JPL Publication 94-20

Parallel Digital Modem Using Multirate Digital Filter Banks

Ramin Sadr
P. P. Vaidyanathan
Dan Raphaeli
Sami Hinedi

August 15, 1994

NASA

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Abstract - A new class of architectures for an all-digital modem is presented in this report. This architecture, referred to as the parallel receiver (PRX), is based on employing multirate digital filter banks (DFBs) to demodulate, track, and detect the received symbol stream. The resulting architecture is derived, and specifications are outlined for designing the DFB for the PRX. The key feature of this approach is a lower processing rate than either the Nyquist rate or the symbol rate, without any degradation in the symbol error rate. Due to the freedom in choosing the processing rate, the designer is able to arbitrarily select and use digital components, independent of the speed of the integrated circuit technology. PRX architecture is particularly suited for high data rate applications, and due to the modular structure of the parallel signal path, expansion to even higher data rates is accommodated with ease. Applications of the PRX would include gigabit satellite channels, multiple spacecraft, optical links, interactive cable-TV, telemedicine, code division multiple access (CDMA) communication, and others.

Table of Contents

iv

List of Figures

List of Tables

I. Motivation

With the evolution of high speed satellite and terrestrial communications, the applications of high data rate communication systems are becoming abundant. Existing earth orbital missions such as the Telecommunication and Data Relay Satellite System (TDRSS) support data rates of up to 300 Mbps. Communication systems must today process faster and handle an ever rising data throughput.

Advances in digital integrated circuit (IC) technology have made switching speeds close to 1GHz possible. However, the widespread use of high speed components is costly both in price and power consumption. One of the key bottlenecks in DSP design for an all-digital receiver is the availability of components (e.g. multiply-accumulator) that process each sample at the input sampling rate, when the latter exceeds 200 MHz or so. The objective here is to explore a cost effective solution to this problem. The ideal solution is to employ lower speed (50-70 MHz) components using IC technologies such as the Complementary Metal Oxide Semiconductor (CMOS) technology. CMOS has many known advantages such as low cost, low power, and high density. The data acquisition technology also has undergone rapid advancements, where today, one giga-sample per second analog-to-digital (A/D) converters are emerging. By using a single high speed A/D component and a low number of high speed components (e.g. multiplexers only), a fundamental question is posed:

Is it possible to architect a digital receiver such that the processing rate is slower than both the sampling and the symbol rate?

The answer to the above question is "yes". In this work, we devise a new approach for designing a digital receiver that trades off processing rate with parallelism. Our presentation is largely based on the evolving disciplines of multirate signal processing and digital filter bank (DFB) theory. Classically, the filter banks have been used for subband coding applications. Using the filter bank theory for designing the digital receiver, the resulting system is an all-purpose receiver which is suited for a variety of different modulation formats for high data rate channels. Our general approach is also suited for other multi-channel communication applications such as multi-carrier modulation systems, multiple spacecraft communication (or users), and spread spectrum communication systems.

Another important by-product of our approach is that the overall architecture of the receiver is modular in the sense that if a higher data rate is desired, the same hardware (at IC or board level) may be replicated and deployed without the need to redesign the whole system. In our previous work [1], we succeeded in formulating a parallel digital phase locked loop (PDPLL). This work formed the basis of the results presented here, and it was expanded to provide a cohesive approach to designing a digital receiver.

II. Introduction

With the rapid growth of the VLSI technology coupled with the flexibilities of digital signal processing techniques, a key objective in the design of the receiver is to implement the system digitally. However, it is generally not feasible (even though more desirable) to sample the received signal directly at the radio frequency (RF). Thus, we assume the availability of an intermediate stage for open loop down-conversion of the RF signal to a convenient frequency for the A/D conversion, referred to as the intermediate frequency (IF).

An all-digital receiver is understood here as a receiver that performs: demodulation, matched filtering, carrier synchronization, and symbol timing recovery. All other required functions (e.g. lock indicators, power estimators, etc.) in a conventional receiver use either the in-phase and quadrature components, or the output of the matched filters derived from these components. The in-phase and quadrature components of the receiver form the sufficient statistics for other estimators and lock detectors. Thus, we do not outline the implementation of these functions. The merits of an all-digital approach versus the analog implementation are widely known. A new generation of all-digital systems has been successfully deployed in NASA's Deep Space Network [2]. These receivers support a symbol rate of up to a few mega-symbols per second. Here, we propose a new approach for the design of the future generation of receivers, that could be potentially used for high data rate applications up to giga-symbols per second.

A typical digital receiver is shown in Fig. 1. The input signal $x(t)$ is sampled and converted to a discrete time sequence by an analog-to-digital (A/D) converter with a uniform sampling period of T_s seconds. The output of the matched filter in this block diagram is the estimated symbol sequence (complex or real). In a coded communication system, this sequence is used by the channel decoder (e.g., Viterbi decoder) to estimate the transmitted bit stream.

The symbol duration hereon is denoted as T seconds, the sampling period is denoted by T_s seconds, and the single-sided input bandwidth of the equivalent lowpass prefilter prior to A/D conversion in the receiver is denoted by W in Hz as shown in Fig. 2. We assume a soft decision symbol output for generality.

Some fundamental choices we made in the architecture of the digital receiver depicted in Fig. 1 are:

1. The receiver employs bandpass sampling for A/D conversion of the received signal. It is noted that there are other alternative versions of this I&Q receiver structure. A widely used alternative (baseband sampling) is used to perform the demodulation in the analog domain, prior to A/D conversion, and then sample the I&Q components separately in the baseband, using two separate A/Ds. In bandpass sampling, a single A/D converter is used. In reference [3], these alternative approaches are investigated, and it is concluded that bandpass sampling is more suited for space communication. In bandpass sampling (see Fig. 2), the minimum sampling rate is $f_s = 4W$ and the center frequency of the anti-aliasing filter prior to the A/D conversion is positioned at $f_c^{IF} = (2k+1)W$ for some integer k. Usually the integer k is chosen to result in a convenient center frequency for designing the anti-aliasing filter. Due to practical limitations in filter design [e.g. surface acoustic wave (SAW)], off-the shelf filters are only available in a finite range of center frequencies.

2. The phase tracking loop is closed in the digital domain. The merits for this loop closure versus the digital-to-analog conversion and closing the loop in the IF section are discussed in [3].

It is noted that our approach can also be applied with minor modification when baseband sampling is employed for A/D conversion, or the phase tracking loop is closed in the analog domain. The processing rate in our scheme is not limited by the minimum sampling rate, as exhibited later in this report. In the conventional approach, the processing rate (shown as a one sided arrow in the bottom left hand side of Fig. 1) in the digital signal processing building blocks following the A/D converter is at the minimum processing rate $1/(2T_s)$. We seek to parallelize the structure shown in Fig. 1 such that the processing rate throughout the receiver can be arbitrarily selected by the designer. This selection is only limited by the amount of resources (hardware) available for fabricating the receiver, and it is **not** dictated by the input sampling rate **or** the speed of the ICs used in the signal path.

The architecture of PRX derived in this paper is of the form depicted in Fig. 3. Here M denotes the decimation rate in each subband. In this architecture the input signal is parallelized into $2M$ separate signal paths. The input signal is filtered using a Discrete Fourier Transform (DFT) based analysis and synthesis filter bank, augmented with parallel equivalent of the matched filtering operation. The resulting output of the overall system is the detected output symbol sequence. The key feature of this implementation is the parallelization of the input signal and processing of the input samples at the rate of $1/(MT_s)$, illustrated in Fig. 3 as a single sided arrow. In the foregoing, we outline the derivation and the design of the filter banks for transforming the structure shown in Fig. 1 into the final form depicted in Fig. 3.

II.1 Outline

Section II.2 begins with an introductory section on detection of signals in additive white Gaussian noise (AWGN) channel. Section II.3 contains some relevant results from multirate signal processing that are used in the sequel. **Sections II.2 and II.3 may be skipped by those readers familiar with these subjects.** Section III describes the design and architecture for the parallel implementation of the demodulation and filtering using multirate filter banks. In Section IV, the derivation of and our approach to combined digital matched filtering and demodulation are discussed. Section IV.2 contains a design approach for digital matched filtering using interpolated finite impulse response filtering; this section is independent of the other section in Section IV and may be used independently of other results in this report. Sections V and VI contain the design for the symbol timing recovery and carrier tracking in the PRX, respectively. Section VII discusses alternative architectures. Section VIII contains a brief discussion of the processing delay of the PRX. In Section IX, simulation results for a 16-

channel PRX are provided. Section X outlines some future direction for research in this area. Section XI includes some concluding remarks. The four appendices, namely A,B,C, and D contain respectively the MathCad™ software for generation of the filter banks, interpolated finite impulse response (IFIR) filter design for matched filtering, the receiver block diagram, and the C-source code programs for generating the polyphase components of the filter banks used in simulation of the PRX.

II.2 Detection of Signal in AWGN

The received waveform $x(t) = s(t) + n(t)$ is composed of signal $s(t)$ plus noise $n(t)$, specifically

$$x(t) = \sum_k a_k\, p(t - kT) \cos(2\pi f_c t + \theta) + n(t) \quad (1)$$

where $a_k \in \mathrm{U}$ is the symbol sequence and for binary phase shift keying (BPSK) $\mathrm{U} = \{-1, +1\}$. We assume full response signaling where the pulse shape $p(t)$ satisfies $p(t) = 0$ for $t \notin [0, T)$. Here $n(t)$ is additive white Gaussian noise (AWGN) with a single sided spectral density N_0. The optimum coherent receiver for this received signal is well known [5]. We outline here some of the results used in the sequel that follows.

The optimum receiver for detection of signals, with a known waveform, in AWGN is based on maximizing the cross correlation of the known waveform and the received waveform. Formally, it can be shown that maximizing the a-posterior probability density of the received signal $\Pr\left(a_k = \alpha_m \mid r(t), t \in [kT, (k+1)T)\right)$ for equally likely and independent symbols results in maximizing the metric $\mu(\alpha_m)$ during the k-th signaling interval [5], where in general $m = 1, \cdots, |U|$ (the notation"$|\cdot|$" used on a set denotes the cardinality of the set). Formally, the metric $\mu(\alpha_m)$ is

$$\mu(\alpha_m) = \mathrm{Re}[e^{-j\theta} < x, s_m >] \quad (2)$$

where Re[.] stands for taking the real part, and the inner product is defined by

$$< x,s_m >= \int_{kT}^{(k+1)T} x(t)s_m(t)\,dt \qquad (3)$$

Both $x(t)$ and $s_m(t)$ are represented in complex baseband. Each possible transmitted signal waveform $s_m(t)$ is

$$s_m(t) = \alpha_m p(t-kT) \quad \text{for each } \alpha_m \in \mathrm{U}, \qquad (4)$$

for $t \in [kT,(k+1)T)$, and for BPSK $m = 1,2...$ The optimum detector formally computes (3) during the k-th signaling interval for each value of α_m, and picks the maximum, i.e. the detected symbol is $\hat{a}_k = \max_{\alpha \in \mathrm{U}}^{-1} \mu(\alpha)$. The matched filtering as defined in (3) for computing $\mu(a_m)$ is depicted in Fig. 4. For BPSK signals with $a_k \in \mathrm{U}$, the correlation is performed only once for each symbol, since $s_m(t)$ differs only in sign during each signaling interval. In the remaining part of this report, we use $\mu(\hat{a}_k)$ to denote the matched filter output during the k-th signaling period.

The carrier phase $\theta(t) \approx \theta$ in the received signal is a slowly varying random process and is estimated using a Costas loop [5]. This estimate is used in the voltage controlled oscillator (VCO) for generating the reference signal, $e^{j(2\pi f_c t+\hat{\theta})}$, as depicted in Fig. 5. The Costas loop inputs are the in-phase and quadrature components of the received signal, which respectively are

$$y_m^I(t) \triangleq \mathrm{Re}[a_k p(t-kT)e^{j\theta} + n(t)] \quad y_k^Q(t) \triangleq -\mathrm{Im}[a_k p(t-kT)e^{j\theta} + n(t)], \qquad (5)$$

for $t \in [kT,(k+1)T)$. The Costas loop for tracking the phase of the BPSK signal is shown in Fig. 5a. Throughout this document, we assume that all signal paths are complex. The equivalency of the complex version and the classical model is evident from Fig. 5b.

In the digital counterpart of the above formulation for the optimum receiver [6,7], when the time bandwidth product of the system is large (i.e $WT>>1$ is equivalent to high sampling rate), the inner product in (3) can be approximated by

$$< x,s_m >= \sum_{n \in \Gamma} x_n w_n^k, \qquad (6)$$

where $\Gamma = \{n : kT \le nT_S < (k+1)T\}$, $x_n = x(nT_S)$ and $w_n^k = p(nT_S - kT)$. The dimension of this summation, $N=|\Gamma|$ is the number of samples per symbol and is bounded by $N_{\max} = \lceil T/T_s \rceil$. For simplicity, assume that the symbol period is an integer multiple of the sampling period so $N = N_{\max} = T/T_s$. In a digital system, the input bandwidth W and the sampling frequency f_S must be chosen such that the number of samples per symbol $N \ge 2WT$. In the baseband model, the received signal is sampled and processed as shown in Fig. 6. An anti-aliasing filter, with bandwidth W-Hertz, is used for prefiltering the signal. Subsequently, an A/D converter converts the signal into a discrete time sequence $x_n = x(nT_s)$. Due to prefiltering and A/D conversion errors, particularly when the time bandwidth product of the system is small, the sampled signal waveform which must be used in equation (6) is different from the one derived from the ideal pulse shape $p(t)$. When the pulse shape $p(t)$ is bandlimited by the prefilter, the filtering operation manifests itself as amplitude distortion. In particular, this distortion is significant when using a rectangular pulse shape for non-return-to-zero (NRZ) or bi-phase (also referred to as Manchester) signals. In general, the effect of prefiltering, amplitude, or phase distortion can be compensated for by the discrete time sampled version of the matched filter as discussed in [7].

Bandpass sampling is employed in our approach as motivated earlier in this section. In Fig. 7, the input signal is prefiltered by the anti-aliasing filter with the analog frequency response $H^a(s)$. The filtered signal is converted by a single A/D converter to a discrete time sequence $x_n = x(nT_S)$. Demodulation is performed by the multiplication of the reference carrier signal and the input signal, which is then filtered by the lowpass filter $H^d(z)$ to reject the double frequency components [4]. In the remaining part of this report we drop the superscript d and simply refer to this filter as $H(z)$. The output of this filter is used by the matched filter for detecting the transmitted symbols. Hence, the structure of the optimum detector is performed in two separate stages, namely the demodulation and matched filtering stages. The purpose of the decimator at the output of the demodulation filter and more detailed discussion of the structure in Fig. 7 can be found in Sections III and IV.

Our results to parallelize the signal path and utilize an efficient receiver architecture can be expanded to various I&Q modulation formats, including higher dimensional modulation schemes for bandwidth efficient modulations such as multiple phase shift keying (MPSK), continuous phase modulation (CPM), and partial response signaling techniques. These other schemes could all be cast into a framework to use our methodology for parallelizing the receiver.

