DESIGN AND IMPLEMENTATION OF FIR DIGITAL FILTERS WITH VARIABLE FREQUENCY CHARACTERISTICS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

MIDDLE EAST TECHNICAL UNIVERSITY

BY

HATİCE PİŞKİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2005

Approval of the Graduate School of Natural and Applied Sciences

_____

Prof. Dr.Canan ÖZGEN

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science

_____

Prof. Dr.İsmet ERKMEN

Head of the Department

This is to certify that we have read this thesis and in our opinion it is fully adequate, in scope and quality, as a thesis for a degree of Master of Science.

_____

Prof. Dr. Zafer ÜNVER

Supervisor

Examining Committee Members

Prof. Dr. Kemal LEBLEBİCİOĞLU (METU-EE) _____

Prof. Dr. Zafer ÜNVER (METU-EE) _____

Assoc. Prof. Dr. Engin TUNCER (METU-EE) _____

Assoc. Prof. Dr. Tolga ÇİLOĞLU (METU-EE) _____

Ali TÜRKAY, M. Sc. in EE (ASELSAN) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Hatice PİŞKİN

Signature :

ABSTRACT

DESIGN AND IMPLEMENTATION OF FIR DIGITAL FILTERS WITH VARIABLE FREQUENCY CHARACTERISTICS

PİŞKİN, Hatice

M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Zafer ÜNVER

December 2005, 85 pages

Variable digital filters (VDF) find many application areas in communication, audio, speech and image processing. This thesis analyzes design and implementation of FIR digital filters with variable frequency characteristics and introduces two design methods. The design and implementation of the proposed methods are realized on Matlab software program. Various filter design examples and comparisons are also outlined.

One of the major application areas of VDFs is software defined radio (SDR). The interpolation problem on sample rate converter (SRC) unit of the SDR is solved by using these filters. Realizations of VDFs on SRC are outlined and described. Simulations on Simulink and a specific hardware are examined.

Keywords: Variable digital filter, least squares, software defined radio, sample rate converter, interpolation.

ÖZ

DEĞİŞKEN FREKANS KARAKTERİSTİKLİ FIR FİLTRE TASARIMI VE UYGULAMASI

PİŞKİN, Hatice
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Zafer ÜNVER

Aralık 2005, 85 Sayfa

Değişken frekans karakteristikli sayısal filtreler haberleşme, ses ve görüntü işleme çalışmalarında uygulama alanları bulmaktadır. Bu tez çalışmasında değişken frekans karakteristikli FIR filtre tasarımı ve uygulaması çözümlenmiş ve gerçeklenmiştir. Filtre tasarımı iki farklı yöntem kullanılarak Matlab yazılımı üzerinde uygulanmıştır. Çeşitli filtre tasarımları örnek olarak verilmiş ve karşılaştırılmıştır.

Değişken frekans karakteristikli filtrelerin en önemli uygulama alanlarından biri yazılım tanımlı radyodur. Yazılım tanımlı radyonun örnekleme hızı çeviricisi biriminde görülen enterpolasyon problemi bu tür filtreler kullanılarak çözülebilmektedir. Değişken frekans karakteristikli filtrelerin, örnekleme hızı çeviricisindeki kullanımı tanımlanmıştır. Simulink programı üzerindeki benzetimler ve belirli bir donanım üzerinde elde edilen sonuçlar incelenmiştir.

Anahtar Kelimeler: Değişken sayısal süzgeç, en az kareler, yazılım tanımlı radyo, örnekleme hızı çeviricisi, enterpolasyon.

ACKNOWLEDGEMENTS

I appreciate Prof. Dr.Zafer Ünver for his guidance, support and valuable contributions throughout the study.

I express my deepest gratitude to my mother, my father and my brother for their trust and encouragements throughout my education life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Variable digital filters are digital filters with controllable spectral characteristics such as variable cutoff frequency response, adjustable pass band width, controllable fractional delay, etc. They have been found useful in various signal processing applications such as telecommunication, audio, speech, and image processing, where the frequency response of digital filters are required to be adjustable. Generally, variable digital filters can be classified into two main categories. The first one includes the digital filters with variable magnitude responses such variable filters can be efficiently utilized for implementing variable filter banks for audio signal processing. The second category includes the digital filters with variable phase response especially those with variable fractional-delay responses, which are widely useful in the applications such as timing adjustment in digital receiver, speech coding and synthesis, reduction of channel timing-error effects in time-interleaved analog-to-digital converters, and other applications, whenever new sample values at arbitrary time instants between the existing discrete-time samples need to be interpolated. [6], [7], [10], [11], [12], [13].

Methods for designing variable digital filters can be broadly classified into two categories: transformation [9], and spectral parameter approximation [8] methods. In the former, a prototype filter with certain design characteristics is first designed. Certain transformations such as the all pass transformation method is then applied to the prototype filter to obtain the final variable digital filter. In general, transformation method is applicable to variable digital filters with variable cutoff frequencies; it is not applicable for variable characteristics such as variable fractional delay. Additionally, if the filter is infinite impulse response type, these kinds of transformations may produce delay-free loops in the filter structure, which cannot be implemented digitally.

The spectral parameter method is more general in the sense it assumes that either the impulse responses or the poles and zeros of the filters are polynomials of certain spectral parameters. Typically, the transfer function of a variable digital filter contains a number of spectral parameters that can be used to tune the frequency response of the digital filter. Thus the main objective in the design of a variable digital filter is to find a parameterized transfer function which in a certain sense best approximates a given set of frequency response characteristics that vary with the parameters in a desired manner. The spectral parameter method was proposed by Zaraour and Fahmy [8] where the poles and zeros of an infinite impulse response filter are assumed to be polynomials of the spectral or tuning parameters. The proposed method utilizes a nonlinear optimization method to design a set of digital filters corresponding to the sampled variable frequency domain characteristics. Then the coefficients of a general variable digital filter are assumed to be multidimensional polynomials of spectral parameters such as cutoff frequency, pass band width and transition band width, which define the frequency domain characteristics, and the optimal multidimensional coefficient polynomials are found to best fit the fixed filter coefficients determined before by using a nonlinear optimization method. In practical applications, once the spectral parameters are given, the coefficients of the variable filter can be readily calculated from the determined multidimensional polynomials. However, the technique has two drawbacks; since many fixed constant digital filters have to be designed first by using a nonlinear optimization method, and then a lot of

multidimensional polynomials have to be approximated, the technique is not computationally efficient. Moreover, since the denominator coefficients of the designed variable digital filters are also multidimensional polynomials of a set of spectral parameters and are varied during the course of signal processing, the stability of variable filters cannot be guaranteed.

Most of the works on variable digital filters reported after Zaraour and Fahmy's method focused on the design of infinite impulse response variable digital filters and the methods for guaranteeing their stability. More recently the design of variable finite impulse response digital filters has received considerable attention due to their simple design procedure and good filtering performance. Among the design methods developed for finite impulse response variable digital filters defined by spectral parameters, the Lagrange multiplier method seems to be the most attractive due to its simplicity, but the frequency response of the resulting Lagrange interpolator cannot be uniformly balanced in the entire frequency band, i.e., the frequency response in the low-frequency region is superior to that in the high-frequency region. Thus it is difficult to achieve a satisfactory design with low filter orders in the whole frequency band by using the Lagrange multiplier method. To solve this problem, a general technique using the weighted least squares method is proposed, which can yield a more satisfactory design with lower order than the Lagrange interpolator. Design of finite impulse response variable digital filters is generalized to a linear combination of basis functions. Optimal least squares solution can also be obtained by solving a system of linear equations. The resulting variable digital filters can be implemented using the Farrow [14] structure.

Another direct method for designing finite impulse response variable digital filters is the weighted least squares method, which assumes that the variable filter coefficients are multidimensional polynomials of the spectral parameters, and then the optimal polynomial coefficients are determined through minimizing the weighted squared-error between the desired and actual variable frequency responses. However, the computational complexity of the weighted least squares design is rather heavy

especially when the number of the spectral parameters increases. Moreover, the discretization of frequency band is required during the computations.

If only the pass band linear-phase characteristics is required, optimal design problem can be formulated as a semi definite programming problem, which is a versatile framework for designing finite and infinite impulse response filters. Although the design time required for semi definite programming is more than that of the least squares case, the method is considerable since it offers lower system delay in the pass band.

As an application area of variable digital filters software defined radio takes more attention nowadays. Software defined radio is a general hardware/software platform for supporting inter-communication systems. Sample rate converter units are employed in software defined radios to isolate the desired user's channel from the signal spectrum and convert to an appropriate sampling rate for further processing. Sample rate converter unit can be realized by variable digital filters which reduces the implementation complexity of the software defined radio system.

1.2 Studies Carried Out in This Thesis

Throughout this thesis previous studies on design and implementation of finite and infinite impulse response digital filters are examined in detail. After gaining knowledge by this literature survey, various methods for the design of variable finite impulse response are implemented. These methods are investigated and compared for various design criteria. Illustrative examples for various filter types are studied and corresponding frequency responses are given.

Implementation areas for variable digital filters are investigated; and in particular implementation on software defined radio is studied.

Hardware implementation of variable digital filters on software defined radio is investigated. Comparison between the traditional, and the proposed hardware structures is made.

1.3 Outline of the Thesis

Introduction for this thesis is given within this chapter (Chapter1) in which the general information about variable digital filters, previous studies and the objectives for this thesis are stated.

In Chapter 2, the variable finite impulse response digital filter design with direct integration method is described. Mathematical derivations and illustrative examples are given.

In Chapter 3, the variable finite impulse response digital filter design with singular value decomposition based vector array decomposition method is described. Mathematical derivations and illustrative examples are given.

In Chapter 4, the variable digital filter application on software defined radio application is considered. A brief information about software defined radio is given. Variable filter application on the software defined radio is described. The hardware implementation on software defined radio is investigated. The advantage of usage of the variable digital filters on software defined radio is outlined and the comparison of the traditional and the proposed methods is given.

Conclusion is given in Chapter 5.

CHAPTER 2

VARIABLE FIR FILTER DESIGN WITH DIRECT INTEGRATION METHOD

2.1 General

In this chapter variable finite impulse response (FIR) filter design by direct integration method is described. Mathematical derivations for this method are outlined. Illustrative examples for variable low pass, band pass and fractional delay filters and design results for these types of filters are given.

2.2 Design Method

The parameters that used to change spectral, frequency, characteristics of the filters are called spectral parameters. In spectral parameter method, the impulse response or the poles and the zeros of the variable digital filters (VDF) are assumed to be a polynomial function of the tuning parameters. Since direct tuning of the poles and zeros yields undesirable transient response during tuning, tuning of impulse response will be considered.

The impulse response of the variable FIR digital filter h(n,**Φ**) is assumed to be a linear combination of functions $\psi_m(\mathbf{\Phi})$ of the spectral parameters **Φ**,

$$h(n,\Phi) = \sum_{m=0}^{M-1} c_m[n]\ \psi_m(\mathbf{\Phi})$$

(2.1-a)

where

$$\psi_m(\mathbf{\Phi}) = a + b\phi + c\phi^2 + \ldots$$

(2.1-b)

$c_m[n]$'s are the coefficients of expansion. The design objective is to determine $c_m[n]$ given $\psi_m(\mathbf{\Phi})$ so that the frequency response h(n,**Φ**) will approximate the desired frequency responses as a function of **Φ**.

The z-transformation of the specified variable digital FIR filter of length N is,

$$H(z,\phi) = \sum_{n=0}^{N-1} h(n,\phi) z^{-n} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} c_m[n] \psi_m(\phi) z^{-n}$$

(2.2)

Interchanging the order of summations, the above equation can be rewritten as

$$H(z,\phi) = \sum_{m=0}^{M-1} C_m(z).\psi_m(\phi)$$

(2.3-a)

where

$$C_m(z) = \sum^{N-1} c_m[n] z^{-n}$$

(2.3-b)

(2.3-a) can be implemented by the structure shown in Figure 2.1.

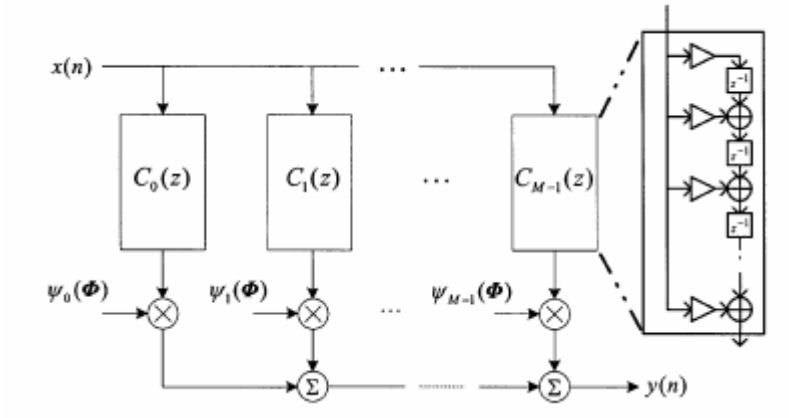

Figure 2.1 Proposed variable digital filter structure.

This structure can be viewed as a generalization of the Farrow structure for implementing a fractional delay digital filter where h(n,**Φ**) is approximated by a polynomial in the delay parameter $\mathbf{\Phi} = \phi$, that is $\psi_m(\mathbf{\Phi}) = \phi^m$.

If the desired frequency response is $H_I(e^{jw}, \mathbf{\Phi})$, then the approximation error is

$$E(w,\phi) = H_I(e^{jw},\phi) - \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} c_m[n]\psi_m(\phi)e^{-jnw} \qquad \textbf{(2.4)}$$

The L_2 norm of E(w, **Φ**) is given by

$$E = \int_{\phi_S}\int_{\Omega_S} W(e^{jw},\mathbf{\Phi}).|E(w,\mathbf{\Phi})|^2 \, dw d\mathbf{\Phi} \qquad \textbf{(2.5)}$$

where $W(e^{jw}, \mathbf{\Phi})$ is a positive weighting function used to control the amount of approximation error in frequency and tuning spaces. Ω_S is the frequency support over which $H_I(e^{jw}, \mathbf{\Phi})$ is to be approximated, and $\mathbf{\Phi}_S$ is the parameter space over which the spectral parameter vector $\mathbf{\Phi}$ is to be varied.

Letting $k = n + Nm$ and $z = e^{jw}$, the desired frequency response takes the form

$$H(e^{jw}, \phi) = \sum_{k=0}^{NM-1} a_k \phi_k(w, \phi)$$

(2.6)

where $a_k = c_m[n]$ and $\Phi_k(w, \mathbf{\Phi}) = \psi_m(\mathbf{\Phi})\, e^{-jnw}$.

Substittuting (2.) into (2.5) and simplifying gives

$$E = a^T Q a - 2b^T a + c$$

(2.7)

where

$$a = \begin{bmatrix} a_0 & a_1 & \dots & a_{NM-1} \end{bmatrix}^T$$

$$b = \begin{bmatrix} b_0 & b_1 & \dots & b_{NM-1} \end{bmatrix}^T$$

$$[Q]_{ij} = \int_{\phi_S} \int_{\Omega_S} W(e^{jw}, \phi).\phi_i(w, \phi)\overline{\phi_j(w, \phi)}\, dw d\phi$$

$$[b]_k = \int_{\phi_S} \int_{\Omega_S} W(e^{jw}, \phi).\mathrm{Re}\{H_I(e^{jw}, \phi).\overline{\phi_k(w, \phi)}\}\, dw d\phi$$

$$c = \int_{\phi_S} \int_{\Omega_S} W(e^{jw}, \phi).|H_I(e^{jw}, \phi)|^2\, dw d\phi$$

(2.8)

Differentiating with respect to 'a' and setting derivatives to zero, one gets the following system of linear equations follows as

$$Q.a_{LS} = b \tag{2.9-a}$$

then the optimal least squares solution, a_{LS}

$$a_{LS} = Q^{-1}b \tag{2.9-b}$$

2.3 Illustrative Examples

2.3.1 Tunable Linear-Phase FIR Low-Pass Filter

As an illustrative example, the design of variable cutoff FIR digital filter design is worth considering. Letting the passband cutoff frequency be denoted by w_p and the stopband cutoff frequency be denoted by w_s, assuming the passband and stopband cutoff frequencies vary linearly with spectral parameter $\mathbf{\Phi}=\phi$, the variable passband and stopband cutoff frequencies could be defined as:

$$\begin{aligned} w_p(\phi) &= \phi(w_{p2} - w_{p1}) + w_{p1} \\ w_s(\phi) &= \phi(w_{s2} - w_{s1}) + w_{s1} \\ \mathbf{\Phi}_s &: \phi \in [0,1] \end{aligned} \tag{2.10}$$

Frequency support of the filter is defined as combination of passband and stopband regions, which are defined by the spectral parameter,

$$\begin{aligned} \Omega_S &= \Omega_p + \Omega_s \\ \Omega_p &= \{w : w \in (0, w_p(\phi))\} \\ \Omega_s &= \{w : w \in (w_s(\phi), \pi)\} \end{aligned}$$

(2.11)

The desired impulse response is denoted by $H_I(e^{jw}, \phi)$ and defined as:

$$H_I(e^{jw}, \phi) = \begin{cases} e^{-j\tau w}, |w| \le w_p(\phi) \\ 0, w_S(\phi) \le |w| \le \pi \end{cases}$$

(2.12)

where τ is the group delay which is assumed to be constant and equal to (N-1)/2 where N is the order of the subfilters.

$h(n, \phi)$ is approximated by a polynomial; and the function $\psi_m(\phi)$ is given by ϕ^m.

The weighting function is defined as

$$W(e^{jw}, \Phi) = \begin{cases} K_p, w \in S_p \\ K_s, w \in S_s \end{cases}$$

(2.13)

2.3.1.1 Mathematical Derivations

Letting $\mathbf{\Phi}_i(w, \phi) = \psi_p(\phi) e^{-jnw} = \phi^p e^{-jkw}$ and i = k + Np, j = n + Nm, where N is the order of the subfilters

$$\begin{aligned} [Q_{ij}] &= \int_{\Phi_S} \int_{\Omega_S} W(e^{jw}, \mathbf{\Phi}).\Phi_i(w, \mathbf{\Phi}) \overline{\Phi_j(w, \mathbf{\Phi})} dw d\mathbf{\Phi} \\ &= \int_{\Phi_S} \phi^{p+m} \int_{\Omega_S} W(e^{jw}, \phi)(\cos((n-k)w) + j\sin((n-k)w)) dw d\phi \end{aligned}$$

(2.14)

By considering a symmetrical integration interval since sinus function odd, the sinus term will disappear in the equation. Moreover, since cosines term is an even function, the above equation can be rearranged as follows

$$\begin{aligned}\left[Q_{ij}\right] &= 2\int_{\Phi_S}\phi^{p+m}\left(\int_{0}^{w_p(\phi)} K_p\cos((n-k)w)dwd\phi + \int_{w_s(\phi)}^{\pi} K_s\cos((n-k)w)dw\right)d\phi \\ &= 2\int_{\Phi_S}\phi^{p+m}\left(\frac{K_p}{(n-k)}\sin((n-k)w_p(\phi))dwd\phi - \frac{K_s}{(n-k)}\sin((n-k)w_s(\phi))\right)d\phi\end{aligned}$$

(2.15)

If n=k the above equation will take the form,

$$\left[Q_{ij}\right] = 2\int_{\Phi_S}\phi^{p+m}(K_p w_p(\phi)) + K_s(\pi - w_s(\phi)))dwd\phi)$$

(2.16)

where $w_p(\phi)$, $w_s(\phi)$ and Φ_s are defined by (2.10).

Similarly,

$$\begin{aligned}\left[b\right]_i &= \int_{\phi_S}\int_{\Omega_S} W(e^{jw},\mathbf{\Phi}).\mathrm{Re}\{H_I(e^{jw},\mathbf{\Phi}).\overline{\phi_i(w,\mathbf{\Phi})}\}dwd\mathbf{\Phi} \\ &= 2\int_{\phi_S}\phi^{p}\frac{K_p}{(k-\tau)}\sin((k-\tau)w_p(\phi))d\phi\end{aligned}$$

(2.17)

If k= τ the above equation will take the form

$$\left[b\right]_i = 2\int_{\phi_S}\phi^{p}K_p w_p(\phi)d\phi$$

(2.18)

where $w_p(\phi)$ and Φ_s are defined by (2.10).

Putting (2.15), (2.16), (2.17) and (2.18) into (2.9) gives weighted least squares solution for variable FIR digital filter with Farrow structure shown in Figure 2.1.

