of the MATLAB (R2011a) codes for basic Multirate Filter is demonstrated

4.2a. MATLAB code for basic Multirate Filters

```
Fs=1000;
A=1.5;
B=1;
f1=50;
f2=100;
t=0:1/Fs:1;
x=A*cos(2*pi*f1*t)+B*cos(2*pi*f2*t);
stem(x(1:25))
xlabel('Discrete time, nT ')
ylabel('Input signal level')

figure (1)
y=interp(x,4);
stem(y(1:100))
xlabel('Discrete time, 4 x nT')
ylabel('Output signal level')

y1=decimate(x,4);
figure (2)
stem(y1(1:25))
xlabel('Discrete time,1/4 x nT')
ylabel('Desimated output signal level')
```

```
y2=decimate(y,4) ;
figure (3)
stem(y2(1:25))
xlabel ('Discrete time, nt')
ylabel('reconstructed output signal level')
```

In this section the description of the MATLAB (R2011a) codes for 10-stage Polyphase Channelizers is demonatrated.

4.3a. MATLAB code for 10-Stage Polyphase Channelizers

```
functionfilter_ten (flag)
% filter_ten (flag) flag=0 for flat sidelobes, flag=1 for falling sidelobes
hh1=remez(169,[0 40 60 500]/500,[1 1 0 0],[1 100]);
frq=[0 40 60 99 100 149 150 199 200 249 250 299 300 349 350 399 400 449
450 500]/500;
gn= [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
pn= [ 1 100 140 180 220 260 300 340 380 420];
hh2=remez(169,frq,gn,pn);
hh=hh1;
if flag==1
hh=hh2;
end

figure(1)
subplot(2,1,1)
plot(hh)
grid
title('Impulse Response: Prototype Filter')
xlabel('Normalized time nT/T')
ylabel('Amplitude')
subplot(2,1,2)
plot((-0.51/1024:.5-1/1024)*1000,fftshift(20*log10(0.000001+abs (fft(hh,1024)))))
```

```
grid
axis([-500 500 -90 10])
title('Frequency Response: Prototype Filter')
xlabel('Frequency (kHz)')
ylabel('Log-Magnitude (dB)')
pause
x1=1*cos(2*pi*(0:2299)*10/1000);
x2=2+cos(2*pi*(0:2299)*15/1000);
x2=x2.*cos(2*pi*(0:2299)*100/1000);
x3=(3*cos(2*pi*(0:2299)*22/1000)+5*sin(2*pi*(0:2299)*6/1000));
x3=x3.*sin(2*pi*(0:2299)*300/1000);
xx=x1+x2+x3;
xx=[xx zeros(1,200)];

figure(2)
subplot(2,1,1)
plot(xx(1:200));
grid
title('Real Input Time Series')
xlabel('Normalized time nT/T')
ylabel('Amplitude')
subplot(2,1,2)
ww=rectwin (1024)';
ww=ww/sum(ww);
plot((-0.5:1/1024:.5-1/1024)*1000,fftshift(20*log10(abs(fft(xx(1:1024).*ww,1024)))))
grid
axis([-500 500 -90 10])
title('Spectrum of Real Input Series')
xlabel('Frequency (kHz)')
```

```
ylabel('Log-Magnitude (dB)')
pause
hold

plot((-0.5:1/1024:.5-1/1024)*1000,fftshif(20log10(0.00001+abs(fft(hh,1024)))),'r')
pause

gg1=hh.*exp(j*2*pi*(-84.5:84.5)*100/1000);
plot((-0.5:1/1024:.5-1/1024)*1000,fftshift(20*log10(abs(fft(gg1,1024)))),'r--')
gg2=hh.*exp(j*2*pi*(-84.5:84.5)*200/1000);
plot((-0.5:1/1024:.5-1/1024)*1000,fftshift(20*log10(abs(fft(gg2,1024)))),'r--')
gg3=hh.*exp(j*2*pi*(-84.5:84.5)*300/1000);