II.3 Multirate Signal Processing and DFB Preliminaries

In this section, a brief overview of the results used in this work from multirate signal processing is presented. Our notations and approach to multirate systems and filter bank theory follow [8].

Decimation and Expansion

Decimation and *expansion* are basic operations in multirate digital systems, as shown in Fig. 8. The output of the decimator $x^d(n)$, and the output of the expander $x^e(n)$ in frequency and time domain respectively are [8]:

$$x^e(n) = \begin{cases} x(n/L) & \text{if } n \text{ is multiple of } L \\ 0 & \text{otherwise} \end{cases} \quad \Leftrightarrow X^e(z) = X(z^L)$$
$$x^d(n) = x(Mn) \qquad \Leftrightarrow X^d(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{1/M} W_M^k) \tag{7}$$

where $W_M = e^{-\frac{2\pi}{M}j}$. When expanding a signal, the original sequence is padded with $L-1$ zeros in the time domain between each sample of the original sequence, which is equivalent to compressing the original spectrum by a factor of M. This is immediately evident when z is replaced by $e^{j\omega}$, as illustrated in Fig. 9. The process of decimation or discarding of $M-1$ samples in the time domain is equivalent to stretching the original spectrum $X(e^{j\omega})$ by an amount M; creating $M-1$ copies of this stretched version, shifting it uniformly by multiples of 2π, and then adding the stretched and shifted versions (divided by $1/M$).

In this paper, all decimation and expansion rates are positive integers, and the notation $A(z)_{\downarrow M}$ denotes the z-transform of the decimated sequence $a(nM)$ and $A(z)_{\uparrow M}$ denotes the z-transform of the expanded sequence $a\left(\frac{n}{M}\right)$.

Blocking

Blocking of a discrete time sequence is the key to parallelizing digital filtering operation. The idea behind blocking of a discrete time sequence is illustrated in Fig. 10.

This is referred to as the "*commutator*" model. The commutator is simply a switch (or a multiplexer) that rotates at a uniform rate and takes M positions periodically. That is, each subsequence $x_i(n)$ can be written in terms of $x(n)$ as follows

$$x_i(n) = x(nM + M - i), \qquad (8)$$

for $i = 1, \cdots, M$. The blocking approach for the commutator model is shown for both the decimator and the expander model. The equivalency of the switching model and the delay chain operation is evident for clockwise and counterclockwise operation in each case. The "*blocked version*" is denoted by the vector

$$\mathbf{x}_B(n) = \begin{bmatrix} x(nM + M - 1) \\ x(nM + M - 2) \\ \vdots \\ x(nM) \end{bmatrix} = \begin{bmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_M(n) \end{bmatrix}. \qquad (9)$$

Following the same notation as in [8], the z-transform of $\mathbf{x}_B(n)$ is defined as

$$\mathbf{X}_B(z) = \sum_{n=-\infty}^{\infty} \mathbf{x}_B(n) z^{-n}. \qquad (10)$$

A useful result for our application in the demodulation stage is to **interchange** the operation of multiplication with the decimation operation as shown in Fig. 11. This commutative property is a direct consequence of blocking operation shown in Fig. 10.

Notation and Abbreviation for Multi-Input and Output Systems

Let *H(z)* denote the transfer function of an arbitrary digital filter, i.e.

$$H(z) = \sum_{n=-\infty}^{\infty} z^{-n} h(n), \qquad (11)$$

where $\{\cdots, h(-2), h(-1), h(0), h(1), h(2), \cdots\}$ is the impulse response of the filter. The input and output of this filter are related by $Y(z) = H(z)X(z)$, where $X(z)$ and $Y(z)$ are the z-transforms of the input and the output respectively.

Boldface symbols represent matrices and vectors. The notations $\mathbf{A}^T$, $\mathbf{A}^*$, and $\mathbf{A}^+$ represent respectively, transpose, conjugate, and transpose conjugate of $\mathbf{A}$. An M-input-M-output system with the transfer function matrix [8] $\mathbf{H}(z) = \sum_n \mathbf{h}(n) z^{-n}$ can be defined such that input and output of the system are related by $\mathbf{Y}_B(z) = \mathbf{H}(z)\mathbf{X}_B(z)$. We use the notation "~"

$$\tilde{\mathbf{H}}(z) = \mathbf{H}^+(1/z^*) = \sum_n \mathbf{h}^+(-n) z^{-n}. \qquad (12)$$

which stands for transposition "T", followed by conjugation "*" of coefficients, followed by replacement of *z* with $(z^*)^{-1}$.

Polyphase Components

Two important results from multi-rate signal processing are the Noble identity and the polyphase decomposition. It is possible to represent $H(z)$ as defined in (11) in terms of its M-component polyphase form

$$H(z) = \sum_{l=0}^{M-1} z^{-l} E_l(z^M). \qquad (13)$$

Here $E_l(z)$ is called the l-th polyphase component of $H(z)$. The sequence $e_l(n)$, the inverse z-transform of $E_l(z)$, is defined as follows

$$e_l(n) = h(nM + l) = h_l(n), \qquad (14)$$

with $0 \leq l \leq M\text{-}1$. The expansion in (13) is simply the decomposition of $\{h(n)\}$ into M sub-sequences $e_l(n)$. For example, by grouping the impulse response coefficients $h(n)$ into even- and odd-numbered samples, i.e., $e_0(n) = h(2n)$ and $e_1(n)=h(2n+1)$, the transfer function $H(z)$ may be represented as

$$H(z) = E_0(z^2) + z^{-1}E_1(z^2), \qquad (15)$$

where

$$E_0(z) = \sum_{n=-\infty}^{\infty} h(2n)z^{-n}, \qquad (16)$$

and

$$E_1(z) = \sum_{n=-\infty}^{\infty} h(2n+1)z^{-n}. \qquad (17)$$

An important consequence of this representation is that when the polyphase component is followed by a decimation operation, then the filtering operation and the decimation can be commuted. This property, known as the Noble identity, is depicted in Fig. 12.

Applying the Noble identity to the polyphase representation of (13), the filter $H(z)$ followed by a decimator can be re-drawn, as shown in Fig. 13.b. In the model shown in Fig. 13.b, the processing rate in each polyphase component is a factor of M slower than the sampling clock. The polyphase representation results in an efficient rearrangement of the computations of the filtering operation. This effectively distributes the computations into a set of parallel filters operating at a lower speed. This in turn, reduces the speed constraints on the digital signal processing hardware, thereby enabling it to process samples at a much lower rate than the sampling rate.

The *polyphase identity* is depicted in Fig. 13.c. This identity is used when the filter $H(z)$ is preceded by an expander and then followed by a decimator, such that the expansion and decimation rates are equal. The cascaded system is a linear time-invariant (LTI) system. For verifying the LTI property note that the input to the decimator has the z-transform $X(z^M)H(z)$ and the output signal has the z-transform $[X(z^M)H(z)]\big|_{\downarrow M} = X(z)[H(z)|_{\downarrow M}] = X(z)E_0(z)$, where $E_0(z)$ is the 0-th polyphase component of $H(z)$.

Digital Filter Bank

A *digital filter bank* [8] is a collection of digital filters, with a common input or a common output. Classically, applications of filter banks have mainly concentrated in the areas of signal compression. In these applications, the output of the system is a reconstructed signal $\hat{x}(n)$ from the subband signals. The M decimated subband signals $x_k(n)$ are derived from the decimated input signal $x(n)$. Filter banks may typically have overlapping or non-overlapping bandwidths, depending on the desired characteristics. The system in Fig. 14 is called a *maximally decimated* [8] analysis / synthesis filter bank, and the set of filters $\{H_k(z), k=0,\cdots,M-1\}$ are the analysis filters and the set of filters $\{F_k(z), k=0,\cdots,M-1\}$ are the synthesis filters.

The output z-transform of the filter bank can be written as a function of the z-transform of the input as follows:

$$\hat{X}(z) = \frac{1}{M}\sum_{l=0}^{M-1} X(zW^l)\sum_{k=0}^{M-1} H_k(zW_M^l)F_k(z), \qquad (18)$$

where $W_M = e^{-j2\pi/M}$. This can be written more compactly as

$$\hat{X}(z) = \sum_{l=0}^{M-1} X(zW^l)\, A_l(z), \text{ where } A_l(z) = \frac{1}{M}\sum_{k=0}^{M-1} H_k(zW^l)F_k(z). \qquad (19)$$

It is clear that if $A_l(z)=0, \forall l>0$, then aliasing is canceled and $\hat{X}(z)=T(z)X(z)$, where $T(z)$ is referred to as the distortion function and it is given by

$$T(z) \triangleq \frac{1}{M}\sum_{k=0}^{M-1} H_k(z)F_k(z). \qquad (20)$$

For a perfect reconstruction filter bank $T(z) = c\,z^{-n_o}$. In this case the distortion function is simply a delay and a gain, and the system is free from aliasing, amplitude distortion and phase non-linearity (or distortion).

The filter bank as depicted in Fig. 14, is called a *maximally decimated filter bank* when the number of subband filters is equal to the decimation and expansion rate M. The input and output of this system are related by the polyphase decomposition of each subband filter. Let

$$\mathbf{h}(z)=\begin{bmatrix} H_0(z) \\ \vdots \\ H_{M-1}(z) \end{bmatrix}, \quad \mathbf{E}(z^M)=\begin{bmatrix} E_{00}(z^M) & E_{01}(z^M) & \cdots & E_{0,M-1}(z^M) \\ \vdots & \vdots & & \\ & & \ddots & \\ E_{M-1,0}(z^M) & & & E_{M-1,M-1}(z^M) \end{bmatrix}, \qquad (21)$$

where $E_{i,j}(z)$ denotes the j-th polyphase component of the i-th analysis filter $H_i(z)$. Accordingly, for the synthesis filter bank, define

$$\mathbf{f}(z)=\begin{bmatrix} F_0(z) \\ \vdots \\ F_{M-1}(z) \end{bmatrix}, \quad \mathbf{R}(z^M)=\begin{bmatrix} R_{00}(z^M) & R_{01}(z^M) & \cdots & R_{0,M-1}(z^M) \\ \vdots & \vdots & & \\ & & \ddots & \\ R_{M-1,0}(z^M) & & & R_{M-1,M-1}(z^M) \end{bmatrix}, \qquad (22)$$

and let the delay chain be denoted by $\mathbf{e}(z)$ where

$$\mathbf{e}(z)=\begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-M+1} \end{bmatrix}. \qquad (23)$$

Then, we can write

$$\begin{aligned} \mathbf{h}(z) &= \mathbf{E}(z^M)\mathbf{e}(z) \\ \mathbf{f}^T(z) &= z^{-(M-1)}\mathbf{e}^T(z^{-1})\mathbf{R}(z^M) \end{aligned}. \qquad (24)$$

The equivalent polyphase representation of the filter bank is illustrated in Fig. 15.

In general, to design filter banks free of any distortion, perfect reconstruction property is desired. The necessary and sufficient condition [8] for perfect reconstruction is that $\mathbf{R}(z)$ and $\mathbf{E}(z)$ must be of the form

$$\mathbf{R}(z)\mathbf{E}(z) = c\,z^{-m_o}\begin{bmatrix} \mathbf{0} & \mathbf{I}_{M-r} \\ z^{-1}\mathbf{I}_r & \mathbf{0} \end{bmatrix}, \qquad (25)$$

for some r such that $0 \le r \le (M-1)$. It is noted here when the polyphase matrix $\mathbf{E}(z)$ is paraunitary, it satisfies

$$\mathbf{E}(z)\tilde{\mathbf{E}}(z) = \mathbf{I}, \quad (26)$$

and if $\mathbf{R}(z) = \tilde{\mathbf{E}}(z)$, then (25) is automatically satisfied.

DFT Filter Bank

The Discrete Fourier Transform filter bank is a special class of filter bank. This class of filter banks can be viewed as a set of equally spaced bandpass filters obtained by modulating a prototype lowpass filter. The $M \times M$ DFT matrix $\mathbf{W}$ has elements $[\mathbf{W}]_{km} = W^{km}$ where $W = e^{-j2\pi/M}$. A DFT filter bank is defined such that every filter $H_k(z)$ is related to a single prototype $H_0(z)$ via $H_k(z) = H_0(zW^k)$. Around the unit circle, the k-th subband filter is the shifted version $H(e^{j(\omega-(2\pi k/M))})$ of $H(e^{j\omega})$. We can express the set of the M impulse responses $h_k(n)$ as

$$h_k(n) = h_0(n)\,W^{kn}. \qquad (27)$$

The simplest DFT filter bank is simply an analysis filter bank with

$$H_k(z) = H_0(zW^k), \quad (28)$$

where

$$H_0(z) = 1 + z^{-1} + \cdots + z^{-M+1}, \quad (29)$$

as shown in Fig. 16.

For an arbitrary prototype filter with a polyphase decomposition as in (13), the polyphase decomposition of the DFT filter bank is

$$H_k(z) = H_0(zW^k) = \sum_{l=0}^{M-1}\left(z^{-1}W^{-k}\right)^l E_l(z^M) = \sum_{l=0}^{M-1}\left(z^{-l}E_l(z^M)\right) W^{-kl}. \qquad (30)$$

Equation (30) is realized by using the structure shown in Fig. 17. It should be noted that the DFT here is performed on each block, where each block is a single input vector as in (8), corresponding to each time index of the subband signals. By taking advantage of radix-2 choice for *M,* it is possible to use the fast Fourier transform (FFT) to perform the matrix multiplication shown in Fig. 17. Another important advantage, evident from (30), is that only **one** prototype filter is designed and the subband filters are simply the modulated versions of the same prototype.

Blocked Digital Filter

Another structure to realize a digital filter is the *Blocked Digital Filter.* Consider the scheme of Fig. 18, where $\mathbf{H}(z)$ is an $\mathrm{M}\times\mathrm{M}$ transfer function matrix [8], and $\mathbf{Y}_B(z)=\mathbf{H}(z)\mathbf{X}_B(z)$. It can be shown that this system is a linear time invariant system, and can be described by a scalar transfer function $H(z)$ iff $\mathbf{H}(z)$ is a pseudo circulant matrix. That is

$$\mathbf{H}(z)=\begin{bmatrix} E_0(z) & E_1(z) & \cdots & E_{M-1}(z) \\ z^{-1}E_1(z) & E_0(z) & \cdots & E_{M-2}(z) \\ \vdots & \vdots & \ddots & \\ z^{-1}E_{M-1}(z) & z^{-1}E_{M-2}(z) & \cdots & E_0(z) \end{bmatrix} \qquad (31)$$

The pseudo circulant property means that $\mathbf{H}(z)$ is constructed such that every row of the matrix is obtained from a circular shift of the previous row, and the elements below the main diagonal are multiplied by z^{-1}. Notice that Fig. 18 also represents a general maximally decimated filter bank (compare with Fig. 15). In filter bank language, we can therefore say that the system is alias free if and only if $\mathbf{R}(z)\mathbf{E}(z)$ is pseudo-circulant. In this case the input-output relation is $Y(z)=H(z)X(z)$ where

$$H(z)=z^{-(M-1)}\left(\sum_{i=0}^{M-1} z^{-i}E_i(z^M)\right) \qquad (32)$$

In block filtering, we can refer to the block filter of Fig. 18 as an implementation of the scalar filter $H(z)$.