2.3.1.2 Design Results and Comparisons

For evaluation of proposed method a variable lowpass filter with the specifications given in Table2-1 is designed.

Table 2-1 Parameters for tunable low pass filter.

Specification	Value
Spectral Parameter (ϕ)	$0 \le \phi \le 1$
Passband Edge1($w_{p1}(\phi)$)	0.2π
Passband Edge2($w_{p2}(\phi)$)	0.4π
Stopband Edge1($w_{s1}(\phi)$)	0.4π
Stopband Edge2($w_{s2}(\phi)$)	0.6π

In order to give a figure of merit to show the quality of the designed filter, some comparisons should be made among them. The quality figures are the maximum ripple in the passband, the maximum ripple in stopband, and error which is calculated according to (2.4).

The weight values specified for pass and stop band regions play an important role for error and ripple values. For the sake of comparison error and ripple values for designed filters for different values of K_p are given in Table 2-2.

Table 2-2 Characteristic values for tunable low pass filter for various values of K_p; for Ks = 1, Filter order = 10, Interpolation order = 3, $\phi = 0$.

K_p	Maximum Ripple in Passband (dB)	Maximum Ripple in Stopband (dB)	Error (dB)
1	-32.5012	-22.0005	-3.826
10	-29.4875	-19.814	-3.6263
100	-35.7768	-8.6555	17.3016
1000	-45.5163	-6.6874	26.9476

As it could be seen from Table 2-2, the higher values for K_p while keeping other filter parameters constant results in reduction in passband ripples, however, the total error and stop band ripple increase rapidly in order to ensure high K_p value.

The comparision of error and ripple values for designed filters for different values of K_s are given in Table 2-3.

Table 2-3 Characteristic values for tunable low pass filter for various values of K_s; for Kp = 1, Filter order = 10, Interpolation order = 3, $\phi = 0$.

K_s	Maximum Ripple in Passband (dB)	Maximum Ripple in Stopband (dB)	Error (dB)
1	-32.5012	-22.0005	-3.826
10	-25.1572	-20.5383	-1.7793
100	-13.8608	-26.3409	20.518
1000	-12.6678	-43.6452	24.3755

It is clearly seen from Table 2-3 that for increasing values of K_s, the stopband ripple of the designed filter decreases considerably, while the passband ripple values and the total error values increase. This result stems from the fact that in order to ensure

high K_s values the other filter parameters become less considerable in the error equations.

The comparision of error and ripple values for designed filters for different values of filter orders are given in Table 2-4.

Table 2-4 Characteristic values for tunable low pass filter for various values of filter order; for Kp = 1000, Kp = 1000, Interpolation order = 3, ϕ = 0.

Filter order	Maximum Ripple in Passband (dB)	Maximum Ripple in Stopband (dB)	Error (dB)
10	-32.5012	-22.0005	-3.826
20	-36.4305	-29.9151	-20.9615
30	-46.0334	-31.9205	-33.437
40	-50.4494	-33.2996	-41.4992

From Table 2-4 it can be seen that the passband, the stopband and the total error values for incresing filter order reduces rapidly. This is a direct and expected result as the increasing filter order always accompanied with reduction in ripples and error values.

The comparison of error and ripple values for designed filters for different values of interpolation orders are given in Table 2-5.

Table 2-5 Characteristic values for tunable low pass filter for various values of interpolation order; for Kp = 1000, Ks = 1000, Filter order = 31, $\phi = 0$.

Interpolation order	Maximum Ripple in Passband (dB)	Maximum Ripple in Stopband (dB)	Error (dB)
3	-46.0334	-31.9205	-33.437
5	-48.4577	-46.7176	-60.5496
7	-54.8899	-46.9443	-76.5439
9	-60.7898	-42.0663	-75.4524

From Table 2-5 it can be seen that the passband, the stopband and the total error values for incresing interpolation order reduces upto interpolation order of 7. Above this value while passband ripples reduces, stopband ripple and total error values increases. As the interpolation order increses the number of sub filters increases. However with increasing interpolation, for added filter branch multiplication value gets smaller values. This results in corresponding filter branch be less significant in total filter characteristic.

Magnitude responses for evenly sampled values of ϕ are given in Figure 2.2.

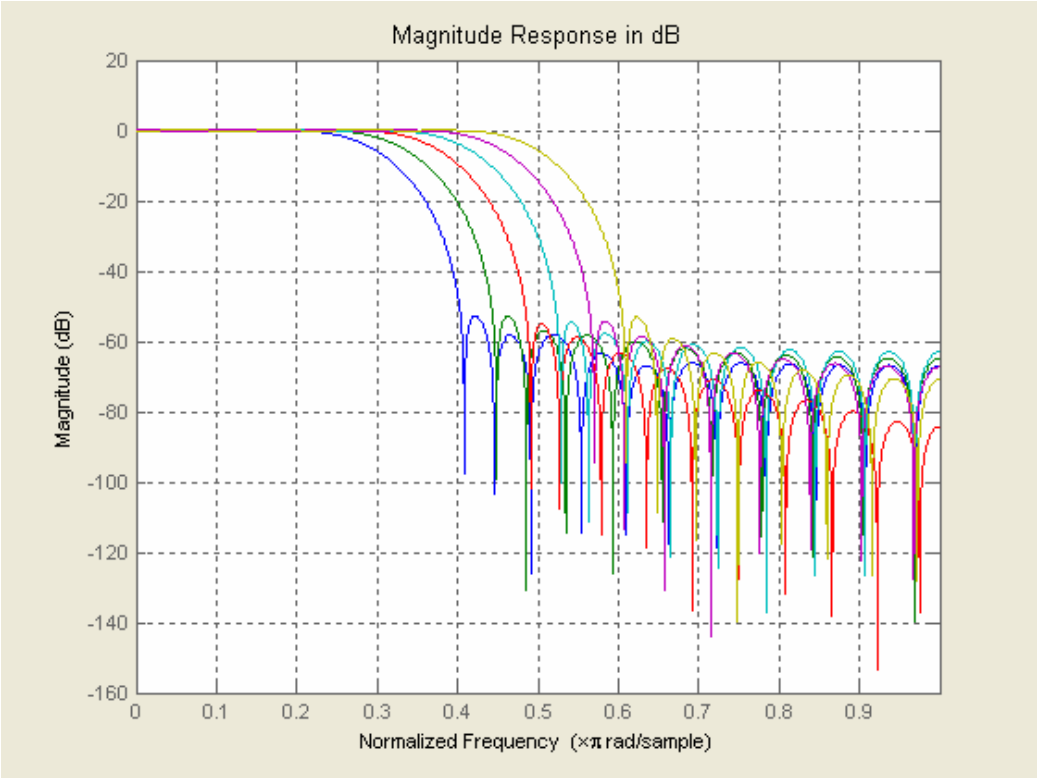

Figure 2.2 Frequency response of variable lowpass FIR linear phase filter evenly sampled in range $\phi = [0,1]$.

2.3.2 Tunable Linear-Phase Band-Pass Filter

The design of variable finite impulse response band-pass digital filter design is also worth considering. Letting the passband cutoff frequency be denoted by w_p and the stopband cutoff frequency be denoted by w_s, assuming the passband and the stopband cutoff frequencies vary linearly with spectral parameter $\mathbf{\Phi} = \phi$, the variable passband and stopband cutoff frequencies could be defined as:

$$
\begin{aligned}
w_p(\phi) &= w_c \pm (\phi(w_{p2} - w_{p1}) + w_{p1})/2 \\
&= w_c \pm w_{pp}(\phi) \\
w_s(\phi) &= w_c \pm (\phi(w_{s2} - w_{s1}) + w_{s1})/2 \\
&= w_c \pm w_{ss}(\phi) \\
\boldsymbol{\Phi}_s &: \phi \in [0,1]
\end{aligned}
$$

(2.19)

Frequency support of the filter is defined as combination of passband and stopband regions, which are defined by spectral parameter.

$$
\begin{aligned}
\Omega_S &= \Omega_p + \Omega_s \\
\Omega_p &= \{w : w \in (w_{p1}(\phi), w_{p2}(\phi))\} \\
\Omega_s &= \{w : w \in \{(0, w_{s1}(\phi)) \cup (w_{s2}(\phi), \pi)\}
\end{aligned}
$$

(2.20)

The desired impulse response is denoted by $H_I(e^{jw}, \phi)$ and defined as:

$$
H_I(e^{jw}, \phi) = \begin{cases} e^{-j\tau w}, w_{p1}(\phi) \le |w| \le w_{p2}(\phi) \\ 0, w_{S2}(\phi) \le |w| \le \pi \cup 0 \le |w| \le w_{S1}(\phi) \end{cases}
$$

(2.21)

where τ is the group delay which is assumed to be constant and equal to (N-1)/2 where N is the order of the sub filters.

2.3.2.1 Mathematical Derivations

(2-15) can be rearranged as in (2-22)

$$\left[Q_{ij}\right]=2\int_{\Phi_S}\phi^{p+m}\left(\frac{K_p}{(n-k)}(\sin((n-k)(w_c+w_{pp}(\phi)))-\sin((n-k)(w_c-w_{pp}(\phi))))dwd\phi+\right.$$
$$\left.2\int_{\Phi_S}\phi^{p+m}\left(\frac{K_S}{(n-k)}(\sin((n-k)(w_c-w_{ss}(\phi)))-\sin((n-k)(w_c+w_{ss}(\phi))))dwd\phi\right.\right.$$

(2.22)

If n=k the above equation will take the form as in (2-16)

$$\left[Q_{ij}\right]=2\int_{\Phi_S}\phi^{p+m}(2K_p w_{pp}(\phi))+K_s\pi)dwd\phi)$$

(2.23)

(2-17) can be rearranged as in (2-24)

$$\left[b\right]_i=2\int_{\phi_S}\phi^{p}\frac{K_p}{(k-\tau)}(\sin((k-\tau)(w_c+w_{pp}(\phi)))-\sin((k-\tau)(w_c-w_{pp}(\phi))))d\phi$$

(2.24)

If k=τ the above equation will take the form as in (2-25)

$$\left[b\right]_i=4\int_{\phi_S}\phi^{p}K_p w_{pp}(\phi)d\phi$$

(2.25)

where $w_{pp}(\phi)$ and $w_{ss}(\phi)$ are defined by (2-19).

2.3.2.2 Design Results and Comparisons

For evaluation of the proposed method, a variable filter with the specifications given in Table 2-6 is designed.

Table 2-6 Parameters for tunable band pass filter.

Specification	Value
Spectral Parameter (ϕ)	$0 \le \phi \le 1$
Passband Edge1($w_{ph1}(\phi)$)	0.6π
Passband Edge2($w_{ph2}(\phi)$)	0.7π
Passband Edge3($w_{pl1}(\phi)$)	0.4π
Passband Edge4($w_{pl2}(\phi)$)	0.3π
Stopband Edge1($w_{sh1}(\phi)$)	0.7π
Stopband Edge1($w_{sh2}(\phi)$)	0.8π
Stopband Edge3($w_{sl1}(\phi)$)	0.3π
Stopband Edge4($w_{sl2}(\phi)$)	0.2π

The comparison of error and ripple values for designed filters for different values of filter orders are given in Table 2-7.

Table 2-7 Characteristic values for tunable band pass filter for various values of filter order; for Kp = 1000, Ks = 1000, Interpolation order = 3, $\phi = 0$, $w_c = 0.5\pi$.

Filter order	Maximum Ripple in Passband (dB)	Maximum Ripple in Stopband (dB)	Error (dB)
10	Inf	-6.5315	67.1928
20	-21.5864	-14.3465	67.4243
30	-17.8884	-17.6304	67.5264
40	-34.2529	-19.2417	67.3992

From Table 2-7 it can be concluded that for increasing values of filter order the ripple values decrease to some extent. The error values are approximately the same.

The comparison of error and ripple values for designed filters for different values of interpolation orders are given in Table 2-8.

Table 2-8 Characteristic values for tunable low pass filter for various values of interpolation order; for Kp = 1000, Ks = 1000, Interpolation order = 31, $\phi = 0$.

Interpolation order	Maximum Ripple in Passband (dB)	Maximum Ripple in Stopband (dB)	Error (dB)
3	-17.8884	-17.6304	67.5264
5	-19.0114	-16.9435	67.4988
7	-19.0461	-16.7118	67.4935
9	-19.0162	-16.7626	67.4951

For increasing values of interpolation order, the pass band and stop band ripples get lower values; however, after the order 5 the decrease in not considerable. The error values are approximately the same for all of the values of interpolation order.

Magnitude responses for evenly sampled values of ϕ are given in Figure 2.3.

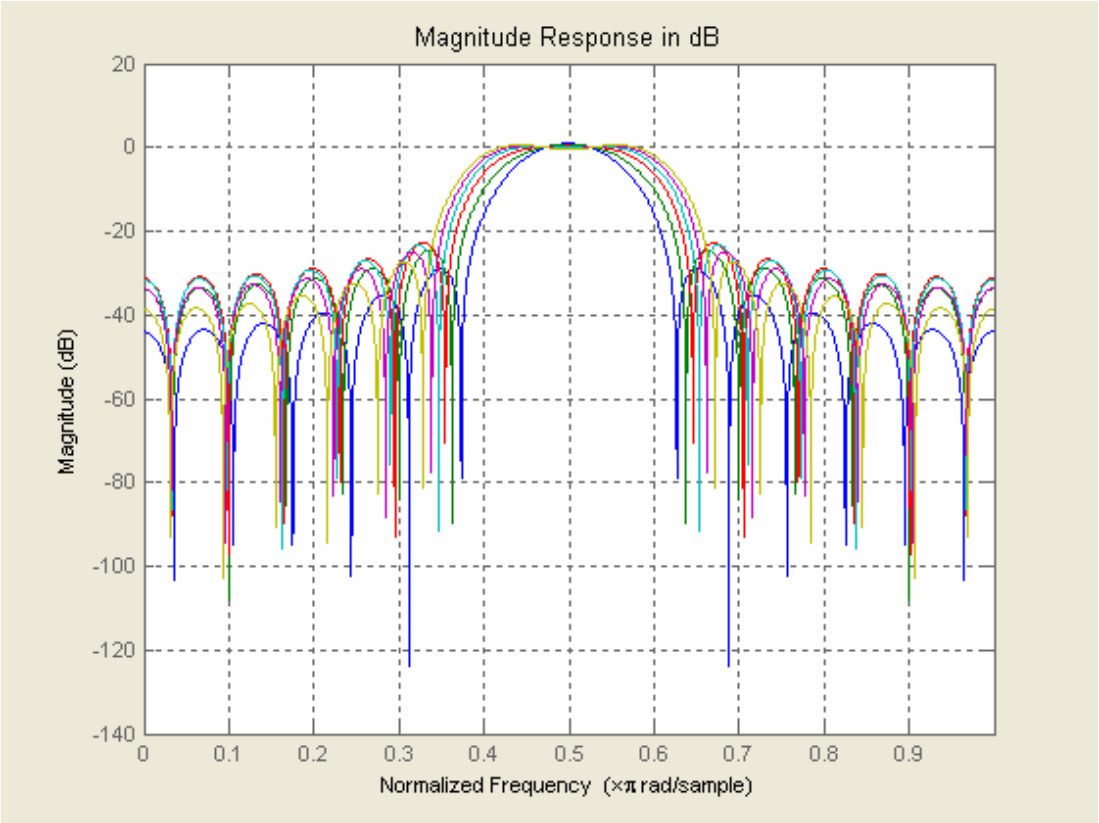

Figure 2.3 Frequency response of variable bandpass FIR linear phase filter evenly sampled in the range $\phi = [0,1]$.

2.3.3 Tunable Fractional Delay Low-Pass FIR Digital Filter

Variable fractional delay digital filters have important application areas especially when timing adjustment is required. For this reason an example of variable fractional delay digital filter is worth considering.

The desired impulse response of variable fractional delay digital filter is denoted by $H_I(e^{jw}, \phi)$ and defined as:

$$H_I(e^{jw},\phi)=\begin{cases} e^{-j\tau(\phi)w}, 0\le|w|\le w_p \\ 0, w_S\le|w|\le\pi \end{cases}$$
$$\tau(\phi)=(N-1)/2+\phi$$
$$\phi\in[-0.5,0.5]$$

(2.26)

2.3.3.1 Mathematical Derivations

(2-15) can be rearranged as in (2-27)

$$[Q_{ij}]=2\int_{\Phi_S}\phi^{p+m}\left(\frac{K_p}{(n-k)}\sin((n-k)w_p)-\frac{K_s}{(n-k)}\sin((n-k)w_s)\right)d\phi$$

(2.27)

If n=k, the above equation will take the form,

$$[Q_{ij}]=2\int_{\Phi_S}\phi^{p+m}(K_p w_p+K_s(\pi-w_s))d\phi$$

(2.28)

(2-17) can be rearranged as in (2-29)

$$[b]_i=2\int_{\phi_S}\phi^{p}\frac{K_p}{(k-(N-1)/2-\phi)}\sin(k-(N-1)/2-w_p)d\phi$$

(2.29)

If k=(N-1)/2+ϕ, the above equation will take the form as in (2-30)

$$[b]_i=2K_p w_p\phi$$

(2.30)

where N is the order of the subfilters.

2.3.3.2 Design Results and Comparisons

For evaluation of proposed method, a variable filter with the specifications given in Table 2-9 is designed.

Table 2-9 Parameters for tunable fractional delay low pass filter.

Specification	Value
Spectral Parameter (ϕ)	$-0.5 \le \phi \le 0.5$
Passband Edge(w_p)	0.2π
Stopband Edge(w_s)	0.4π

The comparison of error and ripple values for designed filters for different values of filter orders are given in Table 2-10.

Table 2-10 Characteristic values for tunable fractional delay filter for various values of filter order; for Kp = 1000, Ks = 1000, Interpolation order = 3, ϕ = 0.

Filter order	Maximum Ripple in Passband (dB)	Maximum Ripple in Stopband (dB)	Error (dB)
10	-34.692	-18.6928	-5.9603
20	-38.646	-34.2072	-39.6818
30	-58.7456	-46.1449	-77.8958
40	-67.3453	-63.3082	-111.7469

From Table2-10 it can be concluded that all of the characteristic values get lower values as filter order incereases. The decrease in parameters are more considerable

because in this case the spectral parameter is the fractional delay, meaning that, there is no difference in magnitude characteristics. This fact results in a considerable decrease in all of the characteristic parameters.

The comparison of error and ripple values for designed filters for different values of interpolation orders are given in Table 2-11.

Table 2-11 Characteristic values for tunable fractional delay filter for various values of interpolation order; for Kp = 1000, Ks = 1000, Filter order = 31, $\phi = 0$.

Interpolation order	Maximum Ripple in Passband (dB)	Maximum Ripple in Stopband (dB)	Error (dB)
3	-58.7456	-46.1449	-77.8958
5	-58.583	-46.1438	-77.8974
7	-58.5829	-46.1438	-77.8974
9	-58.5829	-46.1438	-77.8974

After interpolation order of 5, the ripple values and the total error does not change for this example.

Group delays in pass band region for evenly sampled values of ϕ are given in Figure 2.4 for the filter specificified in Table 2-9. Here the filter order 30, therefore the group delay is chosen to be 15, and fractional delay is assumed to change in between -0.5-0.5.