plot((-0.5:1/1024:.5-1/1024)*1000,fftshift(20*log10(abs(fft(gg3,1024)))),'r--')
gg4=hh.*exp(j*2*pi*(-84.5:84.5)*400/1000);
plot((-0.5:1/1024:.5-1/1024)*1000,fftshift(20*log10(abs(fft(gg4,1024)))),'r--')
hold
pause

hh2=reshape(hh, 10, 17);
reg=zeros(10,17);
n2=1;
fornn=1:10:2500
reg(:,2:17)=reg(:,1:16);
reg(:,1)=flipud(xx(nn:nn+9)');
for mm=1:10
vv(mm)=reg(mm,:)*hh2(mm,:)';
end
yy(:,n2)=fft(vv)';
n2=n2+1;
```

```
end

figure(3)
subplot(5,2,1)
plot(real(yy(1,:)))
grid
title('Time Series, Channels 1-5')
rr=axis;
rr(2)=250;
axis(rr);
subplot(5,2,2)
ww=rectwin(200)';
ww=ww/sum(ww);
plot((-0.5:1/256:.5-1/256)*100,fftshift(20*log10(abs(fft(yy(1,20:219).*ww,256)))))
axis([-50 50 -80 10])
grid
title('Spectra, Channels 1-5')

subplot(5,2,3)
plot(real(yy(2, :)))
grid
rr=axis;
rr(2)=250;
axis(rr);
subplot(5,2,4)
plot((-0.5:1/256:.5-1/256)*100,fftshift(20*log10(abs(fft(yy(2,20:219).*ww,256)))))
axis([-50 50 -80 10])
grid
```

```
subplot(5,2,5)
plot(real(yy(3,:)))
grid
rr=axis;
rr(2)=250;
axis(rr);
subplot(5,2,6)
plot((-0.5:1/256:.5-1/256)*100,fftshift(20*log10(abs(fft(yy(3,20:219).*ww,256)))))
axis([-50 50 -80 10])
grid

subplot(5,2,7)
plot(real(yy(4,:)))
grid
rr=axis;
rr(2)=250;
axis(rr);
subplot(5,2,8)
plot((-0.5:1/256:.5-1/256)*100,fftshift(20*log10(abs(fft(yy(4,20:219).*ww,256)))))
axis([-50 50 -80 10])
grid

subplot(5,2,9)
plot(real(yy(5,:)))
grid
rr=axis;
rr(2)=250;
axis(rr);
subplot(5,2,10)
```

```
plot((-0.5:1/256:.5-1/256)*100,fftshift(20*log10(abs(fft(yy(5,20:219).*ww,256)))))
axis([-50 50 -80 10])
grid
pause

figure(4)

subplot(2,2,1)
plot(0,0)
hold
for mm=1:10
plot(0:1/64:1-1/64,(10*abs(fft(hh2(mm,:),64))))
end
hold
grid
title('Spectral Magnitude Response of Ten Polyphase Filters')
xlabel('Normalized Frequency')
ylabel('Amplitude')
axis([0 0.5 0.0 1.2])

subplot (2,2,3)
plot(0,0)
hold
for mm=1:10
plot(0:1:16,0.5*mm+10*hh2(mm,:))
end
plot([7.55 8.45],[5.628 1.128],'r')
hold
grid
```

```
title('Impulse Response of Ten Polyphase Filters')
xlabel('Normalized Time')
axis([-1 17 -0.2 6])

subplot(2,2,2)
plot(0,0)
hold
for mm=1:10
plot(0:1/64:1-1/64,unwrap ((angle((fft(hh2(mm,:),64)))))/pi)
end
hold
grid
title('Spectral Phase Response of Ten Polyphase Filters')
axis([0 0.5 -9 0])
xlabel('Normalized Frequency (f/fs)')
ylabel('Phase Shift (\theta/2\pi)')

subplot(2,2,4)
plot(0,0)
hold
for mm=1:10
vv=unwrap(angle(fftshift(fft(hh2(mm,:),64))))/(2*pi);
vv2=64*filter([1 -1],1, vv);
plot(0:1/64:1-1/64,fftshift(vv2))
end
hold
grid
title('Spectral Group Delay Response of Ten Polyphase Filters')
xlabel('Normalized Frequency (f/fs)')
```

```
ylabel('Group Delay (d\theta/d\omega) (Samples)')
axis([0 0.5 -9 0])
```