The block digital filtering formalism provides a possible realization of a filter for moving the filtering operations to a lower rate. This structure can also be used when there is no rate conversion preceding or following the filtering operation. However, it is difficult to fabricate hardware for a set of digital filters to perform block digital filtering in the matrix form. Block digital filtering is an expensive operator due to matrix filtering operations, particularly when the number of subbands M is large. The complexity of block digital filtering applied to our problem is assessed in Section VII.1.

Subband Convolution Theorem

The analog of the convolution theorem for using filter banks is outlined in [9] and the idea is illustrated in Fig. 19. The *subband convolution theorem* for filter banks permits the convolution of two signals using the subband signals. In using this approach for computing the convolution, the decimated version of the convolution results is obtained as shown in Fig. 19. Given an analysis/synthesis filter bank $\{H_k, F_k\}$ for $0 \le k \le M-1$ that forms a perfect reconstruction (e.g., bi-orthonormal) system, the theorem states that using the set of filter banks $\{H_k, F_k\}$, the M-fold decimated version of convolution $x(n)*y(n)$ can be computed by computing the convolutions $x_k(n) * y_k(n)$ for all $k = 0,\cdots, M-1$, and then summing the results. That is

$$(x(n) * y(n))_{\downarrow M} = \sum_{k=0}^{M-1} (x_k(n) * y_k(n))_{\downarrow M} \qquad (33)$$

The interested reader may refer to [9] for detailed discussion of properties of the subband convolution theorem. This concludes our preliminaries and now we can begin to consider the problem of parallelization of our receiver.

III. Parallel Architecture for Demodulator

Digital filter bank theory offers a number of different methods for realizing a digital filter with a transfer function $H(z)$, at a lower processing rate than the input sampling rate. Here, we begin by describing the underlying problem, and then a class of digital filter banks are derived that are well suited for our application.

The demodulation and filtering operations are performed as shown in Fig. 20. The heterodyning receiver uses a mixer, shown here as a multiplier, to translate the signal to baseband, and the filter $H(z)$ is used to filter the double frequency images produced by the mixing operation. In this figure, $\hat{f}_c$ is the estimated carrier frequency. In the PRX, the mixing operation is performed in the subbands, as illustrated in Fig. 11.

It is possible to decimate the output of the filter, since the bandwidth of the filter is always less than the total bandwidth of the input signal $x(n)$. When bandpass sampling is used and the minimum rate ($f_s = 4W$) is used to sample the input signal, it is shown in [4] that J=2 can be used, in conjunction with half band filter for implementing H(z). Here we present the parallel implementation of the demodulator of Fig. 20. This implementation must satisfy the properties listed in Table III.1.

Filter Design Requirements
(1). Phase linearity of the filter is essential for tracking the phase, and extracting information such as the Doppler effects.
(2). Minimal distortion of the signal while maximally rejecting the double frequency images; with no additional loss in the new structure when compared to the traditional filtering approach.
(3). Facilitate mixing and filtering operation at the lowest possible rate; i.e., all arithmetic operations (additions and multiplications) are performed at the lowest possible rate.
(4). Another desirable (but not necessary) property of the parallel demodulator is to provide a discrete time sequence corresponding to each subband. The signal bandwidth is divided to evenly spaced subbands. This requirement translates into a set of analysis filter banks which are essentially a set of bandpass filters tuned to equally spaced center frequencies and which have equal bandwidth.

Table III.1. Filter Design Requirements

The fourth requirement broadens the scope of application of the demodulator particularly for multi-carrier modulation systems, or existing deep space mission modulation format in which a subcarrier is present. By having access to the subband signals, with linear phase property, a carrier signal can be directly accessed from the subband, and fed to a DPLL for tracking purposes. Returning to the parallelization of the demodulator, in Table III.2 the various approaches, their merits, and shortcomings are summarized.

Possible Approaches
(1). **Blocked digital filtering** method described in Section II.b, can be applied for parallelizing the computation, which results in a matrix filtering operation using the pseudo-circulant matrix in equation (31). This method entails a matrix filtering operation. However, the fourth design requirement in Table III.1 cannot be fulfilled here at all.
(2). **Digital filter bank** approach has been classically employed in applications of subband coding [8]. In these applications of filter banks, an allpass function is the desired frequency response of the system. In our case of interest, a lowpass function is the desired frequency response of the system. Here the fourth property is readily fulfilled as an added feature at no extra cost to the overall system.
(3). Use the **sub-band convolution theorem** which is another digital filter bank approach. The application of this filter bank is computationally even more complex than the second approach.

Table III.2. Possible Approaches to Parallel Realization of Demodulator

Based on the arguments listed in Table III.2, we are led to consider the digital filter bank solution based on the second approach. Typically, there are three sources of distortion in filter banks, these are: aliasing, amplitude distortion, and phase distortion. In a maximally decimated filter bank, the input signal is split into M subband signals $x_k(n)$ by M analysis filters $H_k(z)$ as shown in Fig. 14. In the case of maximally decimated filter banks, it can be shown that there exists a class of perfect reconstruction filters, referred to as M-channel *Quadrature Mirror Filters* (QMF) [8] which eliminate all three distortions for full band reconstruction. However, all our four design criteria cannot be simultaneously met by the QMF filter bank and maintain perfect reconstruction.

We refer to our filtering problem here as a "*partial band reconstruction*" as opposed to perfect reconstruction in which case the objective is to reconstruct the signal around the whole unit circle. This may be accomplished by considering the filter banks in Fig. 14, covering the full band $[0,2\pi)$, and simply keeping a subset of the synthesis filters and discarding the rest as shown in Fig. 21. The remaining subset constitutes the passband of the overall system, and the discarded set constitutes the stopband.

Let $\Omega = [0, 2\pi)$ denote the frequency domain, and $\Xi = \{H_0(z), H_1(z), \cdots, H_{M-1}(z)\}$ denote the set of M real-valued analysis filters, and let the non-overlapping frequency interval $I_k = \left[\frac{\pi}{M}k, \frac{\pi}{M}(k+1)\right)$ denote the frequency support of the k-th filter with the center frequency $f_k^c = \frac{\pi}{M}(k+1) - \frac{\pi}{2M}$, for $k > 0$ where $\Omega = \bigcup_{k=0}^{M-1} I_k$. By discarding the output of the analysis filters $k = j, \cdots, M-1$, this operation is equivalent to a lowpass filter, i.e., weighting the frequency response over the interval $\bigcup_{k=j}^{M-1} I_k$ with zero. The filter bank shown earlier in Fig. 14 is re-drawn in Fig. 22 to demonstrate the idea of dropping a subset of inputs to the synthesis bank for realizing a low pass frequency response.

In applying maximally decimated filtering, aliasing effects must be considered. In a perfect reconstruction filter bank, this aliasing is effectively canceled. Aliasing error due to dropping a subband is illustrated in Fig. 23. In Fig. 23 a subset of synthesis filter banks $F_i(z)$ for $i > k+2$ and $i < k$ are discarded. The aliasing error in the signal is not canceled in the frequency bands where the adjacent synthesis filters are discarded. This effect is exhibited in Fig. 23 in the frequency intervals I_k and I_{k+2}.

In order to deal with the aliasing we can use an oversampled filter bank, i.e., use more filters without increasing the decimation rate. This choice translates into decreasing the bandwidth of the analysis filters compared to the maximally decimated case. This class of filter banks is referred to as *non-maximally decimated filter banks* [8]. We begin by assigning a frequency support (passband) to each filter $I_k = \left[\frac{\pi}{M}k - \frac{\pi}{2M}, \frac{\pi}{M}k + \frac{\pi}{2M}\right)$ for complex subbands with $\Omega = [0, 2\pi)$, with center frequency $f_k^c = \frac{\pi}{M}k$, where $k = 0, \cdots, 2M-1$. This frequency allocation (or filter stacking) doubles the number of analysis and synthesis filters. However, since the decimation rate is kept at M, the separation between the center frequency of the images of each analysis filter relative to its passband is doubled. The idea of using the non-maximally decimated filter bank is demonstrated in Fig. 24 for the case when M=3. An example for obtaining a lowpass filter by dropping subbands in a non-maximally decimated filter bank for M=3, and $2M$=6 is shown in Fig. 24. The DFT filter bank provides an efficient realization for the non-maximally decimated filter banks. In the DFT filter bank the subbands are complex valued signals.

In a non-maximally decimated filter bank aliasing is **not** canceled but it is **suppressed** for all practical purposes. For applications of detection of signal in noise, a wise choice for the amount of this suppression is to assure that the aliasing level is far below the thermal noise level. The input and output of the filter bank (with a single channel shown in Fig. 25b) are related by

$$Y(z) = \frac{1}{M}\sum_{l=0}^{M-1} X(zW^l)\sum_{k=0}^{2M-1} H_k(zW^l)F_k(z). \qquad (34)$$

Note that the decimation ratio in this case is M and is half the number of subband channels $2M$. The distortion function for a full band reconstruction of a *non-maximally* decimated filter bank, assuming that the aliasing can be neglected, is

$$T(z) = \frac{1}{M}\sum_{k=0}^{2M-1} H_k(z)F_k(z). \qquad (35)$$

A by-product of this approach is the wider spaces between the images and the main signal, thereby providing ample room to eliminate the images. This enables the application of synthesis filters that may have a wider transition bandwidth than their analysis counterparts. The synthesis filter $F_k(z)$ may be designed to have a wide transition bandwidth and thus can be implemented with lower complexity. If $\{H_k(z)\}$ is a set of ideal brickwall filters, then the passband of $F_k(e^{j\omega})$ is the interval $I_k = \left[\frac{\pi}{M}k - \frac{\pi}{2M}, \frac{\pi}{M}k + \frac{\pi}{2M}\right)$ and the transition band is accordingly

$$I_k^t = \left[\frac{\pi}{M}k + \frac{\pi}{2M}, \frac{\pi}{M}(k+1) - \frac{\pi}{2M}\right) \cup \left[\frac{\pi}{M}k - \frac{\pi}{2M}, \frac{\pi}{M}(k-1) + \frac{\pi}{2M}\right). \qquad (36)$$

The frequency support and the inter-relationship of the transition and passband support of a non-maximally decimated analysis/synthesis filter bank is depicted in Fig. 25. In Fig. 25a, the frequency domain support of each analysis filter is depicted. An end-to-end single channel of the non-maximally decimated filter bank is shown in Fig. 25b, the transition bandwidth of the synthesis filter is shown in Fig. 25c, with the frequency support expressed in equation (36).

We choose a $2M$ non-maximally decimated DFT filter bank and recall that all the subbands are obtained by a set of uniformly shifted versions of a single prototype filter, i.e. $H_k(z) = H(zW_{2M}^k)$ and $F_k(z) = F(zW_{2M}^k)$. Ideally, $H_k(z)$ and $F_k(z)$ are designed such that the distortion function in (35) reduces to a constant value. This can be achieved if the sum of the frequency responses of the analysis filters $H_k(z)$ is a constant and the synthesis filter $F_k(z)$ is designed with linear phase and a wide enough passband to pass the subband signal undistorted. The condition imposed on the distortion function in (35) is reduced to

$$\sum_{k=0}^{2M-1} H_k(z)F_k(z) \cong \sum_{k=0}^{2M-1} H_k(z) = 2Mc. \quad (37)$$

It can be shown that this condition is satisfied by a special class of filters referred to as *Nyquist filters*. These filters are obtained by choosing an impulse response for the prototype filter such that

$$h(n2M) = \begin{cases} c & n = 0 \\ 0 & \text{Otherwise} \end{cases} \quad (38)$$

The Nyquist filters as defined in (38) are also referred to as *2M-th band* filters. The condition in equation (37) requires the impulse response of the filter to have periodic zero crossings separated by $2M$-samples.

In summary, our design guidelines for implementation of the filtering operation are listed in Table III.3.

Design Guidelines
(1). $H_0(z)$ must be linear phase and Nyquist (2M).
(2). $F_0(z)$ must be linear phase and may have a large transition bandwidth as specified in equation (36). The passband of $F_0(z)$ must contain the passband of $H_0(z)$ plus its transition band.
(3). Both the analysis and synthesis filter banks are implemented using a DFT filter bank, i.e. $H_k(z) = H_0(zW_{2M}^k)$, and $F_k(z) = F_0(zW_{2M}^k)$.

Table III.3 Design Guidelines for Design of Filter Bank

A more detailed discussion of the filter design for both analysis and synthesis bank using the above design guidelines can be found in Section IX. We can now draw the overall block diagram of the demodulator as shown in Fig. 26.b. In Fig. 26a, the original model is shown as a reference to compare these two structures.

An example of a filter bank for the case when M=16 is considered as satisfying the design guidelines is outlined in this section. The frequency response of $H_k(z)$ and $F_k(z)$ is illustrated in Fig. 27 for the three filters ($k = 0,1$ and 31). In these figures, the horizontal axis is $\omega/(2\pi)$.

IV. Digital Matched Filtering

The ideal digital matched filter for detection of signals in AWGN is the classical correlator shown in Fig. 7. In the digital implementation, an integrate-and-dump filter (IDF) is used to approximate the correlator output, as shown in Fig 28. The number of samples per symbol for full response signaling (i.e. when $p(t-kT)=0$ for $t \notin [kT,(k+1)T)$) is $N=T/T_S$.

The digital IDF detects the k-th symbol by summing all the N samples taken from $t=kT+\tau_0$ to $t=(k+1)T+\tau_0$. When sampling the received signal at a constant sampling rate, the i-th sample occurs at time iT_S. For the k-th symbol the "*offset in sampling*" is defined by the length of time after the start of the symbol to when the first sample in the symbol occurs. This time is $\delta=iT_S-(kT+\tau_0)$, where i is the smallest integer such that $iT_S-(kT+\tau_0)$, is non-negative, and $\tau_0 \le T_s$ is the timing offset. The first sample of each symbol may occur anywhere between 0 and T_s seconds after the beginning of the symbol. A typical symbol waveform and the sampling points are shown in Fig. 29 for the case when a rectangular pulse shape is the transmitted waveform.

There are two ways to deal with the offset. The first approach is to synchronize the sampling clock with the symbol clock. This is not desirable in space communication applications since the sampling clock is synchronized with an ultrastable clock source (such as a MASER) and is used to time tag the carrier phase estimate for ranging applications. The second and more versatile approach is to virtually use a finer granularity in the time domain than the sampling period. The finer resolution in time is only used for pre-computing the matched filter coefficients during the design phase. Conceptually, we begin by expanding the input signal by L and obtain a matched filter with a higher resolution and then decimate by L. The derivation leads to a matched filter which is time varying. The weight sequence is matched to the transmitted pulse shape $p(t)$, i.e. $w_i = p((iT_s/L)-kT+\delta)$. Note that the discrete time index i of the weight sequence w_i varies at the rate $1/L\ T_S$. The output rate of the matched filter is at the symbol rate, hence, the expanded rate is decimated to the input rate by L and then to the symbol rate by D (note that here $D=N$). The integer delay $d=L\delta$ translates into a fraction of the sampling period from the beginning of a symbol period. This delay is estimated by measuring the offset of the expanded sampling clock with respect to the phase of the NCO in the symbol synchronization loop. This delay varies much more slowly than the symbol rate. For every value of *d*, we can formulate the matched filter as a linear time invariant (LTI) system denoted by $Q_d(z)$, as illustrated in Fig. 30. The application of the polyphase identity of Fig. 13.c (since the decimation and expansion rates are equal in Fig. 30) enables us to model the matched filter as an LTI system with the transfer function denoted as $Q_d(z)$ and the inverse z-transform $q_d(n)$. In the following section, the parallelized version of the matched filtering operation in Fig. 30 is considered.