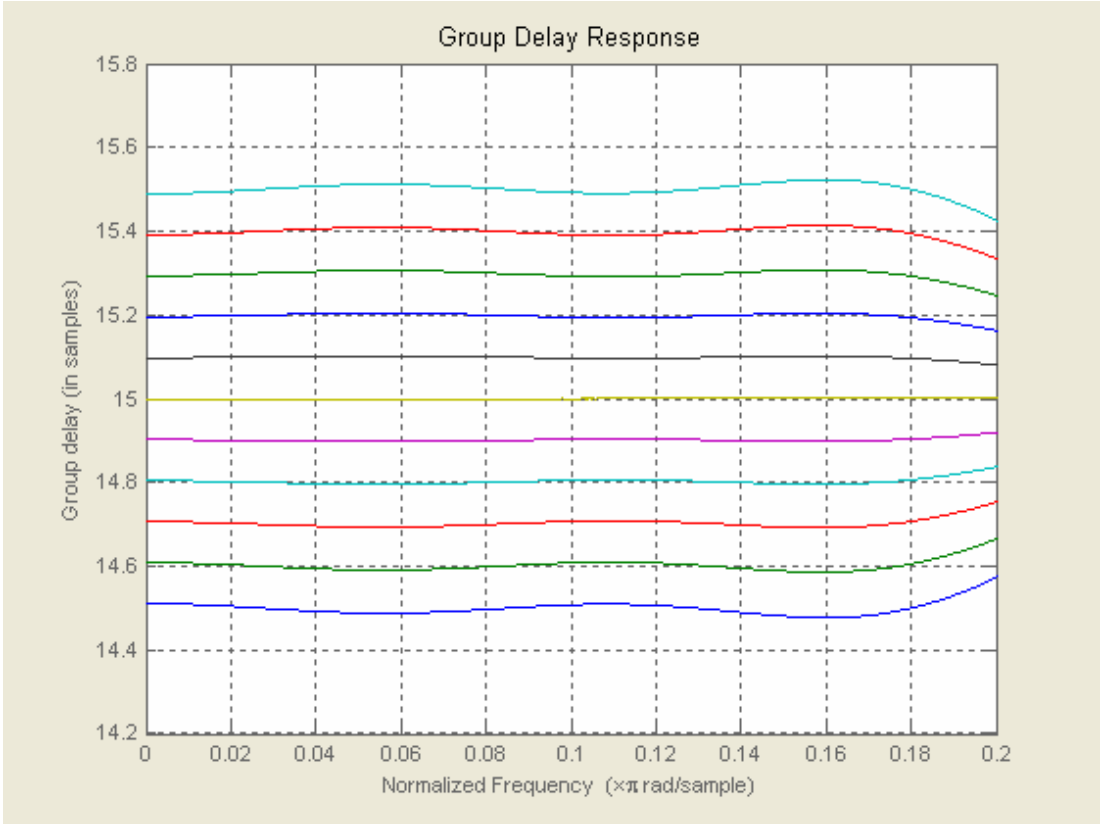

Figure 2.4 Group delay response of variable lowpass FIR filter evenly sampled in range $\phi = [-0.5, 0.5]$.

CHAPTER 3

VARIABLE FIR FILTER DESIGN WITH SINGULAR VALUE BASED VECTOR ARRAY DECOMPOSITION METHOD

3.1 General

In this chapter the variable FIR filter design by singular value decomposition based vector array decomposition method is described and the mathematical derivations method are given. Illustrative examples for variable single spectral parameter low pass and design results for these types of filters are given.

3.2 Design Method

Let $H_I(w,\psi_1,\psi_2,...\psi_K)$ be the ideal variable frequency response specification, where

$$\begin{aligned}\psi_1 &\in [\psi_{1\min}, \psi_{1\max}] \\ \psi_2 &\in [\psi_{2\min}, \psi_{2\max}] \\ &\vdots \\ \psi_K &\in [\psi_{K\min}, \psi_{K\max}]\end{aligned}$$

(3.1)

are the spectral parameters that are use to tune magnitude and/or phase responses. By sampling the parameters $w, \psi_1, \psi_2, ..., \psi_K$ uniformly as

$$\begin{aligned}w(n) &= -\pi + \frac{2\pi(n-1)}{N-1}, n = 1,2,..., N \\ \psi_1(m_1) &= \psi_{1\min} + \frac{(\psi_{1\max} - \psi_{1\min})(m_1 - 1)}{M_1 - 1}, m_1 = 1,2,...M_1 \\ \psi_2(m_2) &= \psi_{2\min} + \frac{(\psi_{21\max} - \psi_{2\min})(m_2 - 1)}{M_2 - 1}, m_2 = 1,2,...M_2 \\ &\vdots \\ \psi_K(m_K) &= \psi_{K\min} + \frac{(\psi_{K\max} - \psi_{K\min})(m_K - 1)}{M_K - 1}, m_K = 1,2,...M_K\end{aligned}$$

(3.2)

Equally spaced samples of $H_I(w, \psi_1, \psi_2, ...\psi_K)$ are

$$\tilde{a}(n, m_1, m_2, ..., m_K) = H_I[w(n), \psi_1(m_1), \psi_2(m_2), ..., \psi_K(m_K)]$$

(3.3)

These samples can be used to construct a (K+1) dimensional complex array $\tilde{A}$ as

$$\tilde{A} = [\tilde{a}(n, m_1, m_2, ..., m_K)]$$

(3.4)

If one spectral parameter is used, K = 1, $\tilde{A}$ is a 2-D complex array, if K = 2, $\tilde{A}$ is a 3-D complex array, and $\tilde{A}$ is a hyper cube if K $\geq$ 3.

Here the design consideration is to decompose $\widetilde{\mathrm{A}}$ as

$$\widetilde{\mathrm{A}} \approx \sum_{\mathrm{i}=1}^{\mathrm{r}} \mathrm{C}_{\mathrm{i}} \otimes \mathrm{R}_{\mathrm{i}}$$

(3.5)

under the following constraints:

- ✓ C_i are complex conjugate-symmetric vectors, and R_i are K-D real arrays
- ✓ Mean-squared decomposition error

$$\begin{aligned} e_r &= \left\| \widetilde{A} - \sum_{i=1}^{r} C_i \otimes R_i \right\|^2 \\ &= \sum_{n=1}^{N} \sum_{m_1=1}^{M_1} \cdots \sum_{m_K=1}^{M_K} \left| \widetilde{a}(n, m_1, m_2, ..., m_K) - \sum_{i=1}^{r} C_i(n) R_i(m_1, m_2, ..., m_K) \right|^2 \end{aligned}$$

(3.6)

is minimized, where $\|.\|$ is the Euclidian norm (l_2 norm), the notation $\otimes$ denotes a special product between the complex vector C_i and the K-D real array R_i, and the decompostion in (3-5) means

$$\widetilde{a}(n, m_1, m_2, ..., m_K) \approx \sum_{i=1}^{r} C_i(n) R_i(m_1, m_2, ..., m_K)$$

(3.7)

The first constraint is imposed for the reason that the complex symmetric vectors C_i can be regarded as the desired frequency responses of constant 1-D digital filters with real coefficients and that the K- dimensional real arrays R_i can be regarded as the desired values of K-D polynomials of the spectral parameters $\psi_1, \psi_2, ..., \psi_K$.

The filter structure of specified variable FIR filter would be as in Figure 3.1.

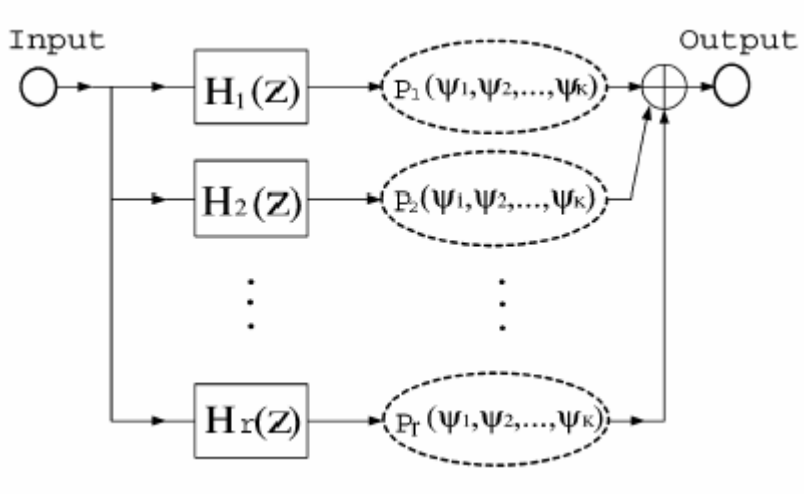

Figure 3.1 Proposed vector array decomposition based filter structure.

Here once the vector array decomposition is obtained, the constant 1-D filters $H_i(z)$ can be designed by approximating the complex vectors C_i, and the K-D polynomials $P_i(\psi_1, \psi_2, ..., \psi_K)$ can be found for approximating the K-D real arrays R_i, the variable 1-D filter

$$H(z, \psi_1, \psi_2, ..., \psi_K) = \sum H_i(z) P_i(\psi_1, \psi_2, ..., \psi_K) \tag{3.8}$$

can be indirectly obtained by connecting $H_i(z)$ with $P_i(\psi_1, \psi_2, ..., \psi_K)$.

3.2.1 Vector Array Decomposition Algorithm

In order vector array decompositon algorithm to be applied first (K+1)-D array should be reduced to matrix form. Based on one-to-one index mappings

$$n \to n$$
$$(m_1, m_2, ..., m_K) \to m$$

(3.9)

where

$$\mathrm{m} = (\mathrm{m}_1 - 1)\prod_{\mathrm{k}=2}^{\mathrm{K}} \mathrm{M}_\mathrm{K} + (\mathrm{m}_2 - 1)\prod_{\mathrm{k}=3}^{\mathrm{K}} \mathrm{M}_\mathrm{K} + ... + (\mathrm{m}_{\mathrm{K}-1} - 1)\mathrm{M}_\mathrm{K} + \mathrm{m}_\mathrm{K}$$

(3.10)

such that

$$a(n, m) = \tilde{a}(n, m_1, m_2, ..., m_K)$$

(3.11)

Sigular value decomposition of A is

$$\mathrm{A} = \mathrm{U}\sum \mathrm{V}^* = \sum_{\mathrm{i}=1}^{\mathrm{R}} \sigma_\mathrm{i} \mathrm{u}_\mathrm{i} \mathrm{v}^*{}_\mathrm{i} = \sum_{\mathrm{i}=1}^{\mathrm{R}} \tilde{\mathrm{u}}_\mathrm{i} \tilde{\mathrm{v}}^*{}_\mathrm{i}$$

(3.12)

where R is the rank of A.

$$\mathrm{U} = \begin{bmatrix} \mathrm{u}_1 & \mathrm{u}_2 & \cdots & \mathrm{u}_\mathrm{R} \end{bmatrix}$$
$$\mathrm{V} = \begin{bmatrix} \mathrm{v}_1 & \mathrm{v}_2 & \cdots & \mathrm{v}_\mathrm{K} \end{bmatrix}$$

(3.13)

are unitary matrices and

$$\sum = \mathrm{diag}\begin{pmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_\mathrm{R} \end{pmatrix}$$

(3.14)

is a diagonal matrix with the singular values σ_i as its diagonal entries; $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R \geq 0$ and

$$\tilde{u}_i = \sqrt{\sigma_i}\, u_i$$
$$\tilde{v}_i = \sqrt{\sigma_i}\, v_i$$

(3.15)

Singular value decomposition always guarantees that the mean squared error is minimum for any $r \leq R$.

When it comes to the design of constant filters $H_i(z)$ from vectors C_i, assuming equally spaced samples of the frequency response $H_i(e^{jw})$

$$H_i(k) = \sum_{n=N_1}^{N_2} a_n e^{-jw(k)n}$$

(3.16)

where k = 1,2,...,K. The design objective here is to find the optimal coefficients a_n such that the mean squared error is minimum. That is

$$e_H = \sum_{k=1}^{K} \left| H_i(k) - C_i(k) \right|^2$$
$$= \sum_{k=1}^{K} \left[H_i(k) - C_i(k) \right] \left[H_i(k) - C_i(k) \right]^*$$

(3.17)

where $H_i(k)$ is given in (3-16) and $C_i(k)$ values are obtained from singular value decomposition.

Putting e_H values into the matrix form and after some manipulations, one can obtain

$$\mathrm{Re}\left[\mathrm{E}^{*}\mathrm{Ea}\right]=\mathrm{Re}\left[\mathrm{E}^{*}\mathrm{C}_{\mathrm{i}}\right]$$

(3.18)

where

$$\mathrm{E}=\begin{bmatrix} \mathrm{e}(1,\mathrm{N}_1) & \mathrm{e}(1,\mathrm{N}_1+1) & \cdots & \mathrm{e}(1,\mathrm{N}_2) \\ \mathrm{e}(2,\mathrm{N}_1) & \mathrm{e}(2,\mathrm{N}_1) & \dots & \mathrm{e}(2,\mathrm{N}_2) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{e}(\mathrm{L},\mathrm{N}_1) & \mathrm{e}(\mathrm{L},\mathrm{N}_1) & \cdots & \mathrm{e}(\mathrm{L},\mathrm{N}_2) \end{bmatrix}$$

(3.19)

elements e(k,n) defined as

$$e(k,n)=e^{-jw(k)n},\begin{cases} k=1,2,...,K \\ n=N_1,N_1+1,...,N_2 \end{cases}$$

(3.20-a)

$$a=\begin{bmatrix} a_{N_1} \\ a_{N_L+1} \\ \vdots \\ a_{N_2} \end{bmatrix},C_i=\begin{bmatrix} C_i(1) \\ C_i(2) \\ \vdots \\ C_i(K) \end{bmatrix}$$

(3.20-b)

The optimal coefficient vector **a** can be determined as

$$\mathbf{a}=\left\{\mathrm{Re}\left[\mathrm{E}^{*}\mathrm{E}\right]\right\}^{-1}.\mathrm{Re}\left[\mathrm{E}^{*}\mathrm{C}_{\mathrm{i}}\right]$$

(3.21)

The K-D polynomials $\mathrm{P_i}(\psi_1,\psi_2,\cdots,\psi_K)$ can be obtained by using the polynomial fitting technique. The values of $\mathrm{P_i}(\psi_1,\psi_2,\cdots,\psi_K)$ are

$$P[\psi_1(m_1),\psi_2(m_2),...,\psi_K(m_K)]=\sum_{l_1=0}^{L_1}\sum_{l_2=0}^{L_2}\cdots\sum_{l_K=0}^{L_K}c(l_1,l_2,...,l_K).\psi_1^{l_1}(m_1)\psi_2^{l_2}(m_2)...\psi_K^{l_K}(m_K)$$

(3.22)

Letting the desired values of $P[\psi_1(m_1),\psi_2(m_2),...,\psi_K(m_K)]$ be denoted by $d(m_1,m_2,...,m_K)$ and the optimal coefficients for $P[\psi_1(m_1),\psi_2(m_2),...,\psi_K(m_K)]$ by $c(l_1,l_2,...,l_K)$ for minimizing the approximation error linearly following steps should be followed:

- ✓ Map the K-D index (m_1,m_2,...,m_K) to 1-D index as

$$m=(m_1-1)M_2M_3\cdots M_K+(m_2-1)M_3M_4\cdots M_K+\cdots+(m_{K-1}-1)M_K+m_K$$

(3.23)

where

$$m_i\in\{1,2,\cdots,M_i\}$$
$$i\in\{1,2,\cdots,K\}$$
$$m\in\{1,2,\cdots,M\}$$
$$M=M_1M_2\cdots M_K$$

(3.24)

- ✓ Map the M-D index (l_1, l_2,..., l_K) to the 1-D index l as

$$l=l_1(L_2+1)(L_3+1)\cdots(L_K+1)+l_2(L_3+1)(L_4+1)\cdots(L_K+1)+\cdots+l_{K-1}(L_K+1)+(l_K+1)$$

(3.25)

where

$$l_i\in\{0,1,\cdots,L_i\}$$
$$i\in\{1,2,\cdots,K\}$$
$$l\in\{1,2,\cdots,L\}$$
$$L=(L_1+1)(L_2+1)\cdots(L_K+1)$$

(3.26)

Based on above index mappings, one gets

$$
\begin{aligned}
C(l) &= c(l_1, l_2, \cdots, l_K) \\
a(m) &= d(m_1, m_2, \cdots, m_K) \\
\Phi_{ml} &= \psi_1^{l_1}(m_1)\psi_2^{l_2}(m_2)\cdots\psi_K^{l_K}(m_K)
\end{aligned}
$$

(3.27)

The error is now defined as

$$
E_p = \sum_{m=1}^{M} \left[\sum_{l=1}^{L} C(l)\Phi_{ml} - a(m) \right]^2
$$

(3.28)

In order to minimize E_p, differentiating E_p with respect to each coefficient $C(l)$ and setting the derivatives to zero, one gets simultaneous linear equations

$$
\mathbf{AC} = \mathbf{b}
$$

(3.29)

where

$$
\begin{aligned}
A &= \Phi^t \Phi \\
\Phi &= \begin{bmatrix} \Phi_{11} & \Phi_{12} & \cdots & \Phi_{1L} \\ \Phi_{21} & \Phi_{22} & \cdots & \Phi_{2L} \\ \vdots & \vdots & \vdots & \vdots \\ \Phi_{M1} & \Phi_{M2} & \cdots & \Phi_{ML} \end{bmatrix} \\
C &= \begin{bmatrix} C(1) & C(2) & \cdots & C(L) \end{bmatrix}^t \\
b &= \Phi^t a \\
a &= \begin{bmatrix} a(1) & a(2) & \cdots & a(M) \end{bmatrix}
\end{aligned}
$$

(3.30)

Solving the linear equations, the optimal coefficient vector **C** can be obtained. Finally, the coeficients $c(l_1, l_2, ..., l_K)$ can be determined from the resulting vector **C** in the reverse order.

3.3 Illustrative Examples

3.3.1 Tunable Linear-Phase Low-Pass FIR Filter

For evaluation of proposed method a variable lowpass filter with the specifications given in Table 2-1 is designed.

The comparison of error and ripple values for designed filters for different values of frequency band sample points are given in Table 3-1.

Table 3-1 Characteristic values for tunable low pass filter for various values of band sample points; for spectral parameter sample points =9, Filter order = 10, Interpolation order = 3, ϕ = 0.

Sample points	Maximum Ripple in Passband (dB)	Maximum Ripple in Stopband (dB)	Error (dB)
50	-30.7469	-24.0659	-1.6167
100	-30.7646	-24.1245	-1.6524
500	-30.6607	-24.1092	-1.54
1000	-30.6535	-24.1111	-1.5299

It can be concluded from Table 3-1 that increasing the number of frequency band sampling points have no considerable effect on the characteristic values of the designed filter, especiallly after 100 sample points.

The comparison of error and ripple values for designed filters for different values of filter orders are given in Table 3-2.

Table 3-2 Characteristic values for tunable low pass filter for various values of filter order; for frequency band sample points = 500, Interpolation order = 3, $\phi = 0$.

Filter order	Maximum Ripple in Passband (dB)	Maximum Ripple in Stopband (dB)	Error (dB)
10	-30.6607	-24.1092	-1.54
20	-39.1081	-31.8706	-24.9873
30	-40.7246	-27.2175	-19.7707
40	-46.9886	-27.4835	-20.4907

With the increasing number of filter order the ripple values in pass band, stop band and the error values decrease considerably. This is an expected result with an increse of filter order of subbranches.

The comparison of error and ripple values for designed filters for different values of interpolation orders are given in Table 3-3.

Table 3-3 Characteristic values for tunable low pass filter for various values of filter order; for frequency band sample points = 500, Filter order = 30, $\phi = 0$.

Interpolation order	Maximum Ripple in Passband (dB)	Maximum Ripple in Stopband (dB)	Error (dB)
3	-40.7246	-27.2175	-19.7707
5	-38.3223	-31.4681	-34.2726
7	-39.2236	-30.6817	-35.827
9	-39.2149	-30.6978	-35.8503

From Table 3-3 it can be concluded that, when the interpolation order is increased up to 7 the error and ripple values decrease considerably. After that point a little change can be observed. This result stems from the fact that with increasing number of interpolation order, the multiplication values for each branch also decreases so the difference becomes negligible.

Magnitude responses for evenly sampled values of ϕ are given in Figure 3.2.