IV.1 Combined Demodulator and Matched Filter

An efficient implementation of the matched filter is obtained by combining the matched filtering operation with the demodulator filter bank. The key advantage in combining the demodulation filter bank with the matched filter is to use the same subband signals produced in the demodulation stage, which are already parallelized and sampled at the lowest rate. For simplicity in illustrating the combined structure, we ignore the mixing operation for converting the input signal $x(n)$ to baseband, which has already been outlined in Section III. The methodology here is first to design the subband matched filters assuming an allpass characteristic of the demodulation filter bank, and then later we revert to the previous approach of discarding the subbands for synthesizing a lowpass characteristic.

In what follows, the derivation of combined demodulator and matched filter is presented step-by-step. We begin by considering the allpass filter bank followed by the matched filter in Fig. 31a. The analysis, matched, and synthesis filters are shown in Fig. 31a, with transfer function $H_k(z)$, $Q_d(z)$ and $F_k(z)$ respectively (recall that $r(n)$ was originally defined in Fig. 7). The matched filtering operation can be commuted with the synthesis banks as shown in Fig. 31b.

Let $q_d(n)$ denote the impulse response of the matched filter with the z-transform $Q_d(z)$. It can be shown that the block diagram shown in Fig. 32 is equivalent to the one shown in Fig 31.b. Note that any other filter with appropriate design criteria may be used for computing this convolution. Hence, the synthesis filters have been replaced from $F_k(z)$ to $F_k'(z)$, even though the designer may select to use the same filters. The Fourier transforms at points ① and ② in Fig. 32 are shown in Fig. 33. In Fig. 33, the output of $F_k'(z)$ is denoted by $P_k(z)$ where $P_k(z) = Q_d(z)F_k'(z)$.

The next step in the derivation is the inclusion of a decimator-interpolator pair prior to the convolution and the addition of the synthesis filter $F_k(z)$ for demodulation after the convolution, as shown in Fig. 34. In Fig. 33, the frequency characteristics of the signals at various points of Fig. 32 and Fig. 34 are illustrated. When $F_k'(z)$ and $H_k(z)$ have sufficient stopband attenuation and appropriate passband width and ripple, then for all practical purposes the systems in Fig. 34 and Fig. 32 produce the same output. The Fourier transform at point ③ is the product of the Fourier transforms of the signals at points ① and ②. It is clear that at point ③ the frequency support of the signal is similar to that of the analysis filter. In the same way, the Fourier transform at point 5 is the product of the Fourier transforms at points ① and ④. Then, the k-th image in point ⑤ is equal to the signal in point ③. The filter $F_k(z)$ will only pass the k-th subband signal and reject the remaining images, making the signal at points ③ and ⑥ approximately equal. Inaccuracies may only result from non-ideal frequency responses of the filters.

The idea now is to apply the convolution identity shown in Fig. 35 for moving the expansion operation to the last stage. The application of this identity results in performing all arithmetic operations at the lowest possible rate, prior to rate expansion. In Fig. 35, the convolution operation of the two signals $x_1(n)$ and $x_2(n)$ performed at the high rate is reduced to convolution at the lower rate by a factor of M. Applying this multirate identity to the scheme of Fig. 34, we obtain the subband version of the matched filter shown in Fig. 36.

We use the efficient implementation of a DFT filter bank as shown in Fig. 17. In a non-maximally decimated filter bank, the $2M$ filter bank output is decimated by M. This in turn requires z to be replaced by z^2 in the polyphase filters. The resulting structure is shown in Fig. 37a. In this figure $R_i(z) = E''_{2M-1-i}(z)$, where $E_i''(z)$ is the i-th polyphase component of the synthesis filter $F_0(z)$, $E_i(z)$ denotes the i-th polyphase component of the analysis filter and $E_i'(z)$ denotes the i-th polyphase component of the filter bank used for performing the subband convolutions.

The computations leading to the set of signals $s_k(n)$ can be performed off-line, if desired. That is, $Q_d(z)F_k(z)$ is computed and the result is decimated and stored in the time domain. In Fig. 37.b, the convolution in the subband signals is replaced with a subband matched filter denoted here as $G_k(z)$ for $k = 0,\cdots,2M-1$, where $G_k(z)$ is a filter with impulse response $s_k(n)$.

The synthesis part can be further simplified by noting that the output of the DFT (matrix multiplication by *W*) is composed of 2*M* points and the expansion rate is *M*. Hence, we can now re-organize the system of Fig. 37 as shown in Fig. 38, where the addition of the output sequence of the synthesis bank is performed at the lower rate. For effective implementation of the interleaver structure in Fig. 38, the interleaver can also be reduced to a multiplexor, whereby each symbol output $\hat{a}_k$ is associated with a specific subband. This is the subject of the next section.

IV.1.a Filter Bank Output Multiplexing to Symbols

Let *g* denote the greatest common divisor of *M* and *D* (shown in Fig. 39a), i.e. $g = \gcd(M,D)$, then there exist $\tilde{M}$ and $\tilde{D}$ such that $M = g\tilde{M}$, $D = g\tilde{D}$, where $\tilde{M}$ and $\tilde{D}$ are relatively prime. So there exist integers n_o and n_1 such that $n_o\tilde{M} + n_1\tilde{D} = 1$. It can be shown then that the expander followed by the delay chain and decimator shown in Fig. 39a is equivalent to Fig. 39b.

Let *i* denote the channel number corresponding to the i-th subband, and define l_i and r_i such that $z^{-(l_i\tilde{M}+r_i)} = z^{-in_1}$; i.e., $r_i = i\,n_1$ Modulo $\tilde{M}$ is the remainder part and l_i is the integer part of $\frac{in_1}{\tilde{M}}$. Since gcd($n_1,\tilde{M}$)=1 then $r_i \neq r_j$ for all $i \neq j$, i.e., each r_i is unique. Using the multirate identity depicted in Fig. 39, the overall structure may be drawn as shown in Fig. 40. It is noted that for a fixed symbol rate the delay length for each subband is fixed, hence reducing the interleaver to a multiplexing circuit, and a routing switch.

IV.1.b Efficient DFT Computation of Synthesis Filters

Further reduction in the FFT computation of the matrix multiplication with **W** can be attained by noting that only a subset ($2\tilde{M}$-branches) of the output branches are needed for obtaining the output symbol sequence. Let

$$X(k) = \sum_{n=0}^{2M-1} x(n) W_{2M}^{kn},$$

where $k = g-1, 2g-1, \cdots, 2M-1$, and $M = g\tilde{M}$, i.e., $k = mg-1$ when $m = 1, 2, \cdots, 2\tilde{M}$. Then, it is easy to verify that the FFT may be decomposed into two parts, that is

$$X(mg-1) = \sum_{n_1=0}^{2\tilde{M}-1} W_{2\tilde{M}}^{mn_1} \left\{ \sum_{n_0=0}^{g-1} W_{2M}^{-(2\tilde{M}n_0+n_1)} \; x(2\tilde{M}n_0 + n_1) \right\}.$$

In which case, a g-point FFT can be used to compute the inner DFT for each n_1, and then a $2\tilde{M}$ FFT is performed to compute the final product. This concludes our discussion of combined demodulation and digital matched filtering.

IV.2 IFIR Approximation for Digital Matched filtering

The results of this section can be used independently of the results in other sections of this report. The following method may be incorporated in the filter bank structure when the matched filter has a high order and requires a large number of taps. This method effectively reduces the number of taps at the input rate by realizing an equivalent filter as described here in this section.

We begin by considering the filtering operation shown in Fig. 41. Our approach here is based on using the *interpolated finite impulse response* (IFIR) filtering. This means that we use the following decomposition to realize an FIR approximation to the matched filter in the frequency domain,

$$H^{MA}(z) = G(z^{L/2})\, I(z), \qquad (39)$$

where L is an integer referred to as the stretch factor. This decomposition results in an efficient implementation of a narrow band lowpass filter. Let N represent the order of filter required to meet the specification to implement $H^{MA}(z)$. The stretched filter $G(z^{L/2})$ has transition bandwidth $L\Delta f/2$, so the order of the filter is reduced by a factor of $N/2L$, where Δf is the transition width of the original filter. This translates into substantial savings in multiply accumulate operations by a factor of $L/2$. The unwanted shifted version of $G(z^{L/2})$ is then suppressed with a filter denoted here as $I(z)$. This filter has a very wide transition bandwidth [8] and it requires a low order. The overall filter is shown in Fig. 42.

It is possible to decompose the stretched filter $G(z)$ into even and odd polyphase components and re-draw the system shown in Fig. 42a as depicted in Fig. 42b, and after some manipulation we can arrive at the system shown in Fig. 43, where the even and odd polyphase components of $G(z)$ are denoted respectively as $G_0(z)$ and $G_1(z)$, and the $L/2$ polyphase components of $I(z)$ are denoted as $I_0(z),\cdots,I_{L/2-1}(z)$.

The IFIR digital matched filter described here is designed by considering the impulse response of the digital matched filter denoted here as $h(n)$. The problem can be formulated as a least squares optimization problem as follows: given an arbitrary impulse response $h(n)$ of order N, find $\hat{h}(n)$ such that

$$\sum_n |\hat{h}(n)-h(n)|^2, \qquad (40)$$

is minimized, subject to the constraint

$$\hat{H}(z)=G(z^{L/2})I(z), \qquad (41)$$

where the impulse response of $G(z)$ is $g(n)$ of order N_g and the impulse response of $I(z)$ is $i(n)$ of order N_f, and $N = N_f + (L/2)N_g$. We begin by re-writing the constraint (41) in the time domain and use the fact that multiplication in the frequency domain is equivalent to convolution in the time domain. Let

$$\hat{\mathbf{h}} = \begin{bmatrix} \hat{h}(0) \\ \vdots \\ \hat{h}(N-1) \end{bmatrix} \qquad \mathbf{g} = \begin{bmatrix} g(0) \\ \vdots \\ g(N_s-1) \end{bmatrix}, \qquad (42)$$

and define matrices

$$\mathbf{K} = \begin{bmatrix} i(0) & 0 & 0 & \cdots & 0 \\ i(1) & i(0) & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ i(N_f-1) & i(N_f-2) & \cdots & & i(0) \\ 0 & i(N_f-1) & \cdots & i(1) & i(1) \\ & \vdots & \ddots & \vdots & \\ \vdots & & & i(N_f-1) & \\ 0 & 0 & \cdots & & i(N_f-1) \end{bmatrix}, \quad (43)$$

and

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & \\ 0 & \cdots & \ddots & & \\ \vdots & & & \ddots & \\ 0 & & & & 1 \end{bmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} L/2-1 \text{ rows of zeros} \\ \\ \\ \\ \\ \end{array}, \quad (44)$$

Then, we can write

$$\hat{\mathbf{h}} = \mathbf{KSg}. \qquad (45)$$

Thus, minimizing equation (40) is equivalent to minimizing

$$\left\|\mathbf{h} - \hat{\mathbf{h}}\right\| = \left\|\mathbf{h} - \mathbf{KSg}\right\|^2 \qquad (46)$$

When the matrix columns of the $\mathbf{KS}$ are linearly independent and $(\mathbf{KS})^{\mathrm{T}}(\mathbf{KS})$ is nonsingular, the least squares solution is given by the relation:

$$\mathbf{g} = \left((\mathbf{KS})^{\mathrm{T}}(\mathbf{KS})\right)^{-1}(\mathbf{KS})^{\mathrm{T}}\,\mathbf{h} \qquad (47)$$

The matrix $\left((\mathbf{KS})^T(\mathbf{KS})\right)^{-1}(\mathbf{KS})^T$ is referred to as the pseudo-inverse or the Moore-Penrose generalized inverse of matrix $\mathbf{KS}$. Due to the construction of matrix $\mathbf{KS}$ in our problem, the existence of a solution is always guaranteed.

Similarly, from equation (47), we can find the best **g** such that the objective function $\sum_n |\hat{h}(n) - h(n)|^2$ is minimized. Let

$$\mathbf{D} = \begin{bmatrix} g(0) & 0 & 0 & \cdots & 0 \\ g(1) & g(0) & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ g(N_g-1) & g(N_g-2) & \cdots & & g(0) \\ 0 & g(N_g-1) & \cdots & g(1) & g(1) \\ & \vdots & \ddots & \vdots & \\ \vdots & & & g(N_g-1) & \\ 0 & 0 & \cdots & & g(N_g-1) \end{bmatrix} \text{and } \mathbf{f} = \begin{bmatrix} f(0) \\ \vdots \\ f(N_f) \end{bmatrix}, \qquad (48)$$

then the objective function in (46) is minimized and

$$\mathbf{f} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{h} \;{}^* \qquad (49)$$

The optimization procedure is summarized below:

Design Algorithm for IFIR Matched Filter:
(1). Pick initial guess for $f(n)$, e.g. $f(n) = 1$ for all $0 \le n \le N_f$. (2). Use equation (47) to optimize g. (3). Use equation (49) to optimize f. (4). If $\sum_n |\hat{h}(n) - h(n)|^2$ is not acceptable go to 2.

Suppose $h(n)$ has stopband edge of about π/α. Then L is chosen between $\alpha/2$ and $3\alpha/2$ and the optimization solution is good in general.

*Formulation of this iteration algorithm is due to Yuan-Pei Lin, graduate student, California Institute of Technology.

In Fig. 44, an example is provided for a rectangular pulse. The filter order is chosen to be N=90, L=4, N_g=24 and the original impulse response $h(n)$ is simply a square pulse with the first twenty and the last twenty samples set to zero. The approximation of this impulse response is difficult and rather interesting, due to Gibbs's phenomena. The approximation error, in equation (46) in this example after 10 iterations is only 1.4×10^{-6}, which is a relatively small value.

We conclude from this example that for even a small number of iterations, this algorithm yields a small error for IFIR approximation of the original filter.

V. Symbol Timing Recovery

Symbol timing recovery is accomplished here by using a digital transition tracking loop (DTTL). The DTTL utilizes a matched filter over the duration of the transition epoch of a symbol, giving rise to the name *mid-phase* matched filter. The block diagram for the DTTL is shown in Fig. 45. The product of the output of the mid-phase matched filter with the output of a transition detector (as will be defined shortly) provides the timing phase error. This error is further averaged, filtered and used by the NCO to generate a square wave which is used as the reference recovered symbol clock.