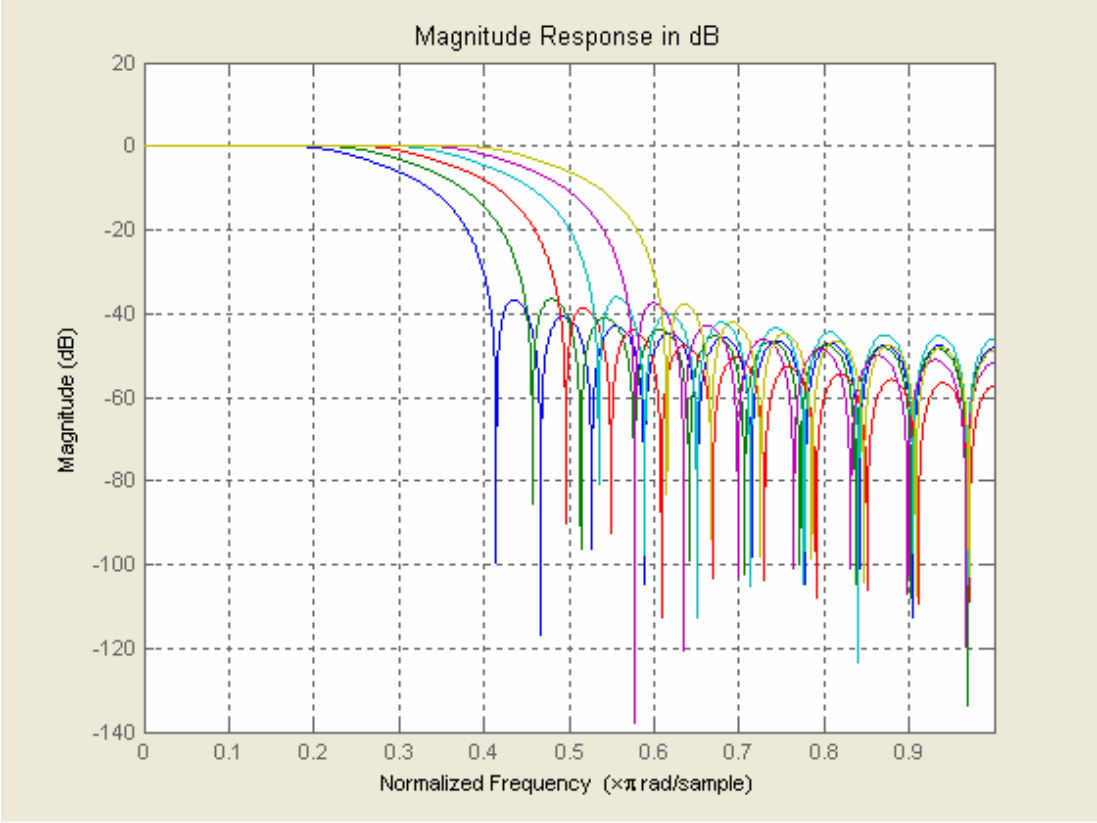

Figure 3.2 Frequency response of variable lowpass FIR linear phase filter evenly sampled in the range $\phi = [0,1]$.

3.4 Comparison of Direct Integration and Singular Value Decomposition Based Vector Array Decomposition Methods

The basic difference between the two methods is the computation times. Since the direct integration method requires integrations to be performed numerically, the required computation time is considerably more than that of the singular value decomposition based vector array decomposition.

When ripple values and errors are considered, it is seen that the performances of the the two methods are not so different.

From these considerations it is concluded that singular value decomposition based method would be preferrable in applications where the computation time is important.

CHAPTER 4

APPLICATION ON SOFTWARE DEFINED RADIO

4.1 General

One of the typical application areas of the variable digital filters is software defined radio. In this chapter, brief information about software defined radio will be given and usage of variable digital filters on software defined radios will be outlined.

Software application to show the implementation of proposed and traditional method will be considered. The obtained results from these implementations will be compared.

A hardware application on software defined radio will be discussed. The PCB (Printed Circuit Card) which will be used to realize traditional structure will be introduced. Operating system requirements in order to communicate with the application card will be discussed.

4.2 Software Defined Radio Description

Software defined radio is a general hardware/software platform for supporting inter-communication between different wireless communication systems. The basic idea of an ideal software defined receiver is to digitize the received signal using high-speed analog-to-digital converters and to process it by a programmable system, consisting of combination of hardware and software that is reconfigurable or programmable. Due to limitations of current technology and signal converters, most software radio architectures considered digitize the decimated signal at the intermediate frequency.

Since different communication standards are based on different master clock rates it is mainly necessary to provide these different clock rates. However, due to strong requirements for clock quality, it is reasonable to assume that only one fixed master clock will be provided in practical sofware radio applications. A solution to this is to provide different clock rates virtually by means of digital sample rate conversion. Hence, with software defined radio a new functionality is introduced: sample rate converters. Sample rate conversion is the task of converting the sample rate while a certain amount of information, usually in a limited frequency band, is preserved. Actually, the main concern in sample rate conversion is is not interpolation but anti-aliasing.

A conventional software defined radio receiver structure could be seen in Figure 4.1. In this general structure a programmable digital decimator and a sample rate converter are employed to isolate the desired user's channel from the signal spectrum and convert it to an appropriate sampling rate for further processing.

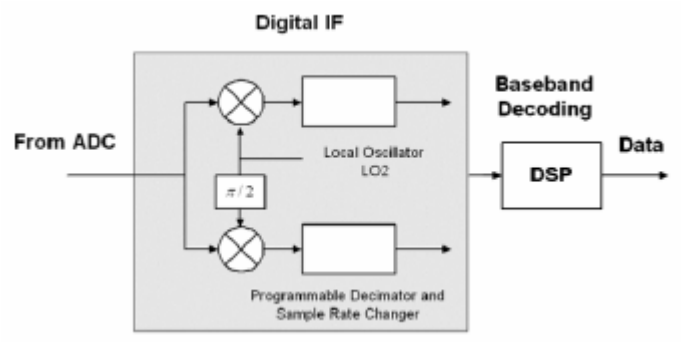

Figure 4.1 Conventional software defined radio receiver structure.

Moreover the programmable digital decimator usually consists of multiple stages of decimators to reduce implemetation complexity and power dissipation as shown in Figure 4.2.

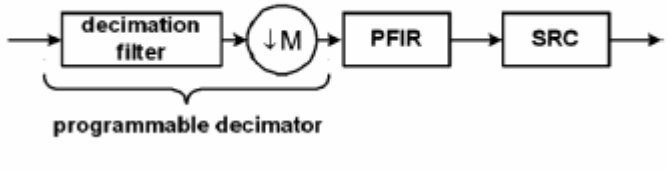

Figure 4.2 Conventional programmable decimator and sample rate converter.

As the sampling rate of the baseband signal is much lower than that of the intermediate frequency, each stage in the decimator will consist of a bandlimiting (anti-aliasing) digital filter and a downsampler (decimator) to filter out the unwanted signals and lower sampling rate. By selecting an appropriate number of stages, different integer downsampling ratios can be implemented. The programmable finite impulse response filter is used to remove the residual interference from adjacent channels. It is because the sampling rate is usually not an integer multiples of the channel spacing. Together with sample rate converter, which provides the necessary arbitrary rate-change factor, it is possible to accomodate signals with a wide variety of bandwidths.

One drawback of this conventional structure is that the output of the multistage decimators, which is obtained by downsampling the high-rate conversion, has to be upsampled again in order to carry out the arbitrary sample rate conversion. Recently authors [3] have proposed a new digital intermediate frequency architecture for software radio receivers shown in Figure 4.3-a.

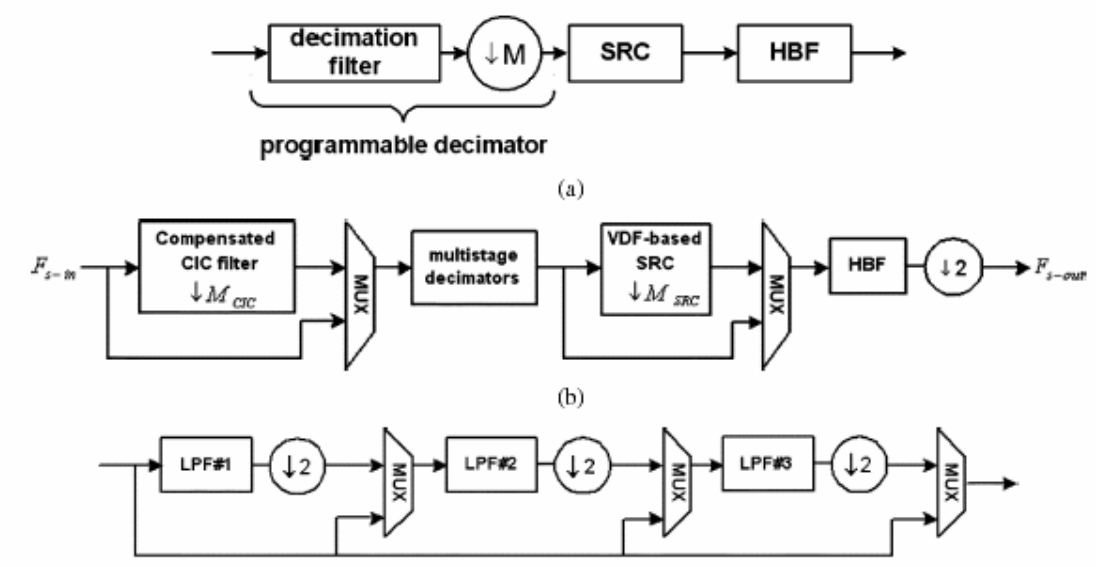

Figure 4.3 (a) Proposed architecture of the programmable decimator and sample rate converter, (b)Architecture of the proposed software radio receiver, (c)Architecture of the multistage decimators.

The sample rate converter which is realized using Farrow based variable digital filter is inserted immediately after the multistage decimators. The basic idea of the variable digital filter based sample rate converter is to provide variable fractional delay in the pass band and additional attenuation in the stop band. This allows replacing programmable finite impulse response filter by a half band filter with fixed coefficients, if the arbitrary rate-change factor is properly chosen.

This architecture eliminates the need for the programmable finite impulse response filter, which is usually a bottleneck of software defined radio application for wideband signals.

4.3 Software Simulations

4.3.1 Simulink Model of Proposed Method

The proposed method is implemented on Matlab Simulink according to Figure 4.3 and overall model could be seen in Figure 4.6. In the figure different sample rates are represented by different colors.

Here as input signal 500 Hz square wave generator is used. The generated signal is then discretized by zero-order hold unit with sampling time 0.00001 seconds.

The CIC, Cascade Comb Integrator, stage is implemented with 4 stage integrator and 4 stage decimator with downsampling ratio of 4. The structure of CIC stage is given in Figure 4.21.

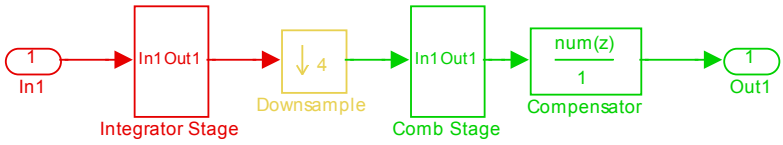

Figure 4.4 Simulink CIC Stage.

The CIC stage consists of an integrator and a comb stage, a downsampler unit is inserted in between. The basic CIC filters [22] are commonly used when large downsampling ratios are required, because of its reasonable performance and low hardware complexity. One drawback of the CIC filter is the pass band droop that limits the quality of the anti-aliasing filters. In [3], a second order CIC compensator

with the following transfer function is applied after the comb stage, realized by 'Compensator' block in Figure 4.4.

$$P(z) = a + bz^{-1} + az^{-2} \tag{4.1}$$

'a' and 'b' are real-valued constants to be determined, in particular, for this simulation these values are determined by using Parks-McClellan algorithm. This compensator can be viewed as the equalizer in the interpolated FIR filters [23].

The integrator and comb filter structures could be seen in Figure 4.5 and Figure 4.6 respectively.

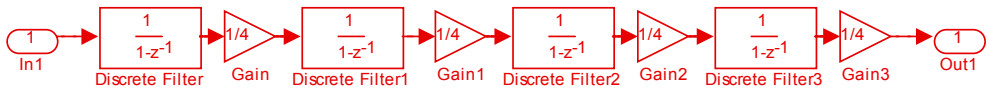

Figure 4.5 Integrator Stage of CIC.

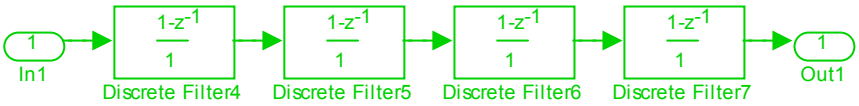

Figure 4.6 CombStage of CIC.

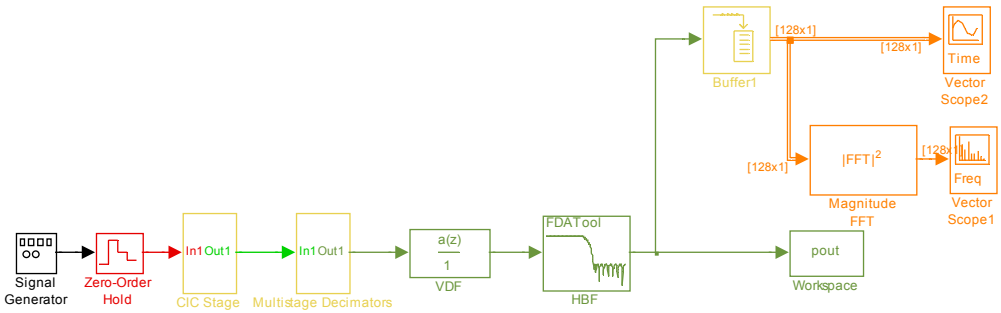

Figure 4.7 Simulink Implementation of Proposed Method.

CIC stage is followed by multistage decimator block. Multistage decimator block consists of three low pass filters and three downsampler units with downsampling ratio of 2. Mutistage decimator unit is given in Figure 4.8.

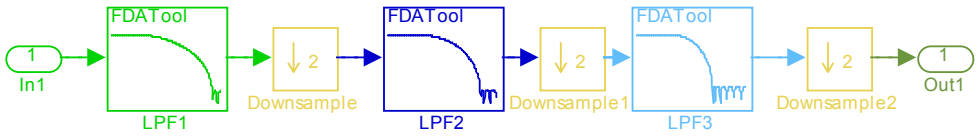

Figure 4.8 Simulink Multistage Decimator Stage.

Multistage decimator is followed by VDF based filter stage. This filter stage coefficients are computed by the direct integration method algorithm, detailed in Section 2, and loaded to filter as an external parameter. The filter pass band cut off frequency is at 0.4 and stop band frequency is at 0.7. Subfilter order is 41 and group delay is capable of changing in between 19.5 to 20.5.

In the proposed method, the VDF stage is followed by half band stage. Here again FIR type half band filter is designed using filter design tool of Simulink.

'To Workspace' block is used to store data for further comparison that will be done using Matlab software.

4.3.2 Simulink Model of Traditional Method

In traditional SDR model CIC and multistage decimator units are used, as they are in proposed method. After multistage decimator stage a programmable FIR, PFIR, filter is used. This filter coefficients are capable of changing their values during operation on request.

After PFIR section SDR unit comes, which is implemented by interpolator unit. For proper operation of interpolator unit input samples should be upsampled. In Matlab Simulink 'Variable Fractional Delay' block is designed to operate in the same manner, input sample values are first upsampled and then interpolation is done together with filtering. Finally decimation is done on samples to reduce the sample rate again.

The output data produced by the simulation is stored for comparison and corresponding model could be seen in Figure 4-9.

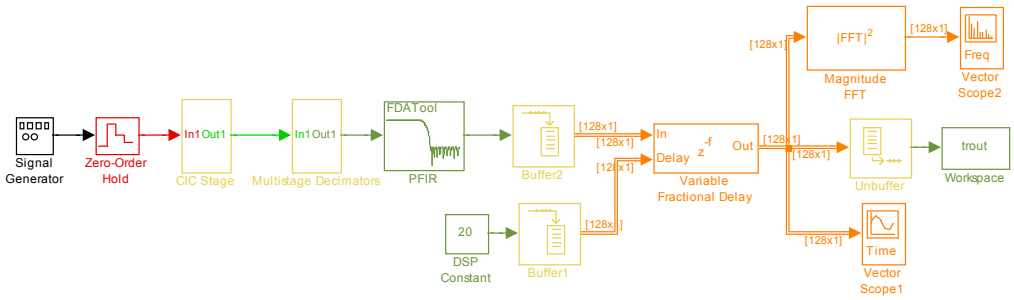

Figure 4.9 Simulink Implementation of Traditional Method.

4.3.3 Simulation Results

Simulation on Matlab Simulink is performed on proposed and traditional methods. Simulation time for both of them is arranged to be 0.05 seconds. The fractional delays are chosen to be 0.15 for both of the cases. The output data waveform and magnitude of fast fourier transform could be seen in Figure 4-10 and Figure 4-11 respectively.

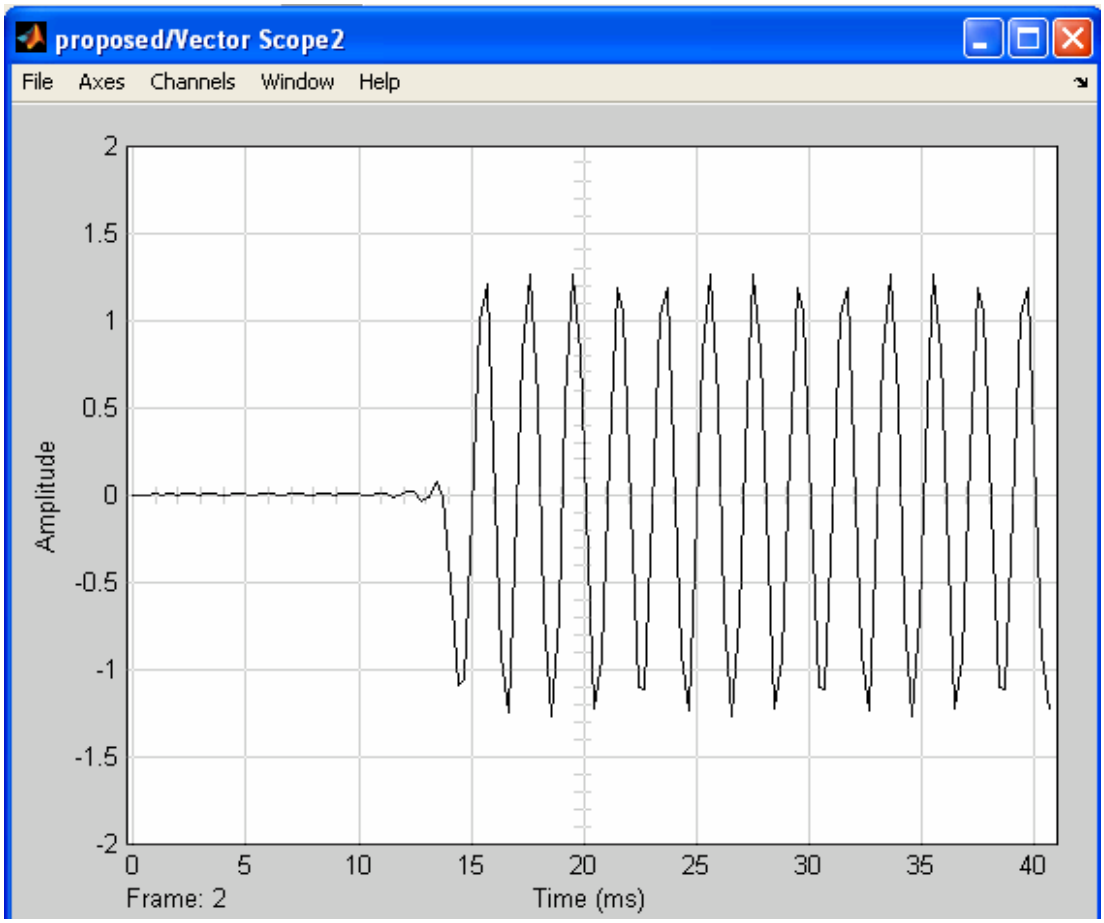

Figure 4.10 Output Data for Proposed Method for Fractional Delay of 0.15.