The timing phase error is $e_k = b_k \tau_k$, where τ_k is the output of the midphase matched filter, that is $\tau_k = \frac{1}{N}\sum_{n\in\Psi} w_n x_n$, here the set $\Psi = \{n : kT + \Delta \le nT_S < kT - \Delta\}$, with the cardinality $N = |\Psi|$, 2Δ is the window size, and the transition detector output is

$$b_k = \frac{Sign(\hat{a}_{k-1}) - Sign(\hat{a}_k)}{2} = \begin{cases} 0 & \text{if } \hat{a}_k = \hat{a}_{k-1} \\ +1 & \text{if } \hat{a}_k = -1 \ \hat{a}_{k-1} = +1 \\ -1 & \text{if } \hat{a}_k = +1 \ \hat{a}_{k-1} = -1 \end{cases}$$

The output of the transition detector determines the sign of the phase error. The timing phase error e_k is averaged over many symbols and further filtered as shown in Fig. 45. The steady state and transient behavior of this loop can be found in [11].

The ideal output of the integrate-and-dump filter (IDF), and the mid-phase IDF are illustrated in Fig. 46 for an-all-one symbol sequence. The saw-tooth behavior in the ideal case, is due to 'dumping' the content of the IDF at the end of each integration period. The digital matched filtering operation can be implemented as a convolution like an FIR filter. When no windowing is used, the output of the mid-phase matched filter (length T seconds) can be obtained from the continuously running matched filter, by simply sampling the matched filter output each $T/2$ seconds, as shown in Fig. 46.

It has been shown that the loop SNR [11] of the DTTL can be improved, by applying a matched filter over a narrower time epoch than the full symbol duration. This is achieved by windowing the input sequence around the transition epoch of the samples within a symbol period. In general, the duration of the window Δ is defined as an even sub-multiple (e.g. 1/2, 1/4, 1/8, etc.) of the symbol period. Thus, the window sequence becomes

$$\Pi(n,k) = \begin{cases} 1 & kT - \frac{T}{2^i} \le nT_s < kT + \frac{T}{2^i} \\ 0 & \text{Otherwise} \end{cases} \qquad (50)$$

The mid-phase matched filter, in the windowed case, is implemented separately from the matched filter with an impulse response given by

$$h^{MP}(n) = \Pi(n,k)\, h^{MA}(n) \qquad (51)$$

With using the truncated impulse response of the new matched filter $h^{MP}(n)$ derived from $h^{MA}(n)$ according to equation (51), the results of parallelization of the matched filter in Section IV become directly applicable here by replacing w_n with $h^{MP}(n)$ in equation (51).

In view of the above facts, we can summarize here our results and state that the mid-phase matched filter can be obtained by simply sampling the matched filter output at the half symbol time period when no windowing is applied, and for the windowed case, the problem reduces to that parallelization of a separate matched filter which was addressed in Section IV. The DTTL can be incorporated into the subbands as shown in Fig. 47. In this figure, M' is used to denote the decimation rate of each parallel path. The parameter M' is essentially the ratio of the symbol rate to the desired processing rate of the parallel DTTL. The designer may select $M' = D$, if desired.

VI. Carrier Phase Estimation and Costas Loop

We begin by considering a Costas loop for phase estimation and tracking. The basic structure of a Costas loop is shown in Fig. 48. The performance of the Costas loop is available in many textbooks such as [5]. The squaring loss, tracking error variance, and the S-curve associated with using this loop are discussed in [10].

Let $\Delta\phi = (\theta - \hat{\theta})$ represent the phase error between the actual and the estimated carrier phase. The in-phase and quadrature components of the output of the matched filter ($\mu(\hat{a}_k)$), respectively can be written as:

$$\begin{aligned} y_c(n) &= (\hat{a}_k + n_k^c)\cos(\Delta\phi) + n_k^s \sin(\Delta\phi) \\ y_s(n) &= (\hat{a}_k + n_k^c)\sin(\Delta\phi) - n_k^s \cos(\Delta\phi) \end{aligned} \qquad (52)$$

The bandpass discrete time noise terms n_k^c, n_k^s in (52) are defined in [5]. The output of the phase detector is the product of the two sequences $y_c(n)$, and $y_s(n)$; which can be approximated as $\xi_k \approx \hat{a}_k{}^2 \sin 2(\theta - \hat{\theta})$. For a binary signal and small phase error $\xi_k \approx 2(\theta - \hat{\theta})$. The phase error is further integrated and filtered by an infinite impulse response (IIR) filter to track the phase perturbations in the carrier phase of the received signal. This phase estimate is then used by the NCO to generate the reference in-phase and quadrature components.

The loop update rate is the rate at which the output of the phase detector is fed into the loop filter. It has been shown in [10] that the digital implementation of the phase locked loop requires the product of the update period and the loop bandwidth to be larger than ten. As an example, for a loop bandwidth of 50 Hz, a minimum update rate of 500 Hz is necessary. Otherwise, the loop behavior will be different from its analog counterpart. The structure shown in Fig. 49 for parallelized Costas loop architecture corresponds to a loop update rate of $1/M''$ of the symbol rate. It is noted that the loop bandwidth in most applications is much lower than the data rate. In this figure, M'' is used to denote the decimation rate of each parallel path. The parameter M'' here is the ratio of the symbol rate to the desired processing rate of the parallel Costas loop. The designer may select $M'' = D$, if desired.

VII. Alternative Architectures

In this section, three different approaches to parallelization of a single filter followed with a decimator, as shown in Fig. 50, are re-examined. Let M denote the decimation ratio between the input sampling rate and the processing rate, let D denote the output decimation rate (which in matched filtering is equivalent to the number of samples per half symbol), and for simplicity we assume M is a multiple of D; and L denotes the filter length of $H(z)$. Here, we consider three options, namely:

(1). *Direct Parallel Architecture*: based on the blocking method for a digital filter as described in Section II, and illustrated in Fig. 18.
(2). *Frequency domain convolution*: using a DFT to perform the convolution of $H(z)$ with $X(z)$ as shown in Fig. 51. This approach has been classically used [8] to compute linear convolution and is referred to as the "overlap and save" method.
(3). *Filter Bank Approach*: based on the filter bank structure derived in Sections III and IV.

VII.1. Complexity and Computational Analysis

The computational complexity of the three options for parallelization of the digital filter followed by the decimator is assessed. The computational complexity of each option is stated in terms of the number of real multiplication operations needed at the low sampling rate.

In option 1, the matrix filtering entails a total of ML real multiplications when all the coefficients are non-zero. Here, only $1/D$ of the rows need to be implemented, leading to a reduction of the complexity by the same factor.

In option 2, the frequency domain convolution requires two FFT's of size $M+L$, and $M+L$ complex additional multiplications as shown in Fig. 51. The total complexity for this option is depicted in the second row of Table VII.1.

In option 3, the complexity of the filter bank approach is approximated by assuming that the order of the synthesis filter bank as well as the bank used for generating the subband filters is $8M$, and the order of the analysis filter bank is $12M$. These approximations have been derived empirically and represent a rough figure for the order of the filter bank. The total complexity of the polyphase components of the analysis filter $H_0(z)$ is equal to the complexity of implementing $H_0(z)$, i.e. $12M$ multiplications. The same applies to $F_0(z)$, but only 1/D of the polyphase components need to be implemented. The order of each subband's filter is about $(8M + L)/M$. There are M such filters with complex coefficients. There are also two FFT's of size $2M$. The result is summarized in Table VII.1.

Option	Operations
I. Block Digital Filtering	$\frac{M}{D} \cdot L$
II. Frequency Domain	$2(M+L) + 2(M+L)\log_2(M+L)$
III. Filter Bank	$4M\log_2(2M) + 28M + \frac{8}{D}M + 2L$

Table VII.1. Complexity of Each Option

The expressions for the number of operations of these three options are plotted in Fig. 52. In Fig. 52, each option is represented as a subspace in the two dimensional plane whose coordinates are L and M. Each subspace represents the range of variables L and M that yield minimal complexity among the three options. It is interesting to note that for small L, the block digital filtering results in the lowest number of operations.

VIII. Delay

The total delay of the receiver is composed of group delay, plus the processing delay. Our proposed structure of the PRX employs only FIR prototype filters with linear phase. Hence, the overall system group delay for carrier and timing synchronization is constant.

The processing delay in a digital demodulator plays an important role. If the delay is too large, it can lead to faulty behavior in the synchronization loops (carrier or timing). The processing delay of the PRX structure is composed of the delays caused by the analysis and synthesis filter bank, plus the delay in FFT and inverse FFT computation, plus the delay in the subband matched filters. Each FFT corresponds to a delay of $\log_2(2M)$; the delay for analysis and synthesis filters used in PRX is the delay of each prototype respectively. The delay of the matched filtering in the subband is the sum of the delays caused by the matched filter prototype and the delay of $F_0'(z)$. The total delay in the PRX is the sum of the individual delays for analysis, synthesis, FFT and FFT inverse, and matched filtering operation.

IX. Simulation of PRX

In this section, the performance of the PRX architecture is verified by simulation. The model simulated here is shown in Fig. 37b, with synthesis section as depicted in Fig. 38. In this section $r(n)$ is the complex output of the IF demodulator, as shown in Fig. 7. The real part of output of the filter bank matched filter after decimation by the integer D is used for detecting the symbol sequence. Both real and imaginary parts of the output of the matched filter are used for obtaining the phase error estimate for closing the Costas loop. The Costas loop implementation is similar to the classical model shown in Fig. 5.

IX. 1. Filter design.

In implementing the PRX architecture, three filters have to be designed, as shown in Fig. 37b. The filters are $H_0(z)$ and $F_0(z)$, which are the prototypes for all the filters $H_k(z)$, $F_k(z)$, and the matched filters $G_k(z)$, for k=0,...,2M-1. The filters $E_k(z)$ are the type-I polyphase components (refer to equation (13)) of $H_0(z)$, and $R_k(z)$ are the polyphase components of $F_0(z)$,with change of index such that $r_l(n) = h(nM - l)$.

The filter $H_0(z)$ must be designed under the constraint of being Nyquist (2M) (refer to equation (37)). By considering the support of each of the signals in Fig. 33, the filter $H_k(z)$ must reject the images of $P_k(z)$ which is derived from the convolution of $F_k(z)$ and the matched filter. Since we have no control on the matched filter, we will assume that $P_k(z) = F_k(z)$. This is equivalent to assuming that the matched filter bandwidth is wider than the synthesis filter bandwidth. Hence, the stopband of $H_k(z)$ must include the transition band of the adjacent images of $F_k(z)$. This requirement is equivalent to setting a limit of π/M for the sum of the transition regions of $H_0(z)$ and $F_0(z)$ as shown in Fig. 53. The effect of $F_0(z)$ on the complexity of the system is more significant, since it also impacts the length of the impulse response of the subband matched filters $G_k(z)$, which have the highest number of taps among all the filters used in the PRX. Effective reduction of the order of $F_0(z)$ can be achieved if the prototype analysis filter $H_0(z)$ is designed with a sharp transition band. This choice of $H_0(z)$ allows for $F_0(z)$ to have a wider transition and thus be lower in complexity. An additional property of the synthesis filter $F_0(z)$ is to possess a wide enough passband with low ripple that covers the transition region of $H_0(z)$. We have chosen the order of $H_0(z)$ to be $12M$ and the order of $F_0(z)$ to be $8M$. Then each polyphase component of $H_0(z)$ has length six and of $F_0(z)$ has length four. These orders are not a minimal choice. It is possible to reduce the length of $H_0(z)$ and $F_0(z)$ by better filter design optimization techniques.

The approach taken here for designing $H_0(z)$ is by windowing the impulse response of an ideal filter. The window chosen here is the Hamming window to provide smooth response and a non-equal ripple stop band. In our application, monotonically increasing stop band attenuation (or equivalently non-equal ripple) is a desired property, since it results in further rejection of distant images from the filter cut-off frequency. This property insures that only the neighboring filters contribute to the aliasing distortion. It must be noted that by choosing the bandwidth to be $\pi/2M$, the resulting filter is forced to be Nyquist-2M, independent of the window shape used for designing the filter bank.

The filter $F_0(z)$ must have a symmetric impulse response for linear phase property. It must also have a wide enough bandwidth to preserve the Nyquist property of $H_0(z)$. Recall that $F_0(z)$ should also satisfy, together with $H_0(z)$, the requirement illustrated in Fig. 53. In our design example, the stop-bands of both $H_0(z)$, and $F_0(z)$ provide better than 60-dB attenuation.

The construction of subband matched filter $G_k(z)$ is as follows. Let $q_d(n)$ be the desired matched filter impulse response. Let $s_k(n)$ be the impulse response of $G_k(z)$. The complex valued sequence $s_k(n)$ is obtained by passing the sequence $q_d(n)$ through the filter $F_k'(z)$ and decimating the output by M. This idea is shown in Fig. 36. Here, we have chosen $F_k'(z)=F_k(z)$ for simplicity. The computation of $s_k(n)$ begins by computing $f_k(n)$ which is the impulse response of $F_k(z)$. The impulse response $f_k(n)$ is calculated by multiplying $f_0(n)$ with W_{2M}^{-kn}. Then $q_d(n)$ is convolved with $f_k(n)$ and the result is decimated by M to obtain $s_k(n)$.

IX. 2. Generation of Input Signal.

The RF input to the receiver is filtered by an analog band-pass filter and then sampled by an A/D converter, forming the input to the digital portion of the receiver. Since the simulation software can only simulate discrete time systems, we have used in the simulation of the analog portion a much higher sampling rate than the one used in the actual receiver, e.g. 100 times. In order to have a different sampling rate in each portion, the simulation of the analog part is executed separately and its output is saved into a file. This file is then used as the input to the simulation of the digital portion. The implementation of the input generation system is illustrated in Fig. 54, where an example with 4 samples per symbol is shown. A Gaussian filter was chosen to model the analog filter. This choice of filter was made for introducing low distortion of the transmitted pulses. The system generates base-band samples, but in the receiver portion these samples are up-converted to IF frequency and then demodulated to baseband again to fully model the IF downconversion stage needed in real applications.

IX. 3. Description of PRX used in the simulation.

This input signal (from a file) is up-converted and the IF signal is formed by multiplying the input signal with a sinusoid of frequency $f_s/4$. This signal, which is the input to the receiver, is demodulated by an NCO (Numerically Controlled Oscillator) to construct the real and imaginary components (or I and Q) of the demodulated base-band signal. This signal is then vectorized to a length $2M$ -vector by a serial to parallel converter. Note that there is an overlap of M samples between every two consecutive vectors, where the M first components of the n–th vector are the M last components of the n–1'th vector. In our Signal Processing Workstation (SPW) simulation, the number of filter banks is $2M$=16, hence, the decimation rate is M=8. In a system with 100 MHz sampling rate, the processing rate would be 12.5MHz.

The vectors are processed at the low sampling rate which is f_s/M. Each of the components of the vector is filtered by the appropriate polyphase components of $H_0(z)$. Altogether, these filters are referred to here as the vector filter $\mathbf{H}$.