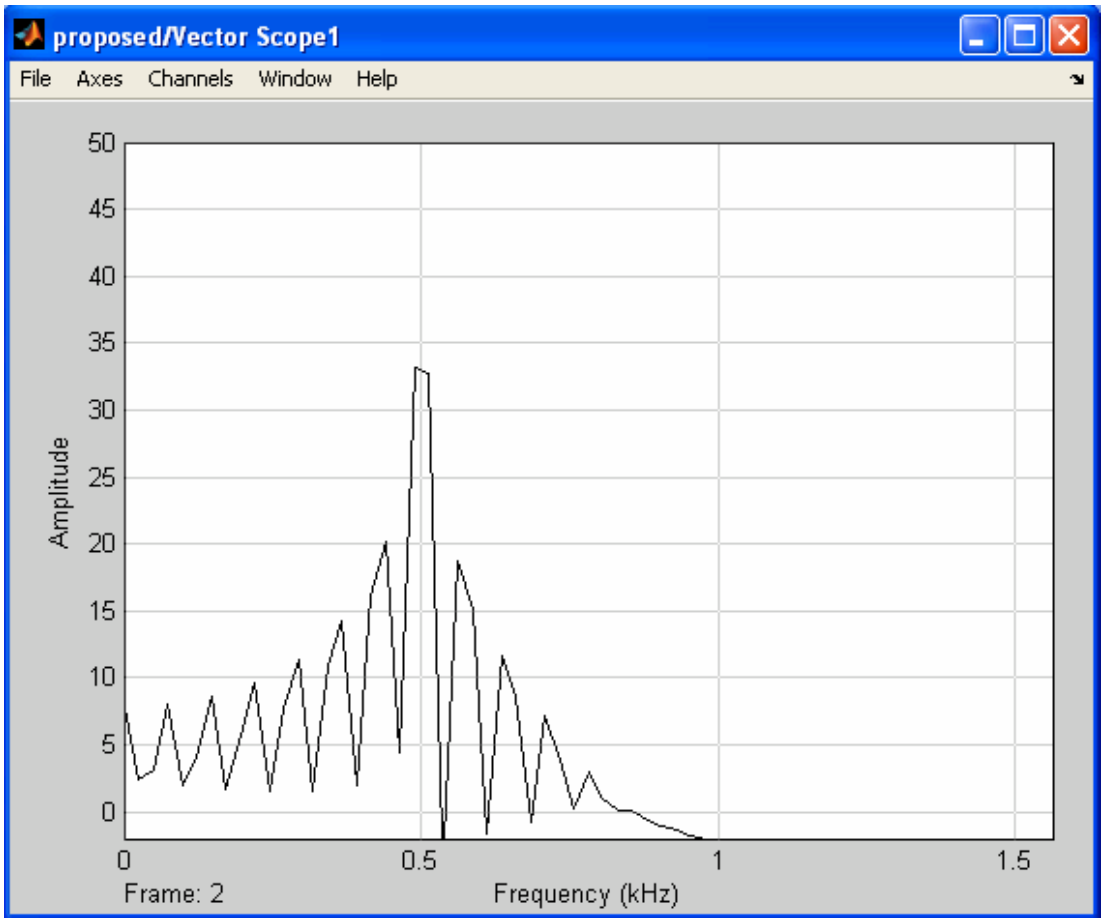

Figure 4.11 Magnitude of FFT of Output Data for Proposed Method for Fractional Delay of 0.15.

The output data waveform and magnitude of FFT for traditional method for fractional delay of 0.15 could be seen in Figure 4-12 and Figure 4-13 respectively.

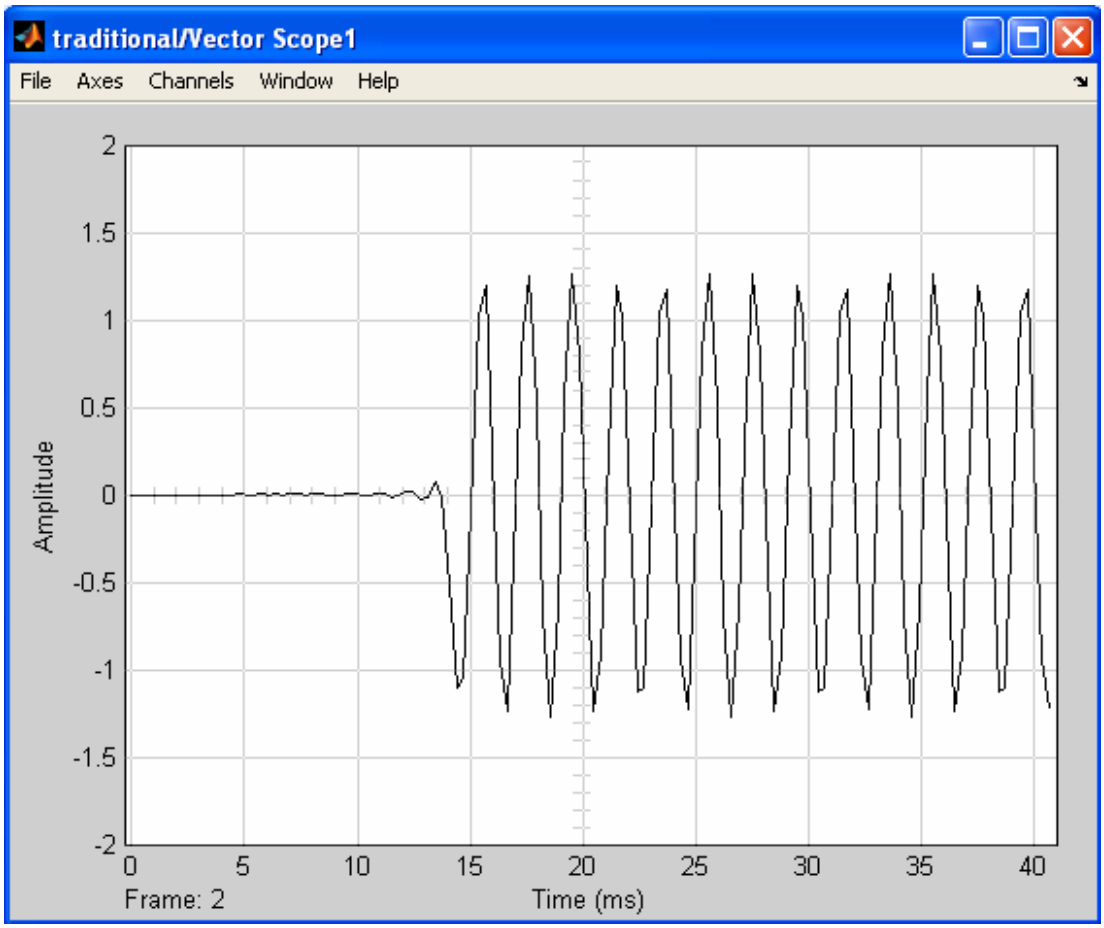

Figure 4.12 Output Data for Traditional Method for Fractional Delay of 0.15.

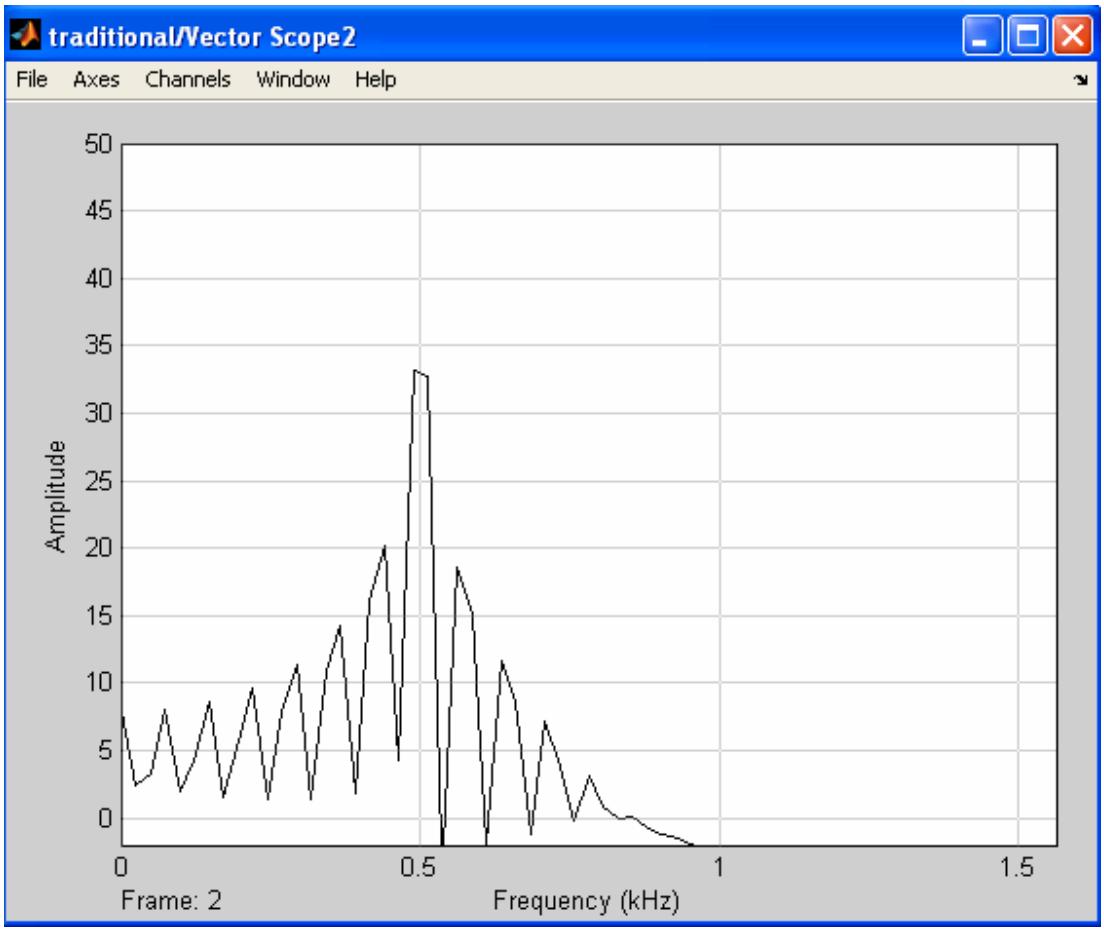

Figure 4.13 Magnitude FFT of Output Data for Traditional Method for Fractional Delay of 0.15.

The difference of the output data that of traditional and proposed method for fractional delay of 0.15 is given in Figure 4-14.

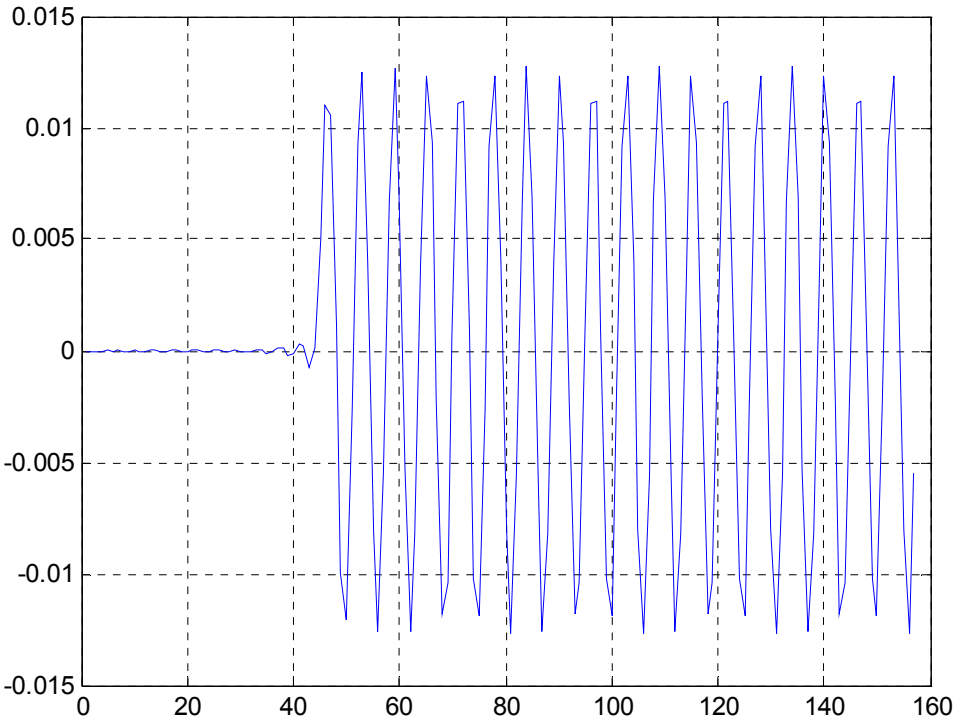

Figure 4.14 Difference of Output Data of Traditional and Proposed Methods for Fractional Delay of 0.15.

From these results, it can be concluded that there is approximately 20dB difference between proposed and traditional method between output values whereas magitude of FFT are nearly the same.

As another example point 0.5 fractional delay is considered. Corresponding waveforms for traditional method output data could be seen in Figure 4-15 and Figure 4-16 respectively.

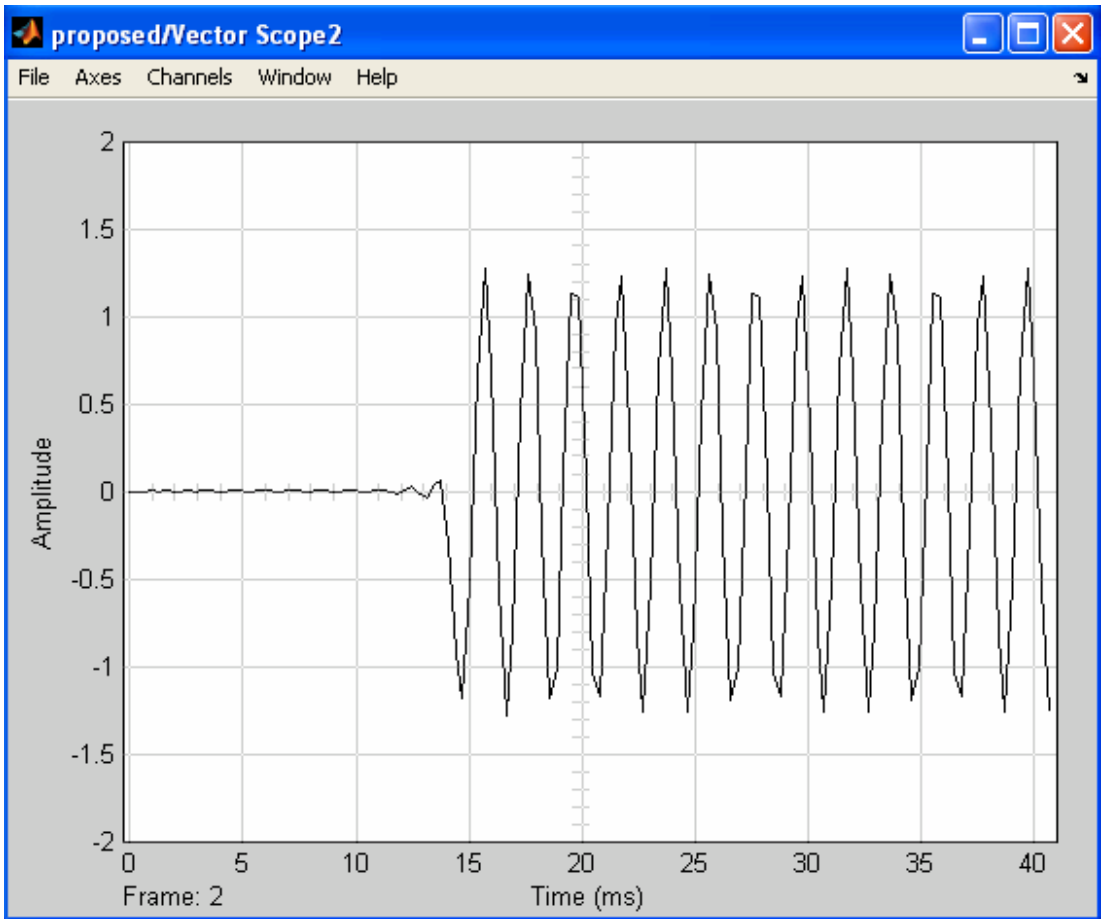

Figure 4.15 Output Data for Proposed Method for Fractional Delay of 0.5.

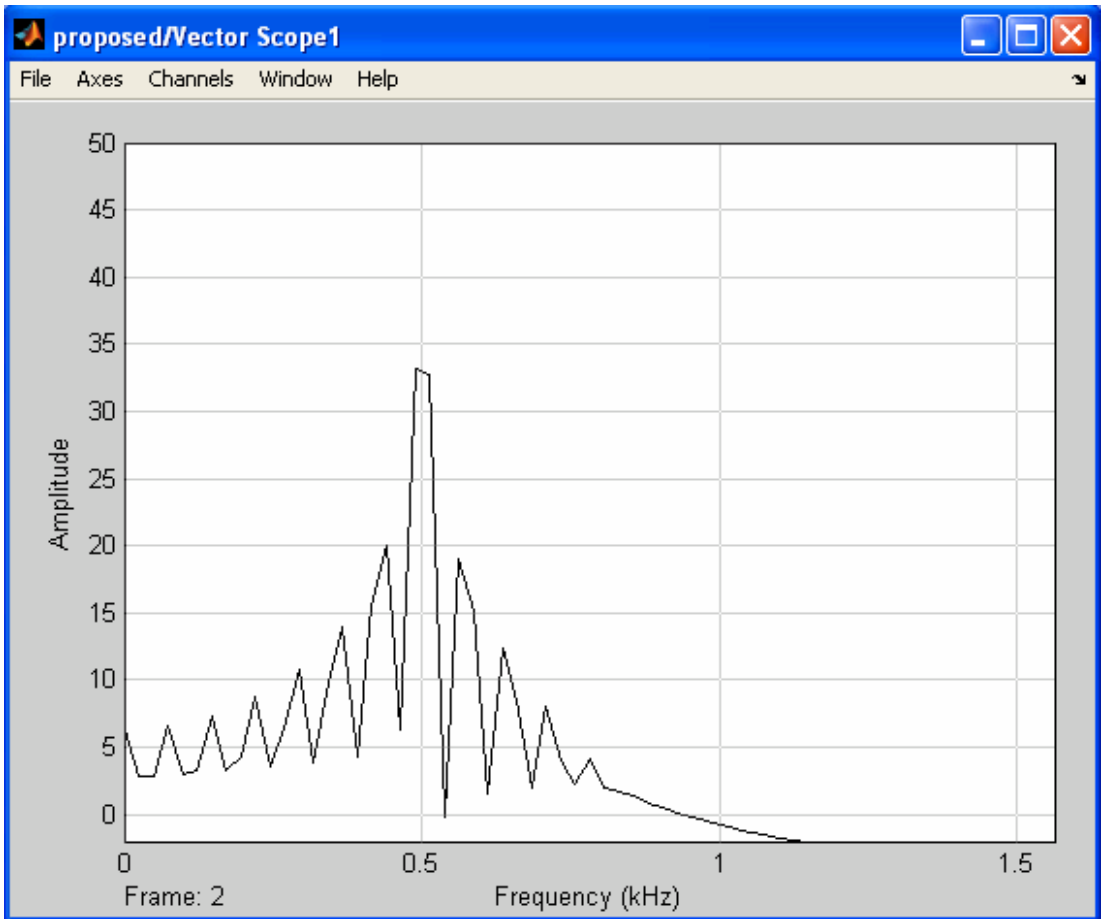

Figure 4.16 Magnitude of FFT of Output Data for Proposed Method for Fractional Delay of 0.5.

The output data waveform and magnitude of FFT for traditional method for fractional delay of 0.5 could be seen in Figure 4-17 and Figure 4-18 respectively.

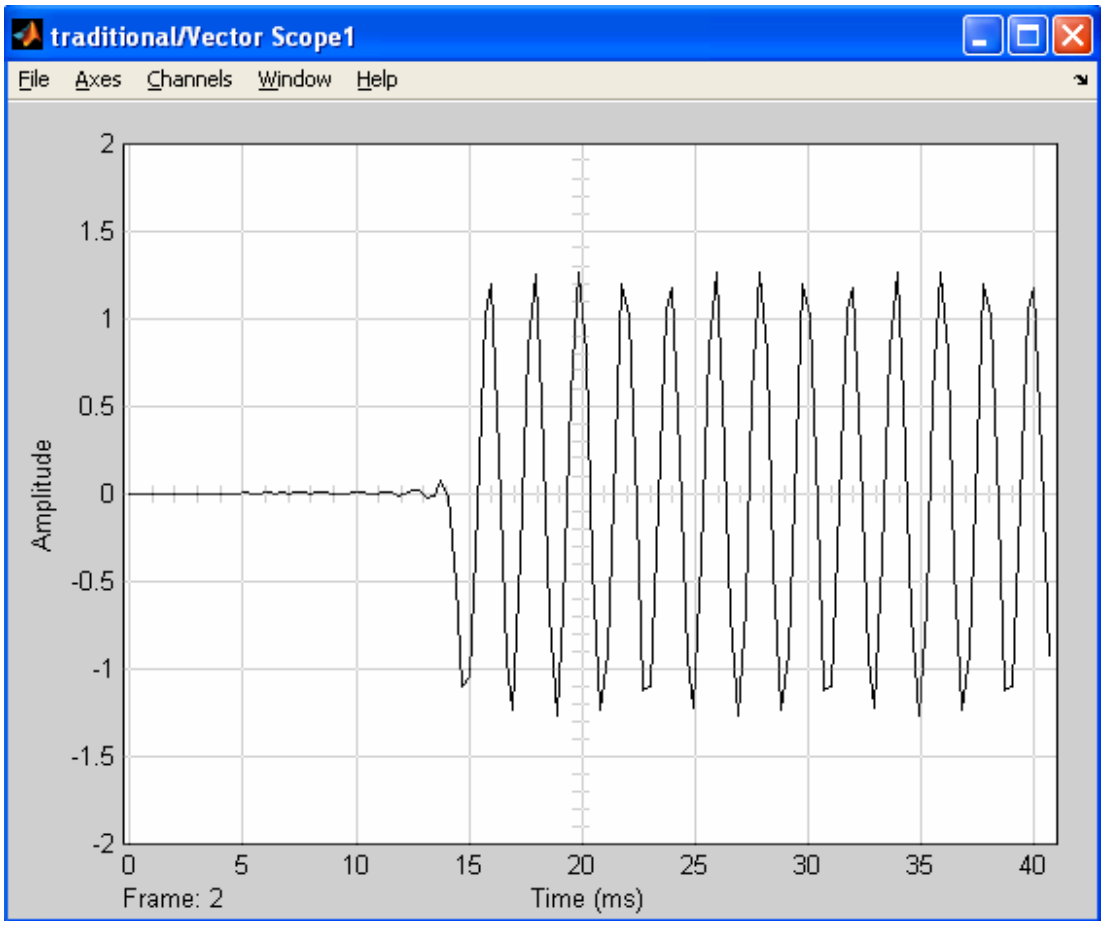

Figure 4.17 Output Data for Traditional Method for Fractional Delay of 0.5.

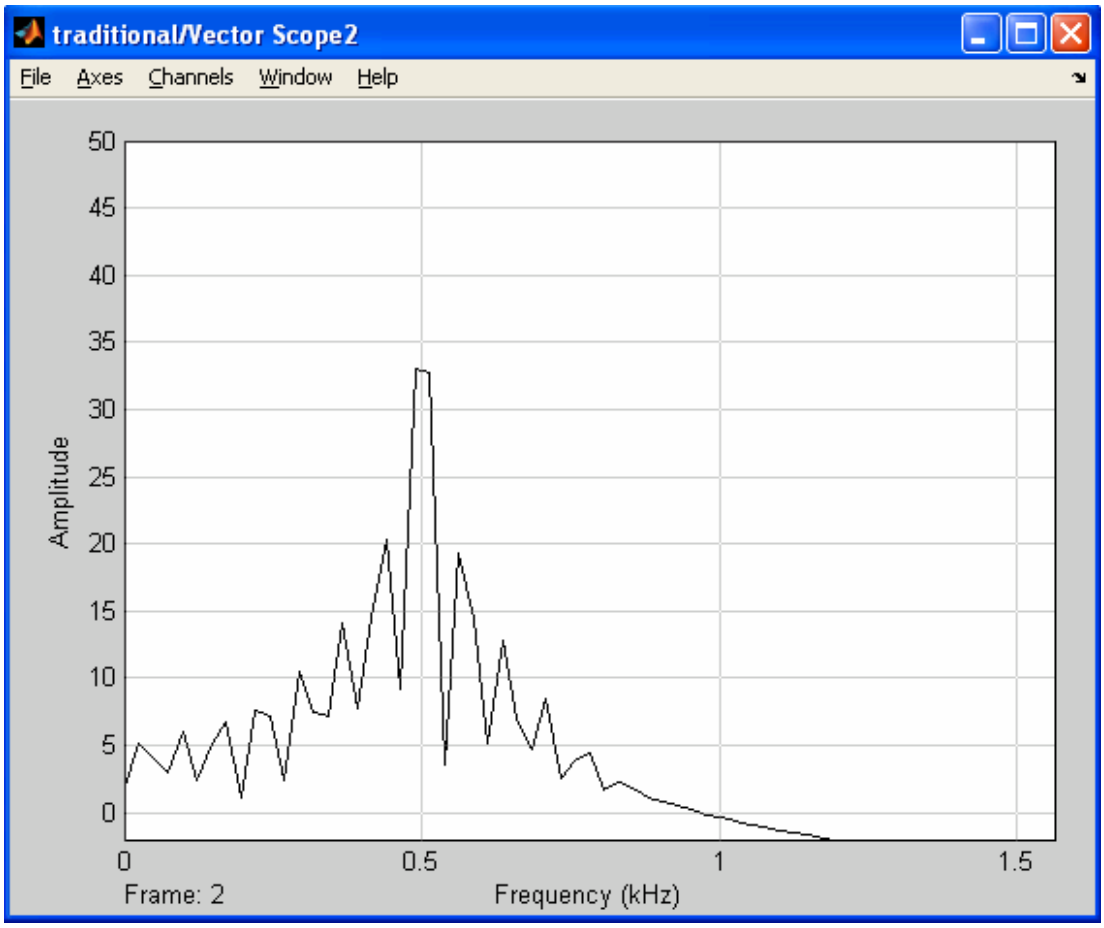

Figure 4.18 Magnitude of FFT of Output Datafor Traditional Method for Fractional Delay of 0.5.

The difference of the output data that of traditional and proposed method for fractional delay of 0.5 is given in Figure 4-19.

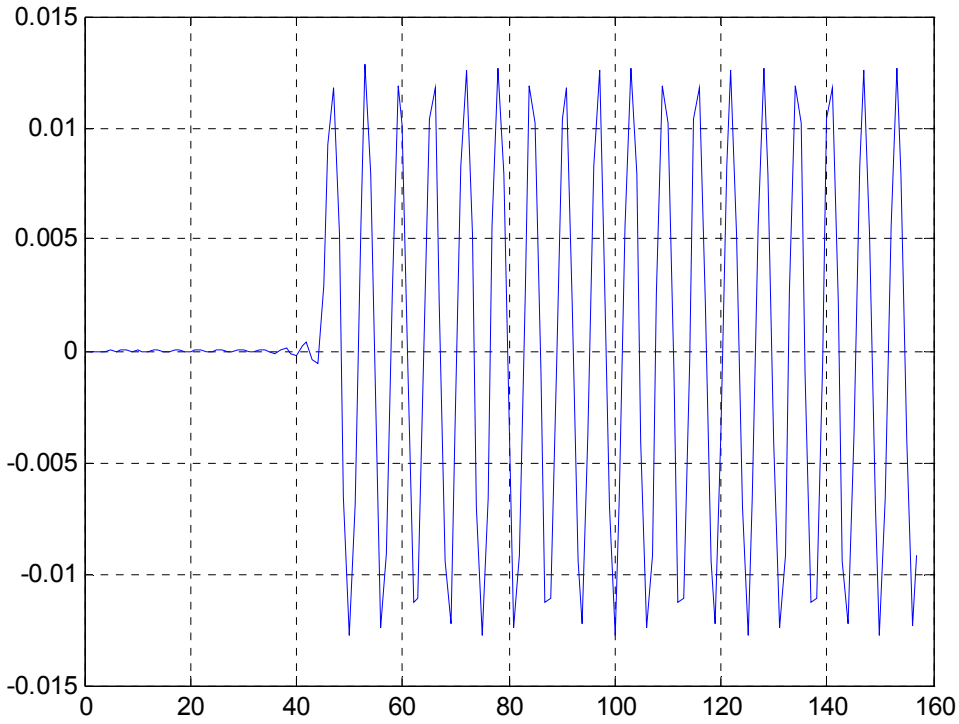

Figure 4.19 Difference of Output Data of Traditional and Proposed Methods for Fractional Delay of 0.5.

Hera again the difference between porposed and traditional method outputs are approximately 20dB for worst case.

4.4 Work on RAD-2

To give a specific example of hardware Pentland Systems RAD2 card is considered, whose detailed operational and functional description is given in Appendix A.

In order RAD-2 to operate in a desired manner, user should specify necessary operating requirements. These need to be set via PCI communication bus. Since PCI is a very complex communication protocol an operating system is necessary for this communication. RAD-2 has capability of communicating over VxWorks, Linux and WinXP.

Since VxWorks needs a proper operating case and Linux is not a commonly used operating system, WinXP is chosen for operation.

For different operating conditions different settings should be done on RAD-2. For this reason custom drivers are needed, though Pentland System supplies drivers for all of these operating systems.

WinXP does not share necessary files for communication over PCI bus, therefore an interface program Jungo Driver is used. This program supplies user the necessary files for PCI interface. By this interface a custom driver is developed by Borland Builder program. With this custom program one could read and write data management registers of RAD-2.

In order to make proper connection RAD-2 and mother board a PCI to PMC card is necessary, which could be seen in Figure 4.20 and Figure 4.21.

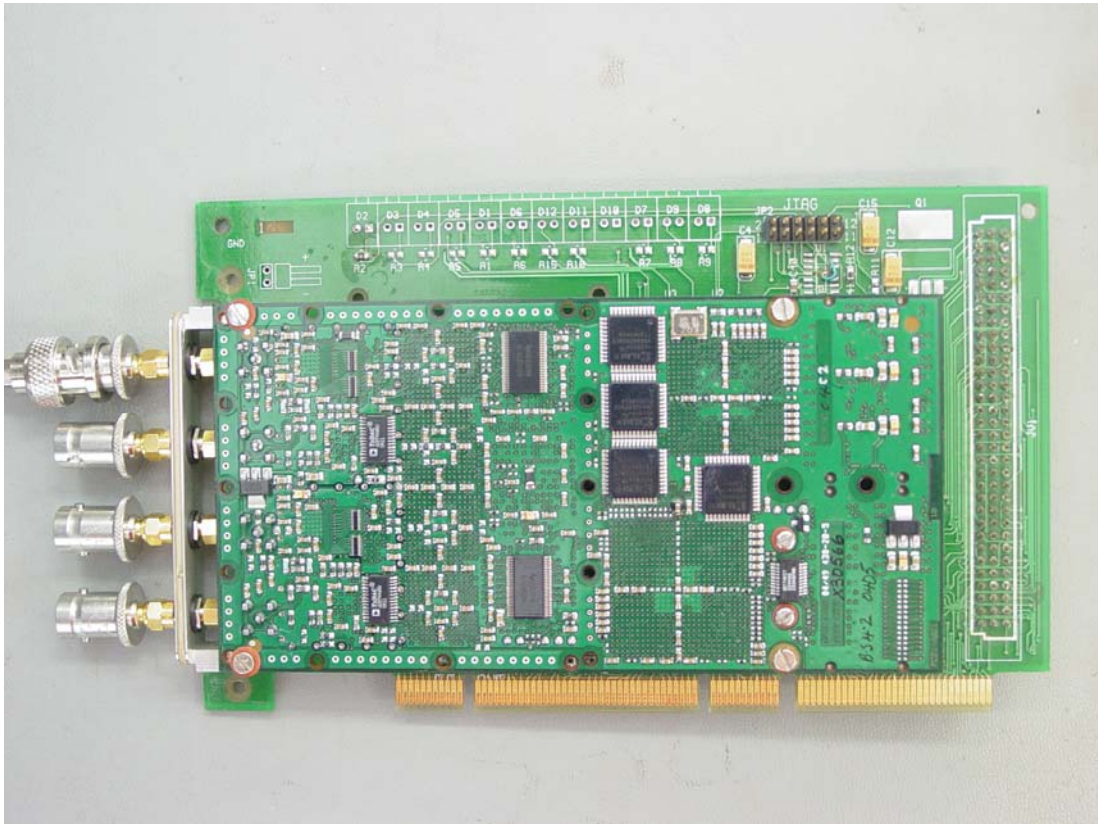

Figure 4.20 RAD-2 and PCI to PMC Card Assembly (Front Side).

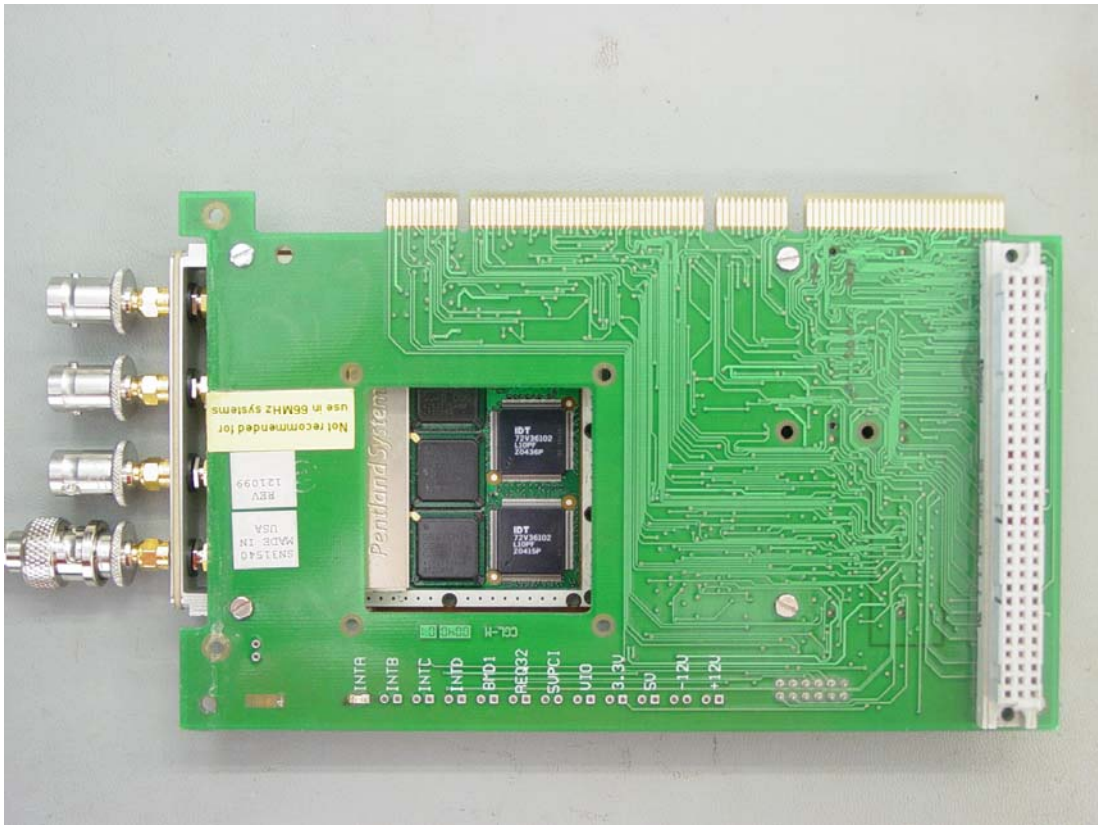

Figure 4.21 RAD-2 and PCI to PMC Card Assembly (Back Side).

The connectors that are used for analog and clock inputs are SMC type, which are not suitable for connecting signal generators, oscilloscopes or so. For this reason SMC to BNC connection units are used, seen in Figure 4.22.

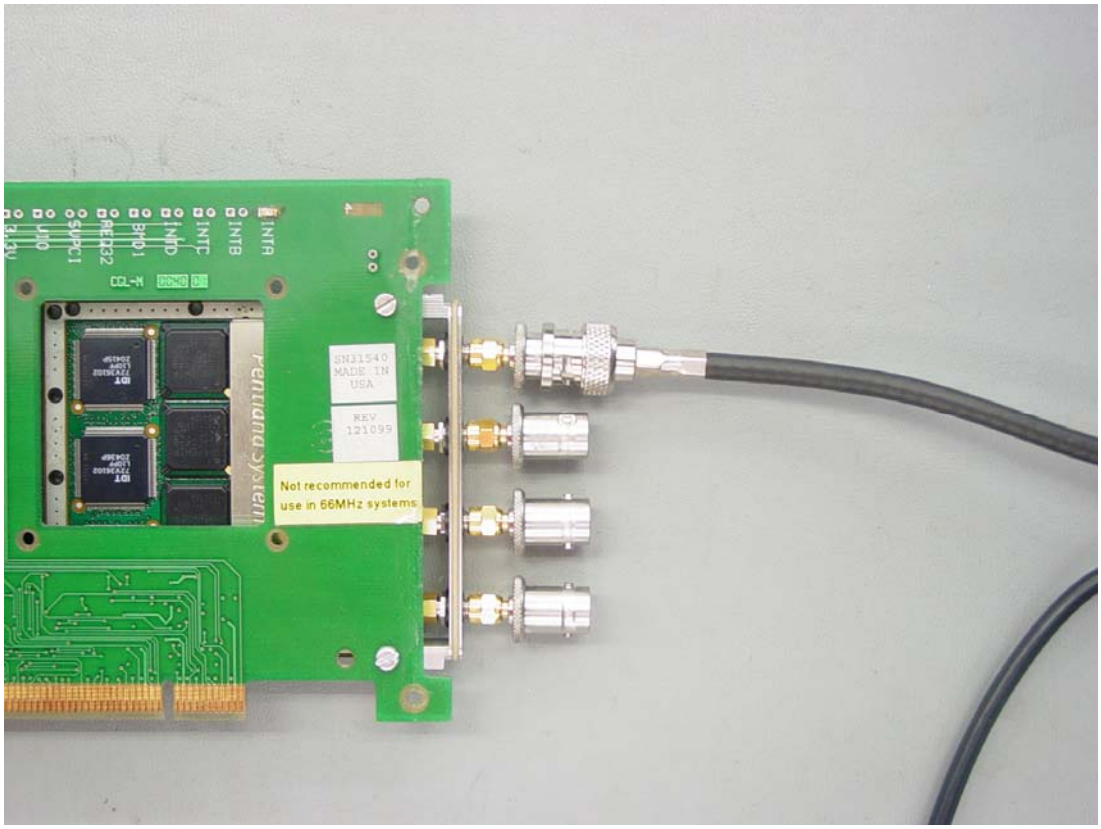

Figure 4.22 SMC to BNC Connection.

RAD-2 should be first set by PCI for proper operation. This is achieved by custon driver's "OpenBoard" function as seen in Figure 4.23.

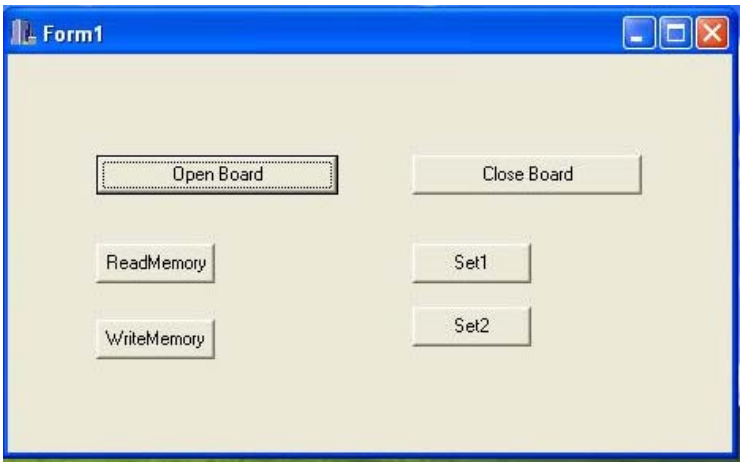

Figure 4.23 Custom Driver Program Snapshot.

For test purposes “Open Board” functionality gives a message if RAD-2 card is found in the hardware, shown in Figure 4.24.

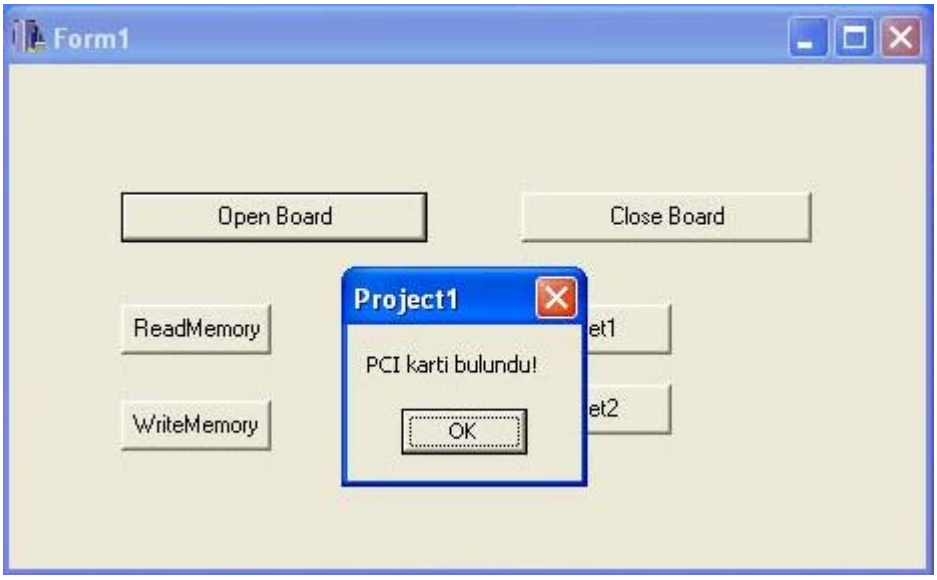

Figure 4.24 PCI Card Message Snapshot.