The resulting sequence of vectors output by **H** is processed by an FFT block, one at a time. The subbands (altogether consisting of a vector sequence) are masked such that only seven (7) bands are fed through the synthesis bank. This mechanism results in realizing a lowpass frequency characteristic from the filter bank, as described earlier in Section III. These seven bands cover the frequency region -0.44π to 0.44π and reject all the frequencies $|f|>0.5\pi$ (in our 100 MHz example, this bandwidth corresponds to a 22 MHz band). This symmetric arrangement of the subbands around the zero frequency (d.c.) manifests itself in a symmetric frequency response of the subband matched filters. The subbands are further processed by a vector filter **G** for subband matched filtering, as described in Section IV. The output of **G** is transformed back by an IFFT and processed by the synthesis vector filter **F**. The output vector of length $2M$ samples is combined to form a length M vector by delaying half of the components, and summing the latter half to the other half as depicted in Fig. 38. This forms the parallel output of the matched filter. The output of the combined demodulator and the matched filter is generated in parallel at the lower rate. A subset of these parallel outputs, specifically $1/D$ of these outputs are used for detecting the symbols, and for closing the Costas carrier tracking loop. Recall that when M is a multiple of D, the symbol sequence has a one-to-one correspondence with the subband signals (refer to Section IV.1.a).

In the following section, simulation results are presented for both the bit error rate (BER) and the mean square error (MSE) associated with the system described here.

IX. 4. Simulation results.

The set of experiments is summarized here as follows.

1. Partial band reconstruction.

This test is intended to verify the reconstruction of a desired band by selection of the appropriate subbands. In this test, the filters $G_k(z)$ are set to unity (no subband matched filtering). One or more subbands' responses are computed by masking out (multiplying by zero) all the other bands. The frequency response of the system is obtained by applying a delta function at the input and computing the FFT of the output sequence from the filter bank. The results are shown in Fig. 55. In Fig. 55a, the response of the seven bands is shown. In Fig. 55b,c two individual subbands are shown. The response in Fig. 55a is the summation of seven such individual responses. This experiment verifies the partial band reconstruction property of our filter bank used for demodulation.

2. Combined demodulation and matched filtering.

Here we demonstrate the impulse response of an integrate-and-dump filter, implemented by the filter-bank, with subband masking included. A low-pass response is formed by allowing only seven sub-bands to be fed through as described earlier in the first experiment. The integrate-and-dump response when incorporated into the filters $G_k(z)$ is depicted in Fig. 56. In this experiment, the proper operation of our structure for combined demodulation and matched filtering is confirmed.

3. BER degradation of baseband filter-bank implementation.

In this simulation our goal is to assess any losses associated with parallel realization of the matched filter. Here the Bit Error Rate (BER) is measured by simulation at the baseband. In the baseband implementation, there is no demodulation stage and the subbands are not masked, since there are no double frequency images to reject. In this simulation, we compared the BER performance of the ideal matched filter (integrate-and-dump) operating on a random BPSK signal in AWGN channel, and the filter-bank implementation of the same filter using the same signal. The analysis and synthesis filter lengths were chosen 12M and 8M respectively. The result indicates that the difference between the two systems is too small to be observed even in very long runs (e.g. 10^7 symbols at low SNR), thus the parallel realization results in negligible loss of performance.

4. Mean Square Error (MSE) measurement.

In order to further quantify the implementation error, in this simulation we use the Mean Square Error (MSE) criterion. The filter bank implementation error of the matched filter was measured by simulation. Having a matched filter (integrate & dump) operating on a random signal and the filter-bank implementation of the same filter operating in parallel on the same input signal, we measure the average power in each realization. This constitutes the MSE measurement here, when the output power of the matched filter is normalized to unity. The very small MSE affirms the negligible loss in the BER measurement. The MSE is tabulated in Table IX.4.1.

Order- $H_0(z)$	**Order-** $F_0(z)$	**MSE**
16M	12M	8.17*10^-6
12M	8M	1.27*10^-5

Table IX.4.1 MSE Measurements

5. BER measurement in IF simulation.

The IF simulation arrangement was described earlier. In this simulation, the BER result is compared to the ideal theory when only four (4) complex samples per symbol are used for detecting the symbols. The Bandwidth-bit-time (BT) product of the simulated analog filter is 1.5. The results are shown in Fig. 57. Note that the degradation shown in Fig. 57 is due only to the low number of samples per symbol [7].

X. Future Direction for Research and Applications

The effect of quantization and finite bit arithmetic on the overall system has to be further investigated. The need may arise that more specific design procedures have to be formulated for low sensitivity filter bank design.

A second area of interest is to investigate and study possible approaches for efficient hardware realization of the receiver. Here, the merits of coarse (e.g. board level) or fine computational processes (e.g. using systolic arrays, custom VLSI chips, ASICs or others) for this application are assessed.

A third area of interest is to tailor the architecture of PRX for multiple spacecraft applications. Missions involving multiple spacecraft within the same line of sight (or beam width of the antenna) could effectively employ a single receiver using our methodology. This is due to natural decomposition of the input signal into non-overlapping frequency bands in the PRX.

In applications with low-to-medium data rates, the PRX can be used to directly record the subband sequences from each filter bank onto a low speed recording medium (such as magnetic tape). The quantized subband sequences could then be transported over a communication link for further software processing at a remote site. The key to this utilization is that the subband sequences are output at a reduced rate, arbitrarily selected by the designer, without prohibitive constraints on the recording rate.

Another area of future research is to augment this structure and derive a new class of architectures tailored for direct sequence spread spectrum communication using the multirate systems approach.

XI. Conclusion

In this report, we succeeded in formulating and devising an architecture for a digital receiver such that the processing rate in the digital signal processing hardware is arbitrarily selected by the designer. A brief overview of multirate and filter bank systems was presented. Each subsystem for a digital receiver was addressed, demodulation, matched filtering, and carrier and symbol synchronization. Specifically, an architecture was devised that operates at the low rate, and the detected symbol stream is directly output from the subbands. Various options for the implementation of the overall receiver were studied and their associated complexities were assessed. Simulation and numerical analysis of the PRX architecture were undertaken, and the symbol error rate obtained in this simulation indicates that there is no loss associated with the PRX when compared to the classical implementation of the receiver.

References

[1]. Sadr R., Shah B., Hinedi S., "*Application of Multirate Digital Filter Banks to Wideband All-Digital Phase-Locked Loops Design*", **TDA Progress Report** 42-111, pp. 101-117, Jet Propulsion Laboratory, Pasadena, CA, November 1992.

[2]. Hinedi S., "*A Functional Description of the Advanced Receiver II*", **TDA Progress Report** 42-100, pp. 150-159, Jet Propulsion Laboratory, Pasadena, CA, February 1990.

[3]. Sadr R., Shahshahani M., "*On Sampling Band-Pass Signals*", **TDA Progress Report** 42-96, pp. 14-20, Jet Propulsion Laboratory, Pasadena, CA, October-December 1988.

[4]. Sadr R., Hurd W., "*Digital Carrier Demodulation for the DSN Advanced Receiver*", **TDA Progress Report** 42-93, pp. 45-63, Jet Propulsion Laboratory, Pasadena, CA, May 1988.

[5]. Simon M.K., Hinedi S., and Lindsey W. C., **Digital Communication Techniques—Signal Design and Detection**, Prentice Hall, N.J., 1994.

[6]. Sadr R., Hurd W., "*Detection of Signals by the Digital Integrate-and-Dump Filter with Offset Sampling*", **TDA Progress Report** 42-91, pp. 158-173, Jet Propulsion Laboratory, Pasadena, CA, Sept. 1987.

[7]. Sadr R., "*Detection of Signals by the Weighted Integrate-and- Dump Filter*", **TDA Progress Report** 42-91, pp. 174-185, Jet Propulsion Laboratory, Pasadena, CA, Sept. 1987.

[8]. Vaidyanathan P.P., **Multirate Systems and Filter Banks**, Prentice Hall, N.J., 1993.

[9]. Vaidyanathan P.P., "*Orthonormal and Biorthonormal Filter Banks as Convolvers, and Convolutional Coding Gain*", **IEEE Trans. on Signal Processing**, Vol. 41, No. 6, June 1993.

[10]. Aguire S., Hurd W., *"Design and Performance of Sampled Data Loops for Carrier and Subcarrier Tracking"*, **TDA Progress Report** 42-79, pp. 81-85, Jet Propulsion Laboratory, Pasadena, CA, July-September 1984.

[11]. Simon M. K., *"Optimization of the Performance of a Digital Data Transition Tracking Loop"*, **IEEE Trans. on Comm.**, Vol. COM-18, No. 3, October 1970.

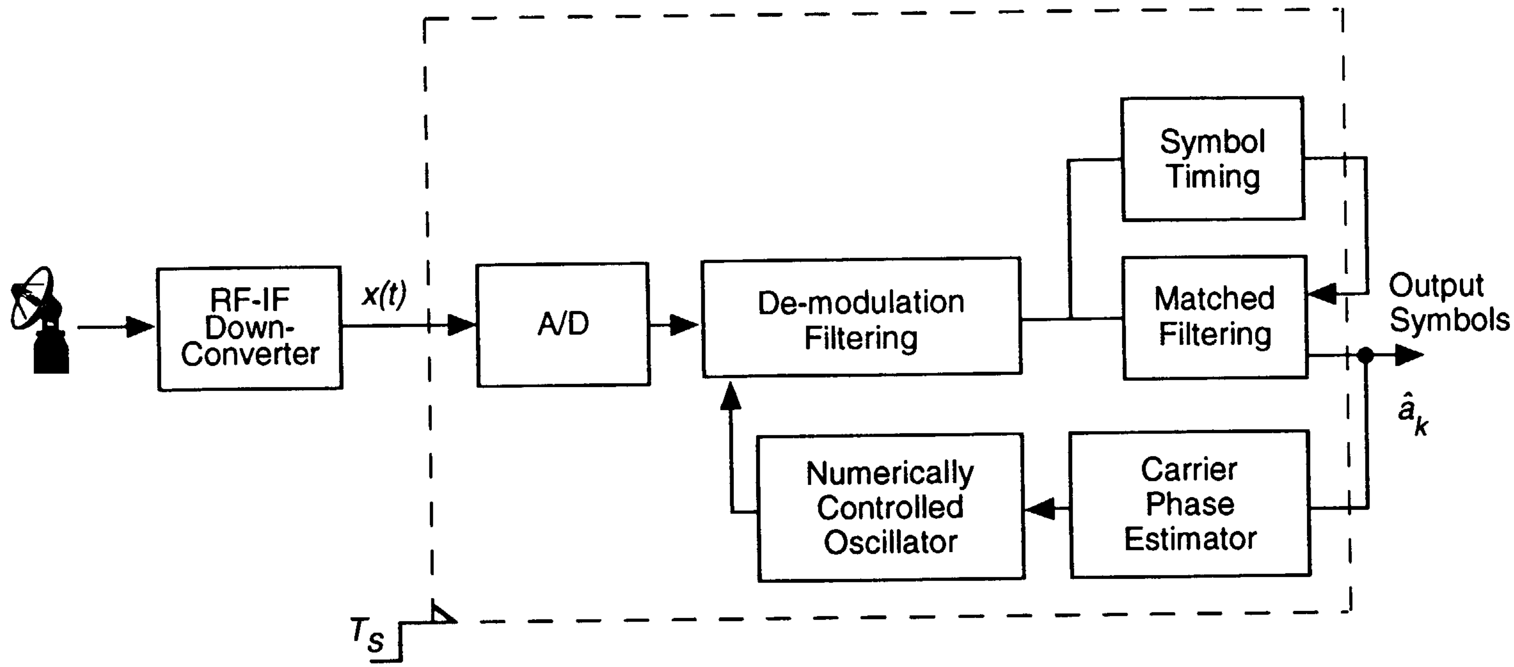

Figure 1. A Typical Digital Receiver

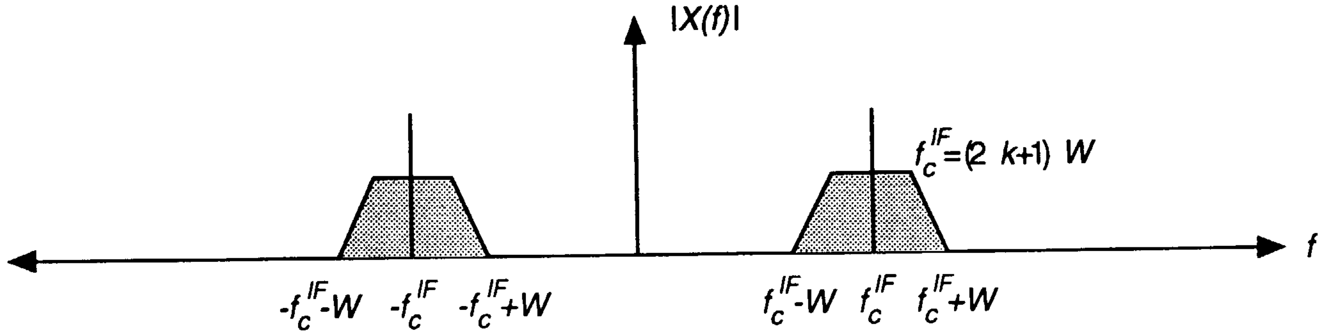

Figure 2. Input Spectrum and IF Carrier for Bandpass Sampling

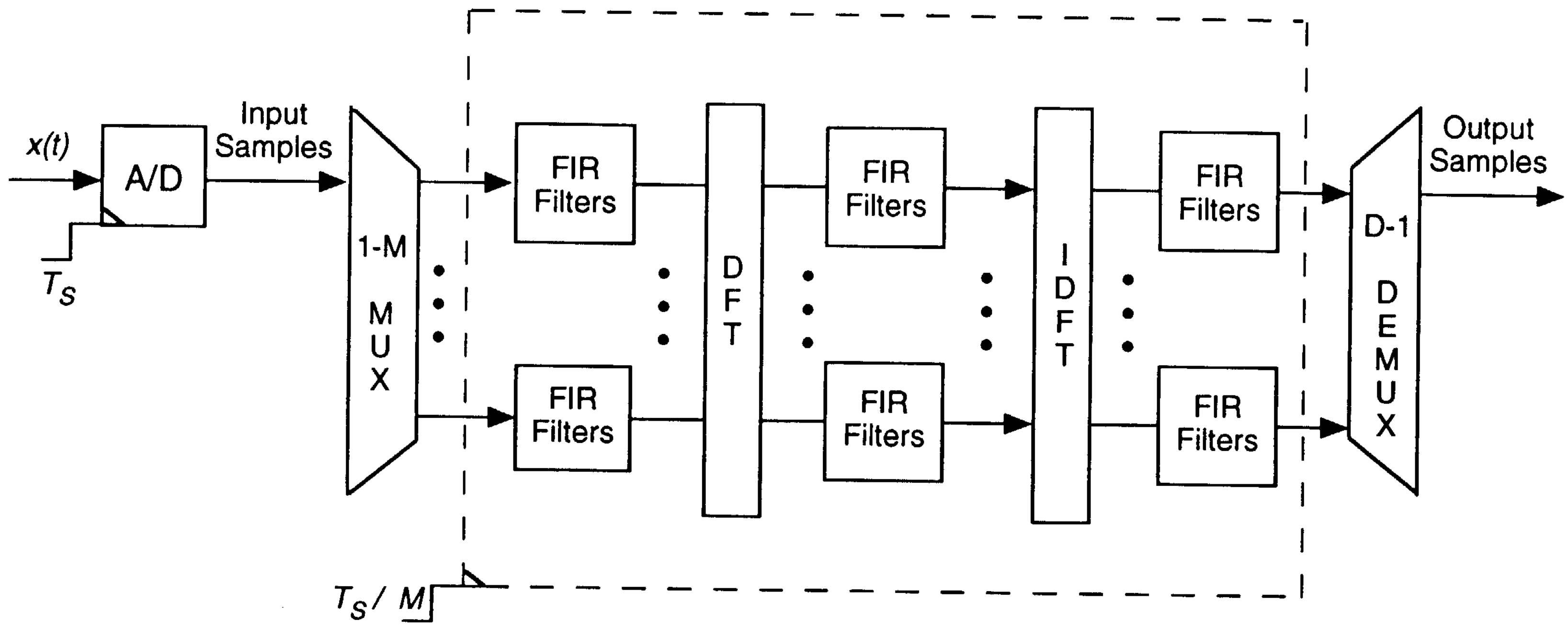

Figure 3. Architecture of PRX

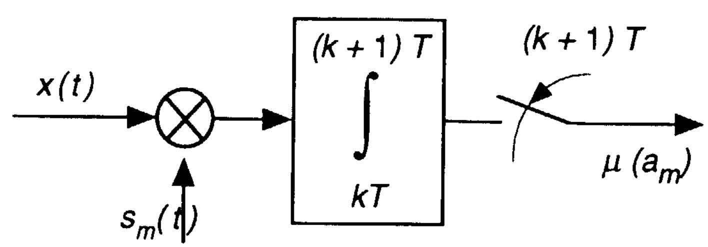

Figure 4. Analog Integrate-and-Dump Filter for Detection of Signals in AWGN

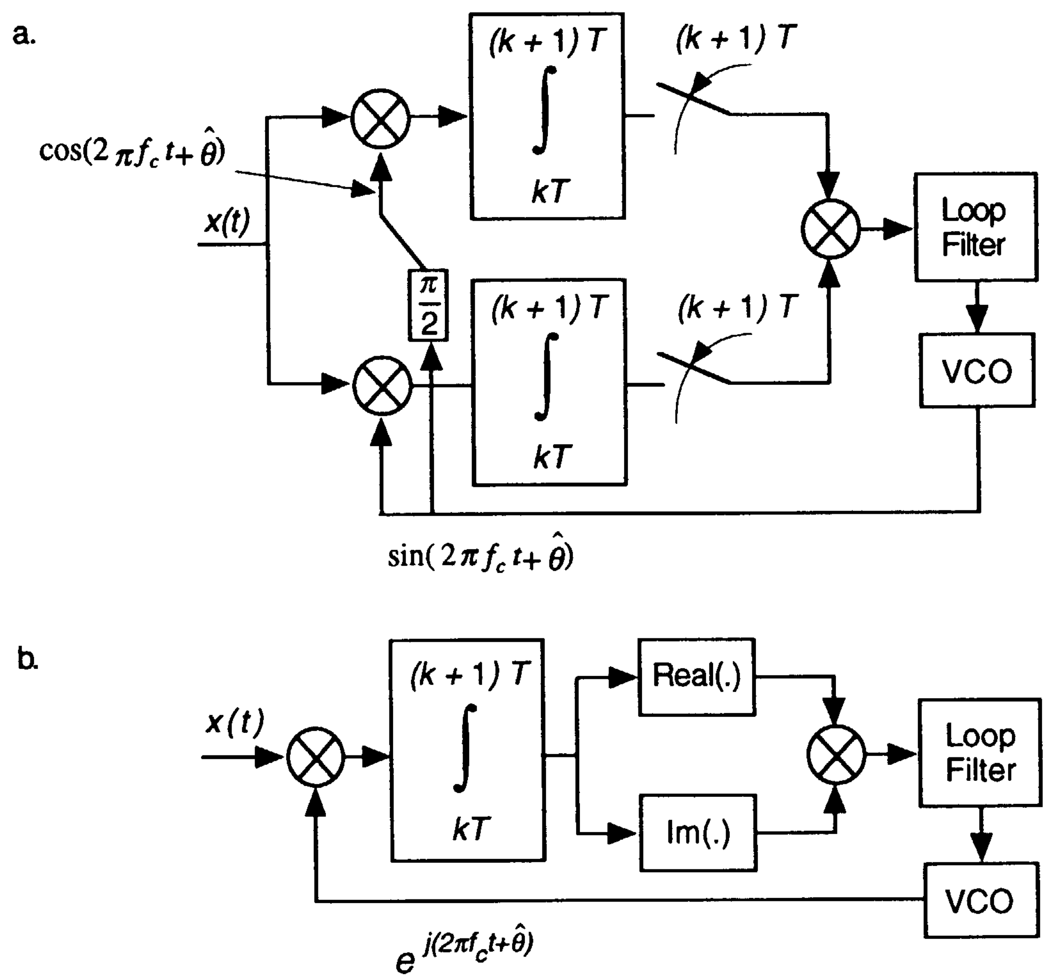

Figure 5. Classical Costas Loop:
a. Original Model,
b. Complex Notation

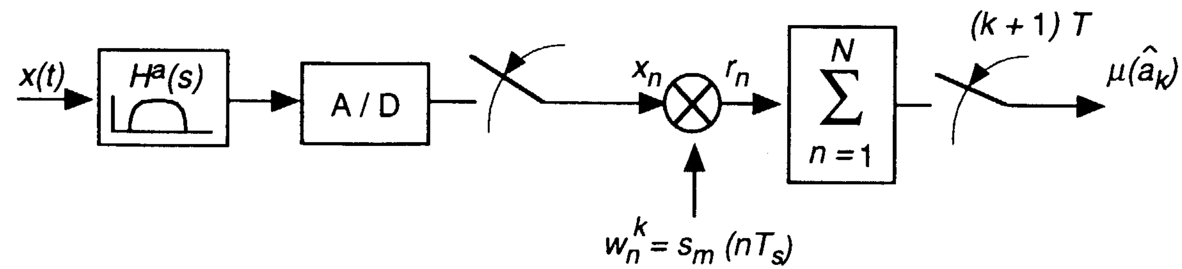

Figure 6. Discrete Time Detector Using Integrate and Dump Filter

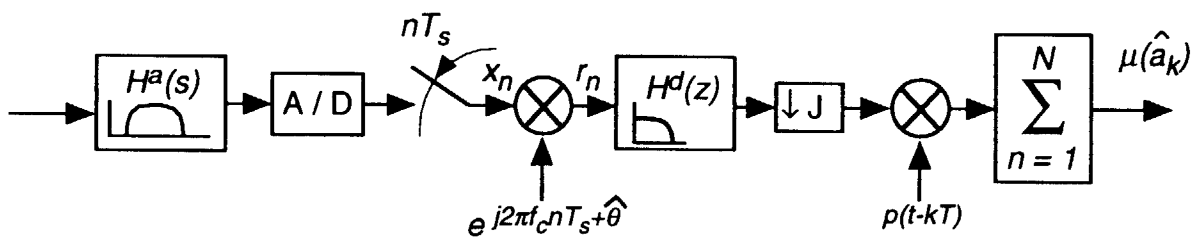

Figure 7. Basic I&Q Demodulator and Matched Filter

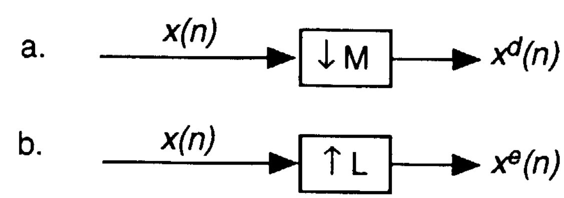

Figure 8. Basic Operations:
a. Decimation,
b. Expansion

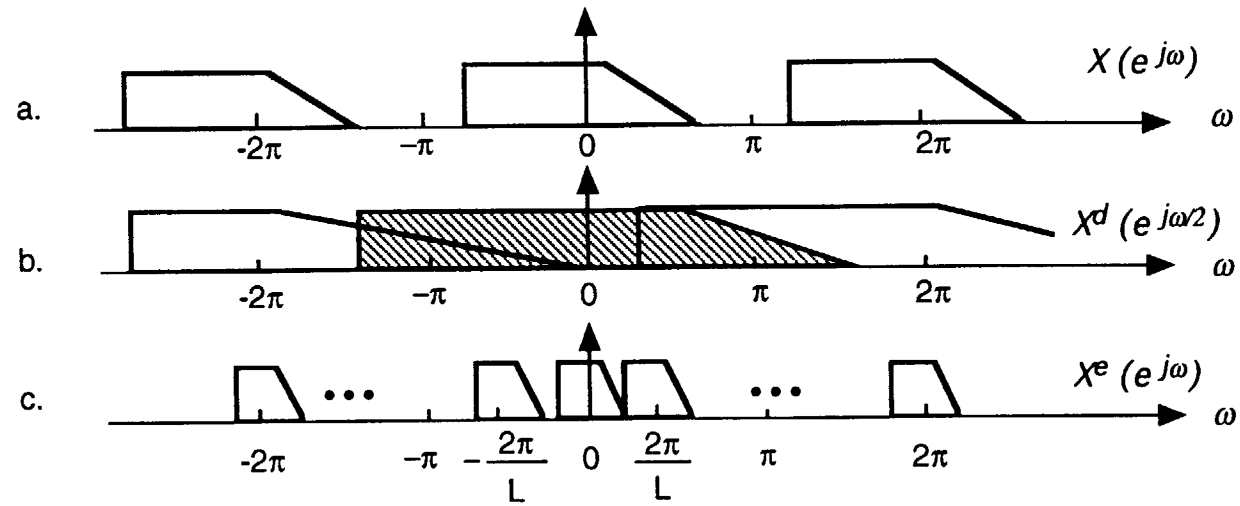

Figure 9. Decimation and Expansion. The Fourier Transform of:
a. Input Signal $x(n)$,
b. Decimated Signal for $M = 2$,
c. Expansion by L

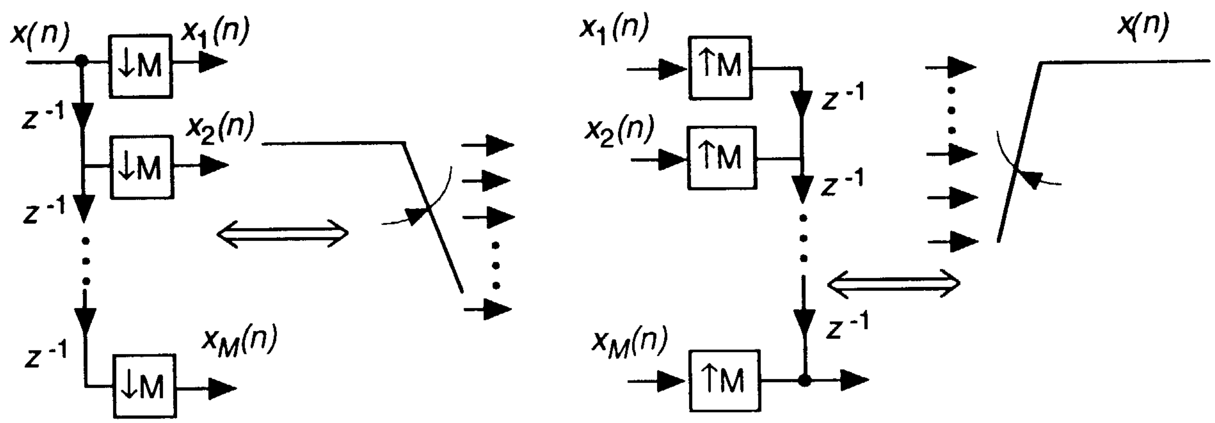

Figure 10. Commutator Model

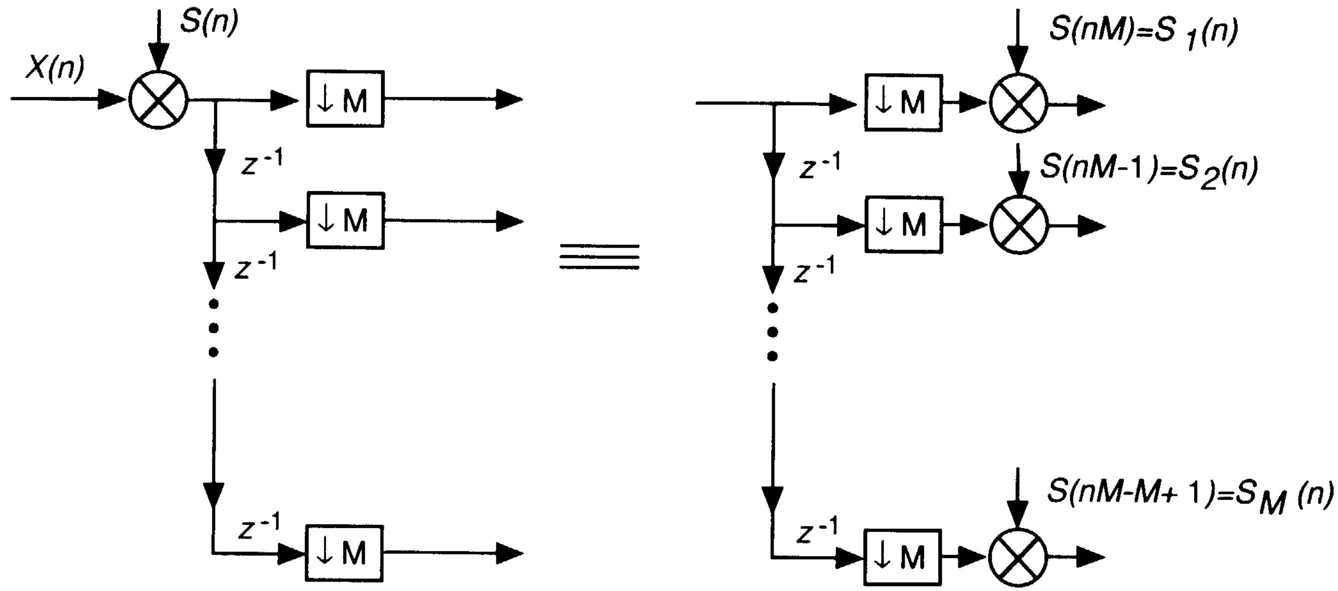

Figure 11. Interchange of Multiplication and Decimation

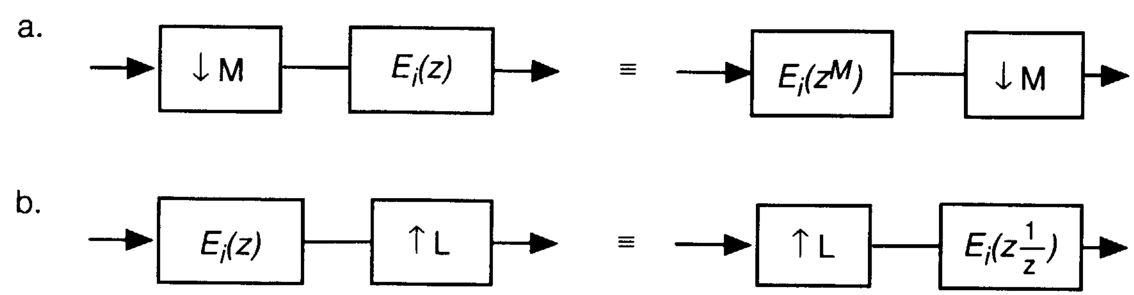

Figure 12. Noble Identity for Multirate Building Block

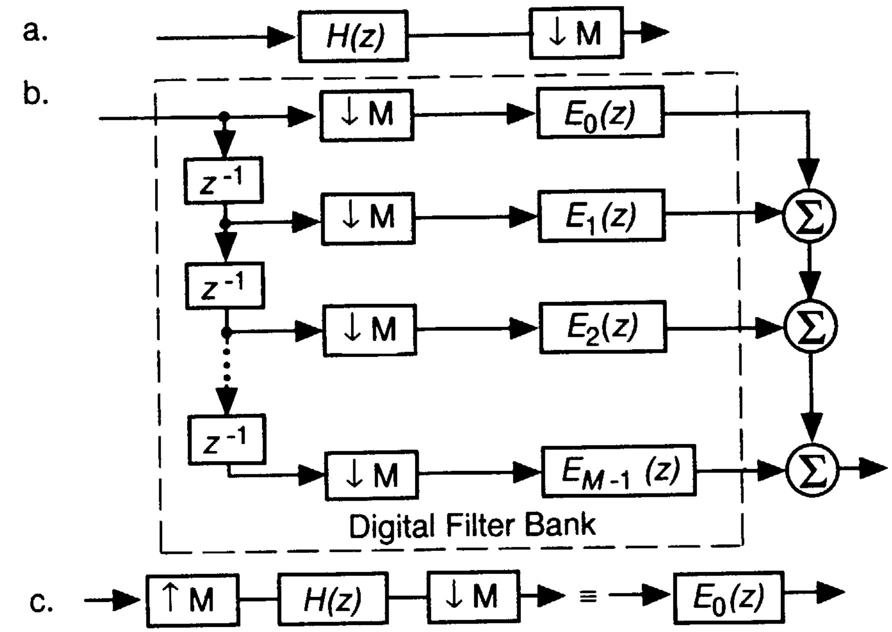

Figure 13. Polyphase Decomposition: a. Original filter, b. Polyphase Implementation, c. Polyphase Identity