Here the application program utilizes the interface supported by Jungo Driver Wizard program. All of the necessary settings for operation of RAD2 is done by this application program.

For the implementation purposes, a signal generator with 5kHz, 1V p-p square signal output s connected to RAD2’ s SK1 channel. As a clock source again a signal generator with 500kHz, 1.5V p-p square wave output is connected to SK2 channel.

The aim of the implementation is to read data that is generated by DDC, Digital Downconverter Unit here GC4016 chip on RAD2, through PCI bus. For this purpose several steps shoud be performed by application program.

Here ‘ReadMemory’ and ‘WriteMemory’ buttons are used for specific read and write functions for predefined memory locations. ‘Set1’ button is used to route the data generated by ADC’s. ‘Set2’ button is used to route DDC generated data to PCI bus.
In order RAD2 to do ‘Set1’ or ‘Set2’ functionalities a series of setting should be done on RAD2.

- ✓ Chip select bits that control the ADCs, BIFIFOs and Virtex should be set, this is done by writing BAR1+0x1013 [07..00] register 0x07h.
- ✓ FIFO reset and initialization should be performed. In order to do this first Virtex sould be put in master FIFO reset mode, by setting BAR1+0x4030 [07] to logical 1. Then QL5064 and FIFOs should be resetted by setting BAR1+0x1000 [08 07] bits to logical 1.In order to reset the chips properly at least 400ns delay should be put in application. The reset bits should be set to logical 0 in the reverse order.
- ✓ DDC configuration should be done. GC4016 channels should be initialized and CFIR and PFIR coefficients should be set properly for filtering operations.

After these necessary steps mode configuration should be selected. Since RAD2 has a capability of bidirectional, 64 or 32 bit split modes communication over PCI bus, the operating mode should be selected.

FIFO controller should be resetted by asserting logical 0 to BAR1+0x1000 [24]. Then, for ‘Set1’ and ‘Set2’ button functionalities, split mode is selected by setting BAR1+0x1000 [21 20] to logical 1. In order communication be from RAD2 to PCI, that is transmit mode, BAR1+0x1000 [17 16] bits should be set to logical 1. After these settings FIFO controller reset should be removed.

Since all ADC and DDC outputs are routed to Virtex, FPGA should also be configured for proper communication. Here for ‘Set1’ functionality ADC data should be routed, that is BAR1+0x1000 [07 … 00] register should be written by 0x01h. For

‘Set2’ functionality this register should be written by 0x02. After these settings ADC should be enabled by setting BAR1+0x2000 [00] to logical 1.

To enable data transfer between the RAD-2 and host, the DMA engines need to be configured before acquisition can start. The QL5064 chip has two DMA, Direct Memory Access, transmit channels, Tx0 and Tx1, and two DMA receive channels, Rx0 and Rx1. In order to operate a DMA channel, the Write/Read address must first be loaded and then the number of bytes to be transferred loaded into the transfer count register. Finally DMA can be started in the DMA Start/Done register. This register can be monitored to check for the completion of the DMA process.

After DMA the data written by RAD-2 is read and stored in file ‘out.txt’ in ASCII format for both ‘Set1’ and ‘Set2’ functions.

CHAPTER 5

CONCLUSION

5.1 Conclusion

In this thesis, various design methods for variable digital filters is outlined. These are basically direct integration method and singular value decomposition based matrix array decomposition methods. Mathematical background and illustrative examples are given.

Application areas for variable digital filters is also mentioned. As an illustrative application method on software defined radio is chosen. Software simulations on Matlab Simulink for proposed and traditional SDR structure are performed. From these results it is seen that proposed method would give better result especially in terms of hardware that it does not need upsampling in SRC unit. Since upsampler units limits the performances of SDR proposed method could be preferrable.

After simulations on computer a hardware; Pentland Systems RAD2 card is considered. WinXP driver application is made in order to operate RAD2. The settings for the operation of the card was successfull. However when it comes to read the output data from RAD2, WinXP PCI bus could not be set for DMA operation.

5.2 Proposed Future Work

As for design improvement of variable FIR filter structures, multiplierless and reduced delay forms may be worth considering. Illustrative studies on specific hardware of SDR may be considered. Multiplierless and reduced delay formed filter structure performances may be tested on these hardware in terms of performance.

REFERENCES

[1] T.B. Deng, 'Design of Arbitrary-Phase Variable Digital Filters Using SVD-Based Vector-Array Decomposition', IEEE Transactions On Circuits and Systems-1: Regular Papers, Vol. 52 No1, Jan 2005.

[2] K.M. Tsui, K.S. Yeung, S.C. Chan, K.W. Tse, 'On The Minimax Design of Passband Linear-Phase Variable Digital Filters Using Semidefinite Programming', IEEE Signal Processing Letters, Vol. 11, No.11, Nov 2004.

[3] K.S. Yeung, S.C. Chan, 'The Design and Multiplier-Less Realization of Software radio Receivers With Reduced System Delay ', IEEE Transactions On Circuits and Systems-1: Regular Papers, Vol. 51, No. 12, Dec 2004.

[4] T.B. Deng, 'Weighted Least-Squares Method for Designing Variable Fractional Delay 2-D FIR Digital Filters', IEEE Transactions On Circuits and Systems:II Analog and Digital Signal Processing, Vol. 47, No 2, Feb 2000.

[5] K.S. Pun, S.C. Chan, K.S. Yeung, K.L. Ho, 'On The Design and Implementation of FIR and IIR Digital Filters With Variable Frequency Characterictics', IEEE Transactions On Circuits and Systems:II Analog and Digital Signal Processing, Vol. 49, No. 11, Nov 2002.

[6] F. Gardner, 'Interpolation in Digital Modems PartI: Fundamentals', IEEE Transactions On Communications, Vol. 41, No.3, March 1993.

[7] F. Gardner, R.A. Harris 'Interpolation in Digital Modems PartII: Implementation and Performance', IEEE Transactions On Communications, Vol. 41, No.6, June 1993.

[8] R. Zarour, M. Fahmy, 'A Design Technique for Variable Digital Filters', IEEE Transactions On Circuits and Systems, Vol. 36, No.11, Nov 1989.

[9] A. Oppenheim, W.F.G. Mecklenbrauker, R.M. Mersereau, 'Variable Cutoff Linear Phase Digital Filters', IEEE Transactions On Circuits and Systems, Vol. 23, No. 4, Apr 1976.

[10] D. B. H. Tay, S. S. Abeysekera, and A. P. Balasuriya, "Audio signal processing via harmonic separation using variable Laguerre filters," in Proc. IEEE Int. Symp. Circuits Syst., vol. III, Bangkok, Thailand, May 25–28, 2003, pp. 558–561.

[11] P. Kroon and B. S. Atal, "Pitch prediction with high temporal resolution," IEEE Trans. Signal Processing, vol. 39, pp. 733–735, Mar. 1991.

[12] J. S. Marques, I. M. Trancoso, J. M. Tribolet, and L. B. Almeida, "Improved pitch prediction with fractional delays in CELP coding," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., vol. 2, Albuquerque, NM, Apr. 3–6, 1990, pp. 665–668.

[13] Y. Medan, "Using super resolution pitch in waveform speech coders," in Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing, vol. 1, Toronto, ON, Canada, May 2–5, 1991, pp. 633–636.

[14] C. W. Farrow, "A continuously variable digital delay element," in Proc. IEEE Int. Symp. Circuits Syst., vol. 3, Espoo, Finland, June 6–9, 1988, pp. 2641–2645.

[15] IEEE Standard Physical and Environmental Layers for PCI Mezzanine Cards (PMC), IEEE Std 1386-2001, ANSI Approved 25 October 2001.

[16] RAD-2 Product Manual Ref: P63648 Issue P01C Pentland Systems.

[17] 14-Bit, 80/105 MSPS A/D Converter, Analog Devices, Data Sheet Rev B.

[18] 14-Bit, 210 MSPS TxDAC D/A Converter, Analog Devices, Data Sheet Rev B.

[19] GC4016 Multi-Standard Quad DDC Chip, Texas Instruments, Data Sheet Rev 1.0.

[20] Virtex-II Platform FPGAs; Xilinx, Data Sheet Rev3.4 March1, 2005.

[21] QL5064 66MHz/64-bit PCI Master/Target with Embedded Programmable Logic and Dual Port SRAM, Quick Logic, Data Sheet, Rev H.

[22] S. K. Mitra, Digital Signal Processing: A Computer-Based Approach. Singapore: McGraw-Hill 1998.

[23] T.Saramaki, Y.Neuvo, and S. K. Mitra "Design of Computationally Efficient Interpolated FIR Filters" IEEE Trans. Circuits Syst., vol. 35, pp.70-88, Jan, 1988.

APPENDIX A

RAD-2 HARDWARE SPECIFICATION

RAD-2 is a PCB card that produced by Pentland Systems Ltd. RAD-2, which is capable of communicating over 64-bit 66MHz PCI (Peripheral Communication Interconnect) and confirms to the IEEE 1386 PMC Specification [15]. RAD-2 can accept two analog inputs.

The functional block diagram of RAD-2 could be seen in Figure A.1.

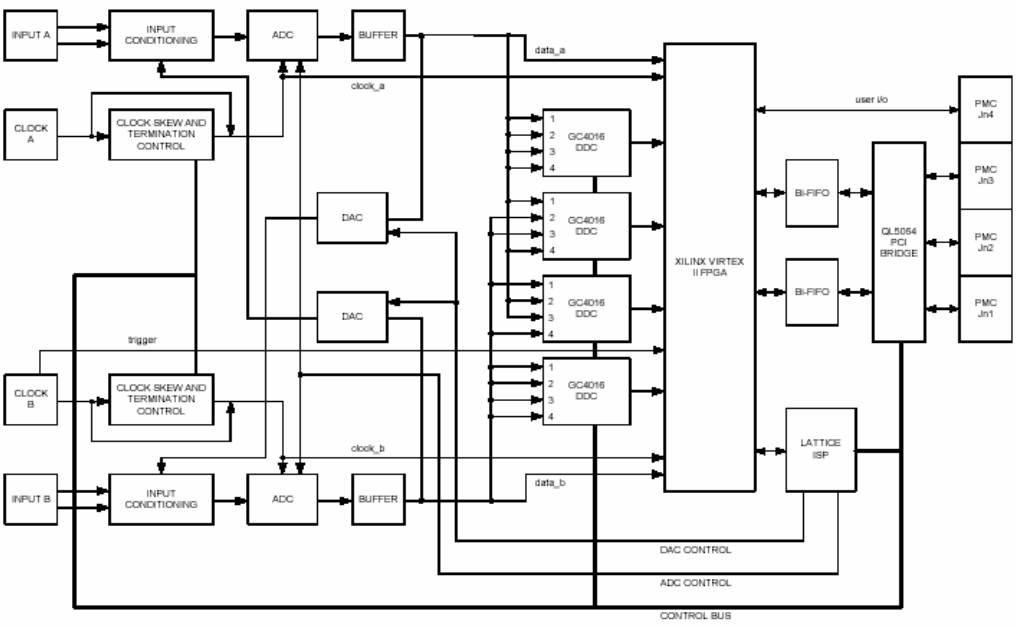

Figure A.1 Funcional Block Diagram of RAD-2.

The board layout of RAD-2 could be seen in Figure A.2.

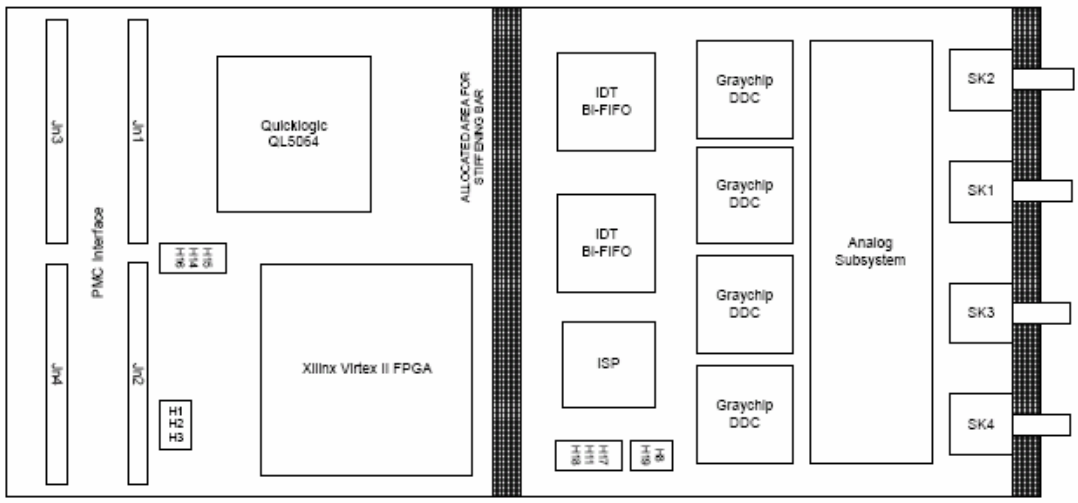

Figure A.2 Board layout of RAD-2.

(For further information about RAD-2 refer to [16])

A.1 RAD-2 Analog Input and Acquisition Subsystem

The RAD-2 has been designed to support independent acquisition channels right through to the PCI interface. Hence two independent clock inputs are provided and the backend channel management allows split FIFO transfers over the PCI bus. Alternatively, for premixed I&Q inputs or synchronised sampling, a single clock input can be used and the PCI transfers synchronised.

Input conditioning diagram of RAD-2 can be seen in Figure A.3.

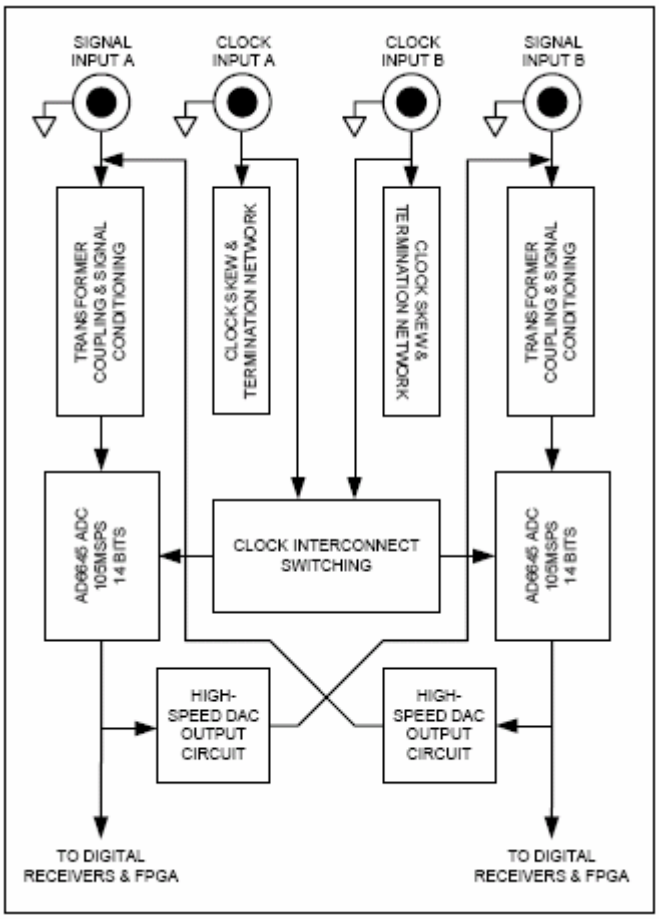

Figure A.3 Input Conditioning Diagram of RAD-2.

The high-speed analog subsystem features two ADC channels, each sampling at up to 105MSPS with 14-bits resolution and using the Analog Devices AD6645 [17] convertor.

Analog signal inputs are connected to SK1 and SK4, each of which is terminated with 50Ω. Nominal full scale input for both SK1 and SK4 is 10dBm (1V peak). With an input bandwidth exceeding 330MHz, the RAD-2 can use undersampling to acquire signals. The inputs are AC-coupled via suitable transformers.

The performance of the RAD-2 is highly dependent on a stable clock source. Channels can be clocked either simultaneously or independently. Clock signals should be connected to SK2 (& SK3) and within the range –10 to +26dBm into 50Ω.

Sinusoidal signals are preferred due to their low harmonic content. Data is sampled on the rising edge of the clock signal.

The RAD-2 analog subsystem contains two 14-bit AD9744 DACs [18]. Although the DACs have a maximum conversion rate of 160MHz, they are clocked at a maximum of 105MHz on the RAD-2.

These DAC circuits can be used to generate dithering signals to enhance the performance of the analog acquisition. With very low signal levels, e.g. ½ LSB, a dithering signal can increase the resolution of the conversion process. This dithering signal, generally in the form of band-limited noise, is then removed after A/D conversion by processing in the FPGA.

The DAC circuits also provide full testing of the analog subsystem; digital test pattern data is converted into analog form and looped back into the ADC input of the opposite channel. By using test data that contains a number of known tones spread over the full dynamic range and using the Graychips or Virtex™ II to filter out the tones, the integrity of the RAD-2 can be verified.

The DACs can also drive an analog output signal through SK1 and SK4. The nominal output signal level is 0dBm into 50Ω, AC-coupled.

A.2 RAD-2 Digital Downconversion Subsystem

RAD-2 realizes digital downconversion by means of Graychip GC4016 [19] DDCs. The GC4016 quad receiver chip contain four identical down-conversion circuits. Each downconvert circuit accepts a real sample rate up to 100 MHz, down converts a selected carrier frequency to zero, decimates the signal rate by a programmable factor ranging from 32 to 16,384 and then resamples the channel to adjust the sample rate up or down by an arbitrary factor. In the real output mode the output sample rate is doubled and the signal is output as a real signal centered at Fout/4. The channels

may be combined to produce wider band and/or oversampled outputs or to process complex input data. The chip outputs the down-converted signals in any one of several formats (microprocessor, four serial lines, one TDM serial line, nibble, LINK, or 24 bit parallel port. The chip contains two user programmable output filters per path which can be used to arbitrarily shape the received data's spectrum. These filters can be used as Nyquist receive filters for digital data transmission. The chip also contains a resampling filter to provide additional filtering and to allow the user complete flexibility in the selection of input and output sample rates. Block Diagram of GC4016 can be seen in Figure A.4.

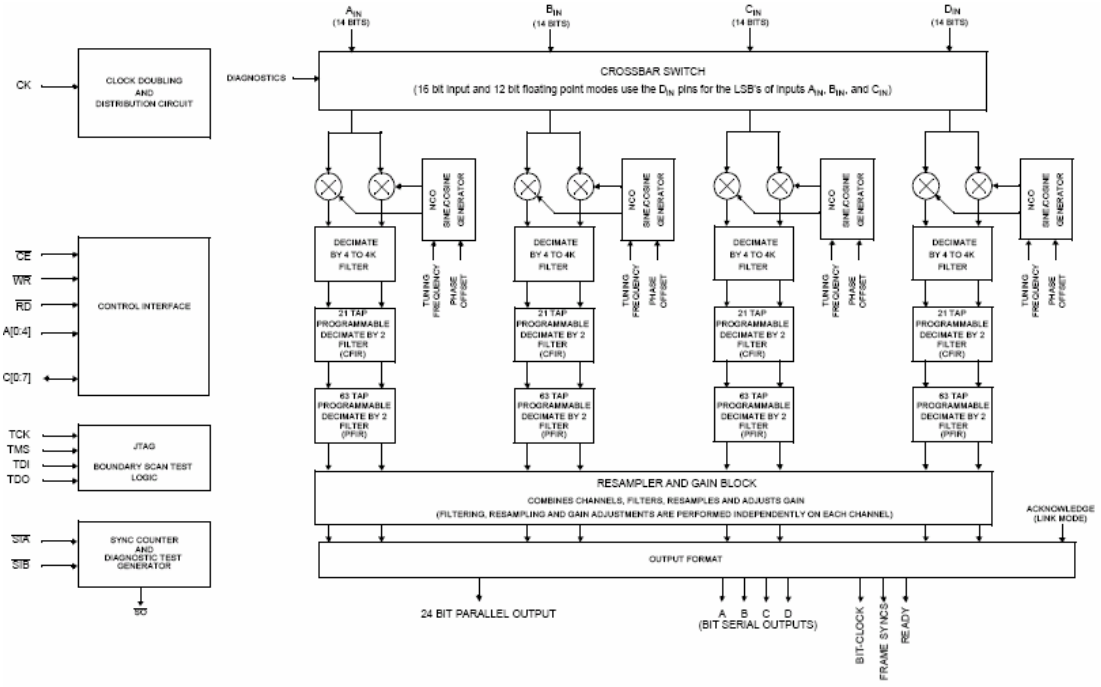

Figure A.4 Block Diagram of GC4016.

The chip is configured by writing control information into control registers within the chip. The control registers are grouped into 8 global registers and 128 pages of registers, each page containing up to 16 registers. The global registers are accessed as addresses 0 through 7. Address 2 is the page register which selects which page is accessed by addresses 16 through 31.

Each down converter contains an NCO and a mixer to quadrature down convert the signal to baseband, followed by a 5 stage Cascade Integrate Comb (CIC) filter and two stages of decimate by two filtering to isolate the desired signal. The signal is then sent to a resampler which can increase or decrease the final output sample rate to match the post-processing requirements for baud rate sampling or oversampling. Block diagram of each channel can be seen in Figure A.5.

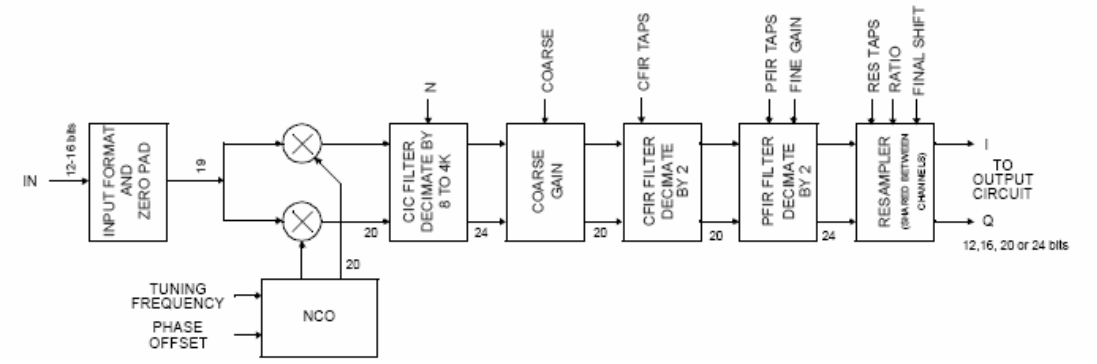

Figure A.5 Block Diagram of Down Converter Channel.

The input samples are normally clocked into the chip at the clock rate, i.e., the input sample rate is equal to the clock rate. Input rates lower than the clock rate can be accepted by using the zero pad mode. When enabled by setting the ZPAD_EN bit in address 19 of the channel control pages, the zero pad mode will insert “NZERO” zeroes between each input sample, where NZERO ranges from 0 to 15, allowing input data rates down to 1/16th the clock rate. NZERO is set in address 19.

Zero padding lowers the effective decimation ratio. For example, the minimum complex output decimation using a single channel is normally 32. If the input data rate is 5 MSPS and the system can clock the chip at 40 MHz, then the zero pad function can be used to insert seven zeros between each sample, padding the 5 MSPS input data rate up by a factor of eight to 40 MSPS. The minimum decimation of 32

from the 40MHz rate results in an output rate of 1.25 MSPS, which is an effective decimation of 4 relative to the original 5 MSPS data.

The tuning frequency of each down converter is specified as a 32 bit word and the phase offset is specified as a 16 bit word. The NCOs can be synchronized with NCOs on other chips. This allows multiple down converter outputs to be coherently combined, each with a unique phase and amplitude. A block diagram of the NCO circuit can be seen in Figure A.6.

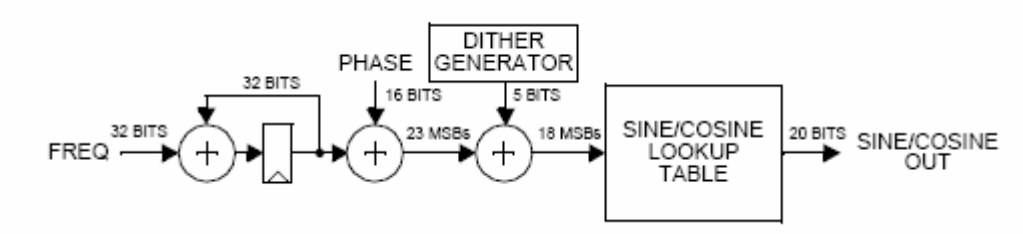

Figure A.6 NCO Block Diagram.

The tuning frequency is set to FREQ according to the formula FREQ = $232F/F_{CK}$, where F is the desired tuning frequency and F_{CK} is the chip's clock rate. The 16 bit phase offset setting is PHASE = $2^{16}P/2\pi$, where P is the desired phase in radians ranging between 0 and 2 π.

The mixer outputs are decimated by a factor of N in a five stage CIC filter, where N is any integer between 8 and 4096 (between 4 and 2048 for SPLITIQ mode). The value of N is programmed independently for each channel in addresses 21 and 22 of each channel control page. The programmable decimation allows the chip's usable output bandwidth to range from less than 4 kHz to over 3 MHz when the input rate is 100 MHz. Wider output bandwidths are obtainable by using multiple channels. A block diagram of the CIC filter is shown in Figure A.7.

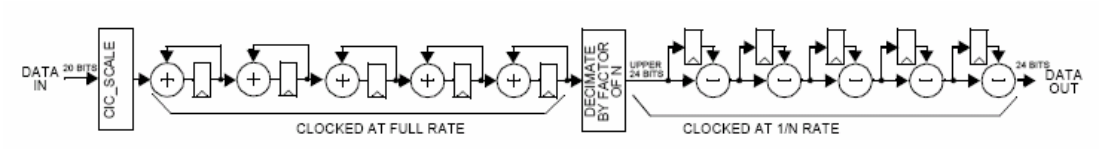

Figure A.7 Block Diagram of CIC filter.

The gain of each channel can be boosted up to 42 dB by shifting the output of the CIC filter up by 0 to 7 bits prior to rounding it to 20 bits. The coarse gain is:

$$\text{COARSE\_GAIN} = 2^{\text{COARSE}}$$

where COARSE ranges from 0 to 7. COARSE is set in address 25 of each channel control page. Overflows in the coarse gain circuit are saturated to plus or minus full scale. The coarse gain is used to increase the gain of an individual signal after the input bandwidth of the downconverter has been reduced by a factor of N in the CIC filter. If the signal power across the input bandwidth is relatively flat, as is the case in most frequency division multiplexed (FDM) systems, then one would want to boost the signal power out of the CIC filter by a factor of COARSE_GAIN = $\sqrt{N}$. Each channel can be given its own coarse gain setting.

The CIC/Coarse gain outputs are filtered by two stages of filtering. The first stage is a 21 tap decimate by 2 filter with programmable 16 bit coefficients. Since this filter decimates by two, a stopband must be created in that portion of the spectrum that would alias into the signal of interest. This filter has very lax transition band specifications so 21 taps is sufficient to both provide the required anti-aliasing stopband, and to provide compensation for the droop in the CIC filter's passband. The CFIR is also used, in some cases, to provide additional stopband rejection for the second stage PFIR filter. Figure A.8 illustrates the passband and stopband requirements of the filter. FCFIR is the input sample rate to the CFIR filter. FCFIR/4 is the output sample rate of the channel before resampling.

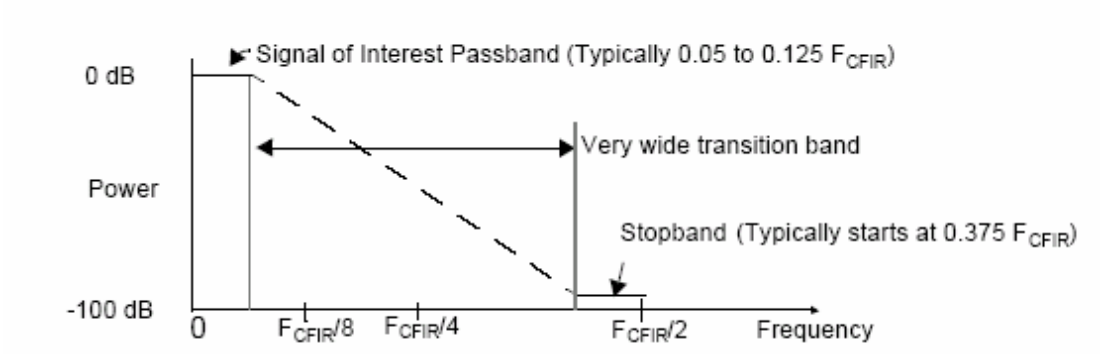

Figure A.8 CFIR Specifications.

The second stage decimate by two filter is a 63 tap decimate by 2 filter with programmable 16 bit coefficients. Fine gain is applied at the output of the PFIR and rounded to 24 bits. Overflows are detected and hard limited. Overflows can be directed to the channel overflow detection block.

The PFIR filter passband must be flat in the region of the signal of interest, and have the desired out of band rejection in the region that will alias into the signal's bandwidth after decimation. Figure A.9 below illustrates the passband and stopband requirements of the filter. FPFIR is the input sample rate to the PFIR filter. FPFIR/2 is the output sample rate of the channel before resampling.

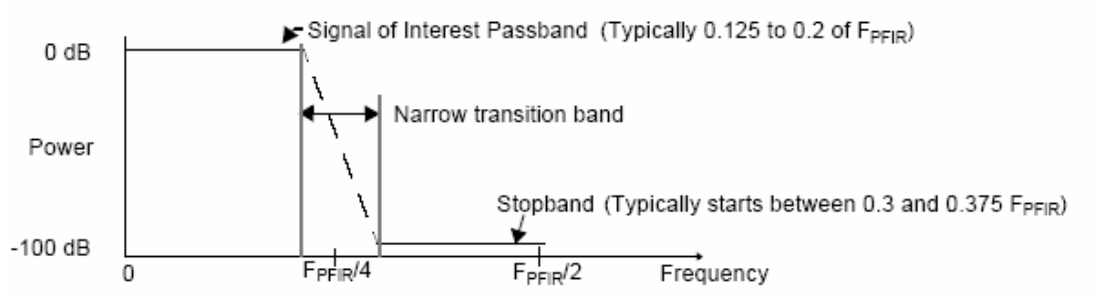

Figure A.9 PFIR Specifications.

The externally downloaded coefficients can be used to tailor the spectral response to the user's needs. For example, it can be programmed as a Nyquist (typically a root-raised-cosine) filter for matched filtering of digital data. The user downloaded filter coefficients are 16 bit 2's complement numbers. Unity gain will be achieved through the filter if the sum of the 63 coefficients is equal to 65536. If the sum is not 65536, then PFIR will introduce a gain equal to

$$PFIR\_GAIN = \frac{PFIR\_SUM}{65536}$$

where PFIR_SUM is the sum of the 63 coefficients.

The resampler will independently filter and change the data rate of each channel. The most common application of the resampler is to increase the sample rate of the data so that it will match a desired symbol or bit rate. Demodulators for digital modulation schemes, such as GMSK, QPSK, QAM or CDMA, for example, require sample rates which are 1X, 2X, 4X or 8X times the bit or symbol rate of the modulation. In these cases, the maximum down converter filter performance is achieved when the PFIR output rate is around 1.5 to 2 times the signal's bandwidth1. The resampler is then used to increase the sample rate up to the required 2X, 4X or 8X rate.

The resampler can also be used as an additional filter to optimize the passband or stopband response of the channel.

The resampler consists of an input buffer, an interpolation filter, and a final shift block. A functional block diagram of the resampler is shown in Figure A.10.

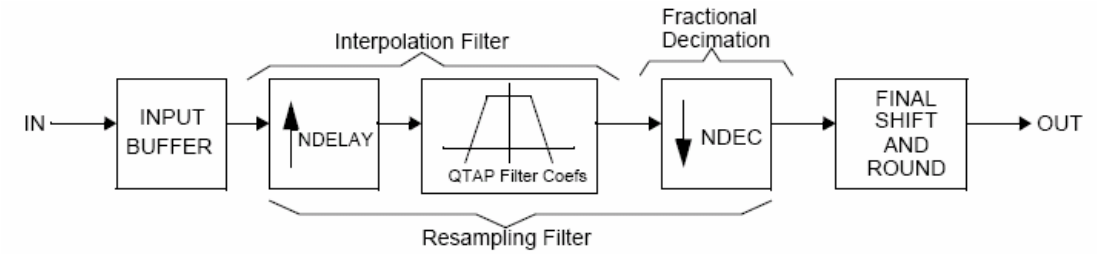

Figure A.10 Resampler Channel Block Diagram.

The resampler's sampling rate change is the ratio NDELAY/NDEC. The decimation amount NDEC is a mixed integer/fractional number. When NDEC is an integer, then the exact sampling instance is computed and there is no phase jitter. If NDEC is fractional, then the desired sampling instance will not be one of the possible NDELAY interpolated values. Instead the nearest interpolated sample is used. This introduces a timing error (jitter) of no more than 1/(2*NDELAY) times the input sample period.

The input buffer accepts 24 bit data from the four input channels, and adds them as necessary to form 1,2, or 4 resampler channels (see the ADD_TO control bits in address 21 of the resampler control page). The input buffer serves both as a FIFO between the channels and the resampler, and as a data delay line for the interpolation filter. The 64 complex word input buffer can be configured as four segments of 16 complex words each to support 4 resampler channels, or as two segments of 32 complex words each to support 2 resampler channels, or as a single segment of 64 complex words to support a single resampler channel. The number of segments is set by NCHAN in address 16 of the resampler control page.

The interpolation filter zero pads the input data by a factor of NDELAY and then filters the zero padded data using a QTAP length filter. The output of the QTAP filter is then decimated by a factor of NDEC.

The three spectral plots shown in Figure A.11 illustrate the steps required to resample the channel data. The first spectral plot shows the data just after zero padding. The sample rate after zero padding is NDELAY*F_S, where F_S is the sample

rate into the resampler. The second spectrum shows the shape of the QTAP filter which must be applied to the zero padded data in order to suppress the interpolation images. The last spectrum shows the final result after decimating by NDEC.

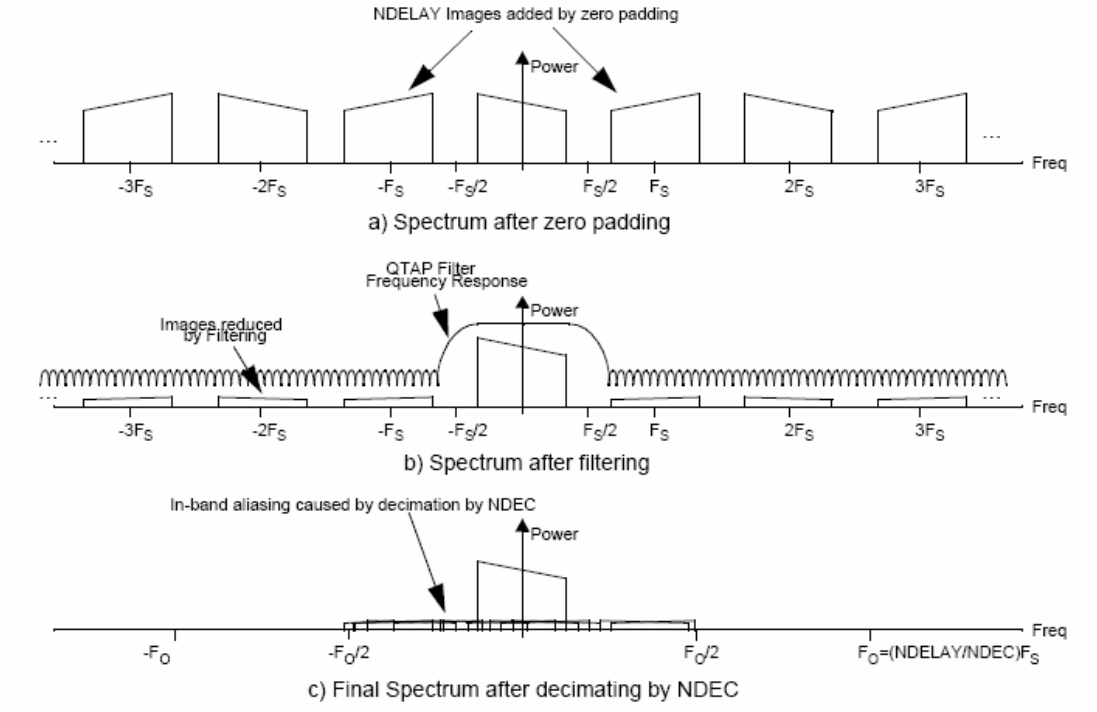

Figure A.11 Resampler's Spectral Response.

On RAD-2, two ADC outputs are connected to GC4016 as seen in Figure A.12.

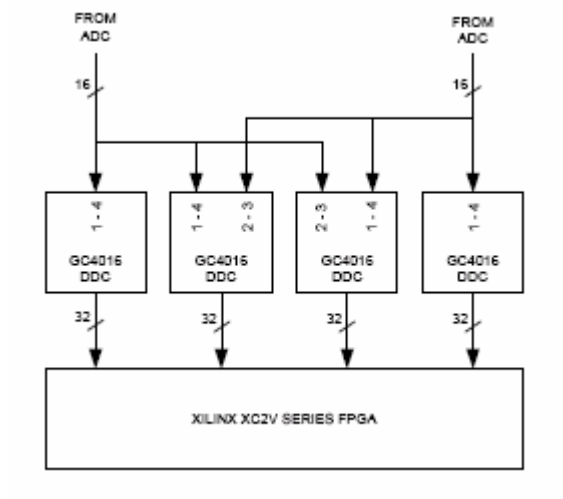

Figure A.12 RAD-2 ADC to Graychip Connection.

If Graychip processing is not required, ADC-generated data can bypass the Graychips and be fed directly to the FPGA.

A.3 RAD-2 Flow Control

RAD-2 data flow control is realized by Xilix Virtex II FPGA [20] (Field Programmable Gate Array). The Virtex II FPGA controls the FIFO data buffering and the transient lengths, trigger level, front porch lengths, gate and trigger modes of the ADCs. A series of 32-bit internal registers are directly accessible over the PCI bus for configuration and control of these functions.

The majority of the signals on the RAD-2 are routed to the Virtex II FPGA, which are Graychip DDC outputs, Data Bus A; with ADC0 generated data, DAC1 input and Graychip DDC Inputs, Data Bus B, with ADC1 generated data, DAC0 input and Graychip DDC Inputs, Clocks A and B and bi-FIFO for PCI data transfers.

The FPGA can be programmed either by the on-board PROMs, or via the JTAG signals over the PCI bus.

A.4 RAD-2 Data Management and Buffering

The RAD-2 has two Bi-FIFOs for host transfers, each with 32-bit width and 64k depth. This equates to 128k samples per input channel and a total of 512kB on-board storage.

The FIFO controller in the PCI bridge allows these FIFOs to operate in either synchronised 64-bit mode, or split 32-bit/32-bit mode. The synchronised mode enables full 64-bit transfers over the PCI bus, with both channels linked together. In split 32/32 mode, each FIFO acts independently and can support different data transfer rates for the two acquisition channels simultaneously.

The multiple DMA engines in the PCI interface can support both interleaved and noninterleaved data transfers. These DMA engines may be dedicated to a specific input channel or, alternatively, one DMA engine can provide a chain DMA capability.

A.5 RAD-2 PMC Interface

The 64-bit 66MHz PCI interface on the RAD-2 provides extremely flexible host communication, with over 500Mbytes/s nominal bandwidth and full interrupt functionality. Standard JTAG signals for test chaining and FPGA programming are available.

The QL5064 PCI bridge device [21] allows direct access to the register sets implemented within the RAD-2. These register sets exist in three separate devices; The QL5064 itself, for PCI bus and FIFO buffer management, Lattice ISP device, for

ADC enabling, DAC enabling and Virtex II Programming, Virtex II FPGA, for comprehensive control of the analog acquisition functionality.

Additionally; the PMC J4 connector has various application specific configurations. If the front panel connectors SK2 and SK3 are used for clocks, then signals on J4 can be configured to act as triggers or gating signals for the analogue data. Alternatively, the J4 connector can be configured as a digital data path, which enables inter-board synchronization and custom data interfacing for proprietary buses. The routing of these signals to external connectors is dependent on the host board noise coupled to these signals, which can affect the analog performance of the unit.