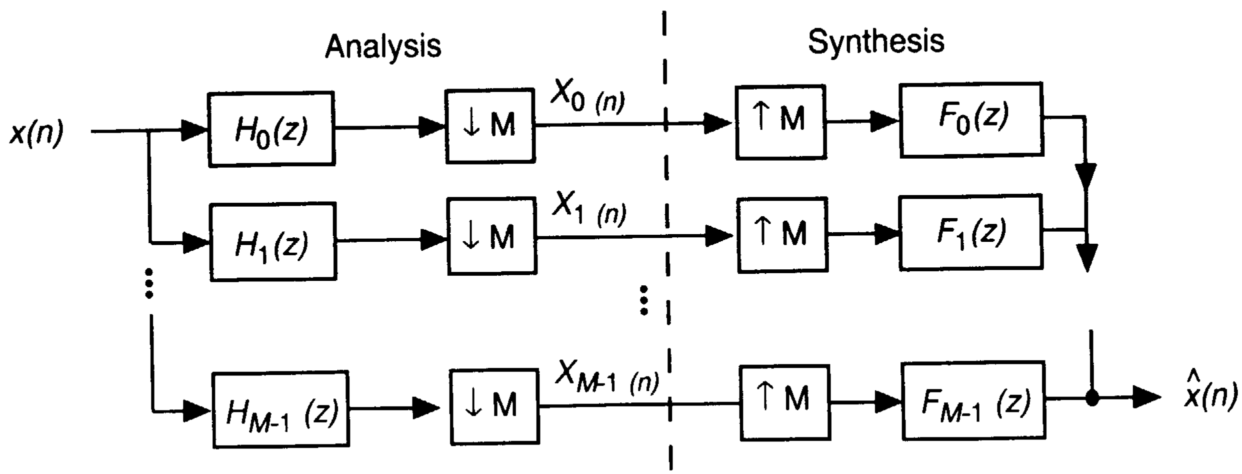

Figure 14. Maximally Decimated Analysis/Synthesis Digital Filter Bank

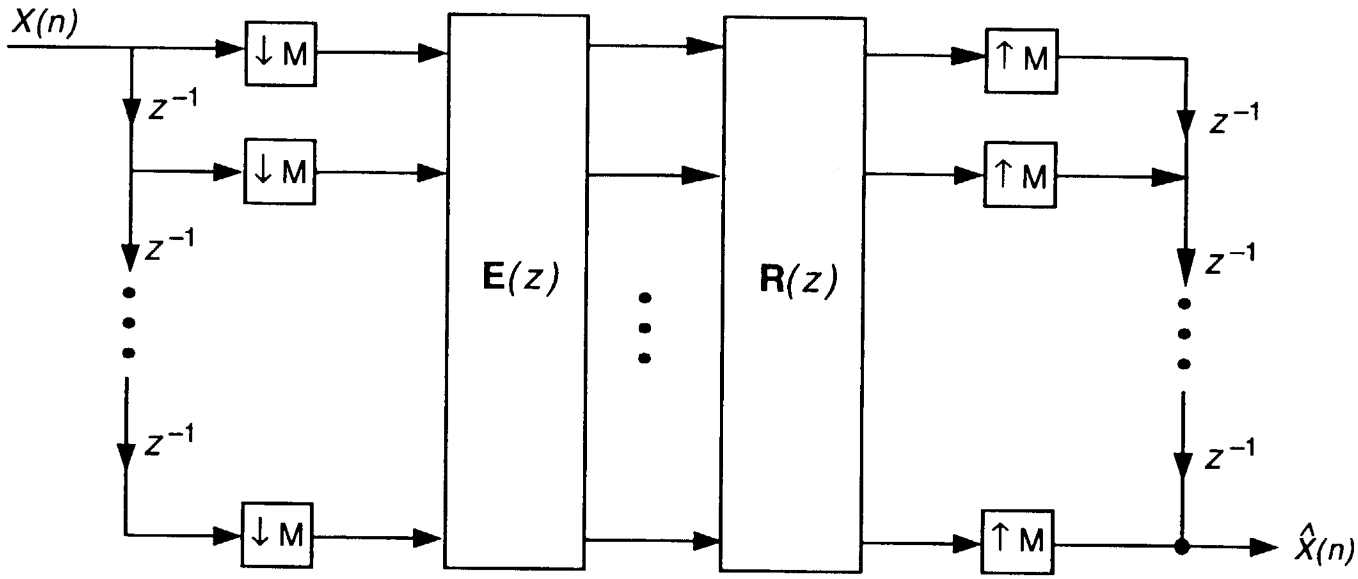

Figure 15. Matrix Polyphase Representation of Analysis/Synthesis Bank

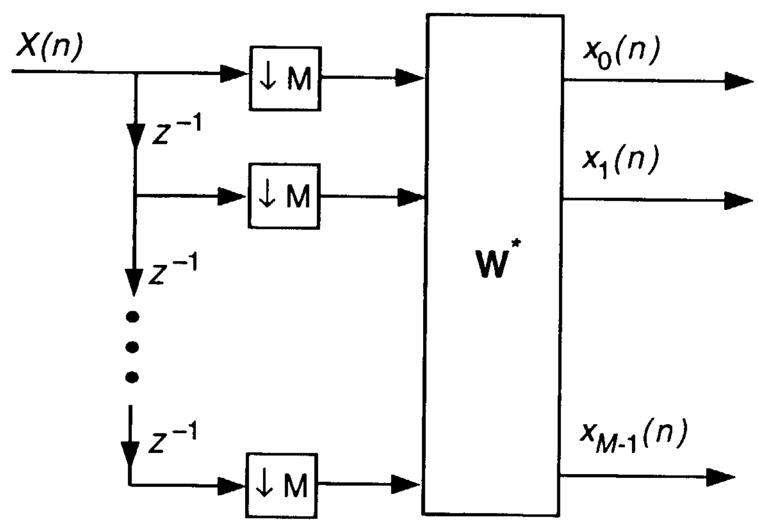

Figure 16. Uniform DFT Filter Bank

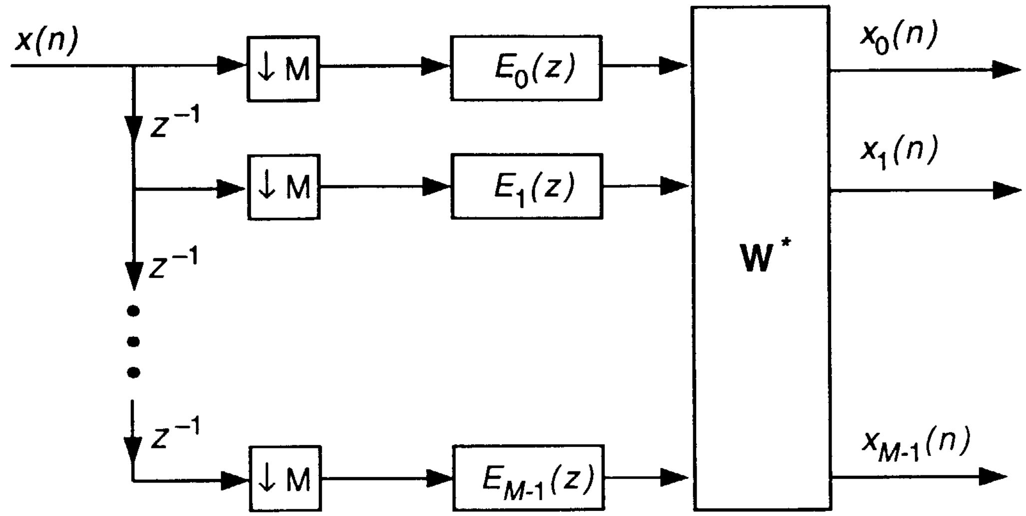

Figure 17. Efficient Implementation of Maximally Decimated DFT Filter Bank

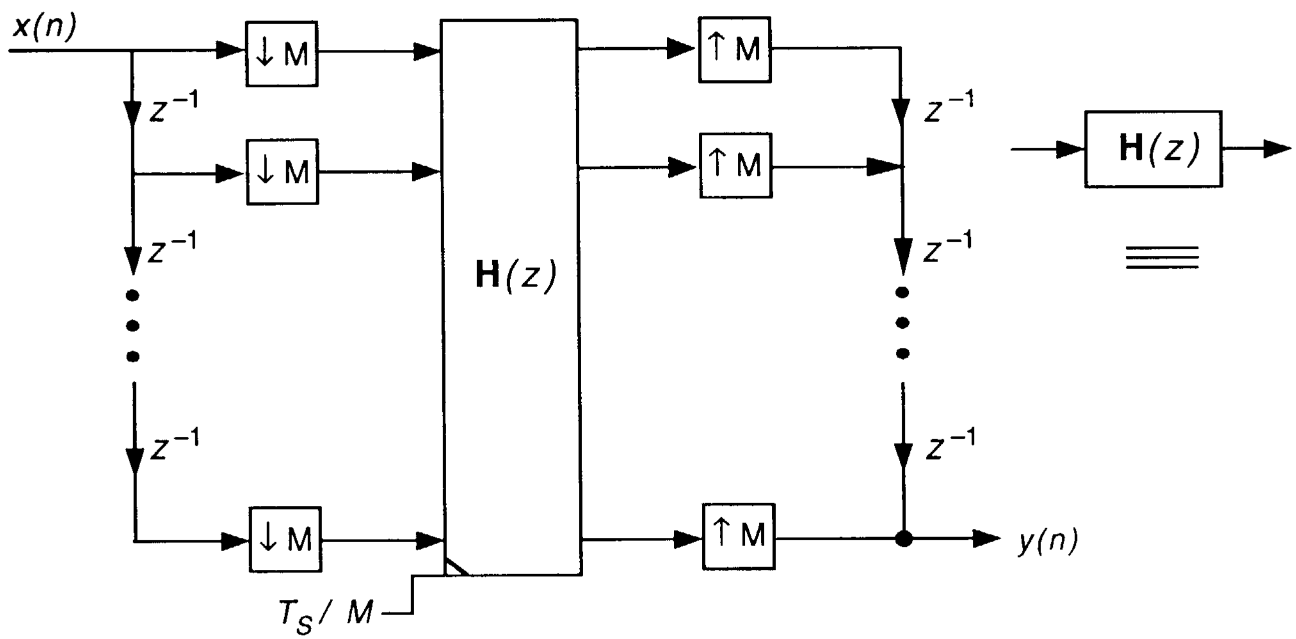

Figure 18. Blocked Digital Filtering

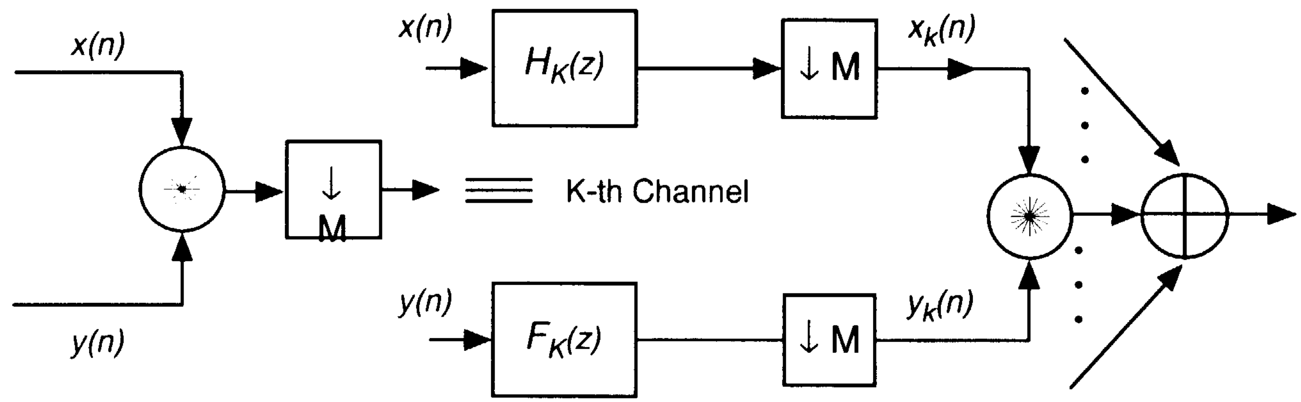

Figure 19. Subband Convolution Theorem: Implementation of the Paraunitary Convolver

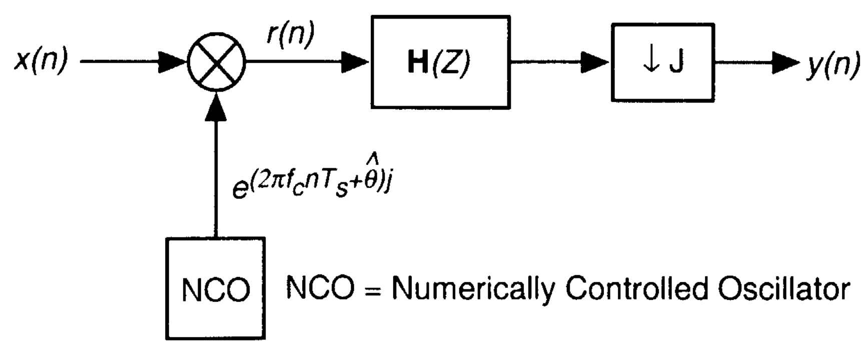

Figure 20. Demodulation and Filtering

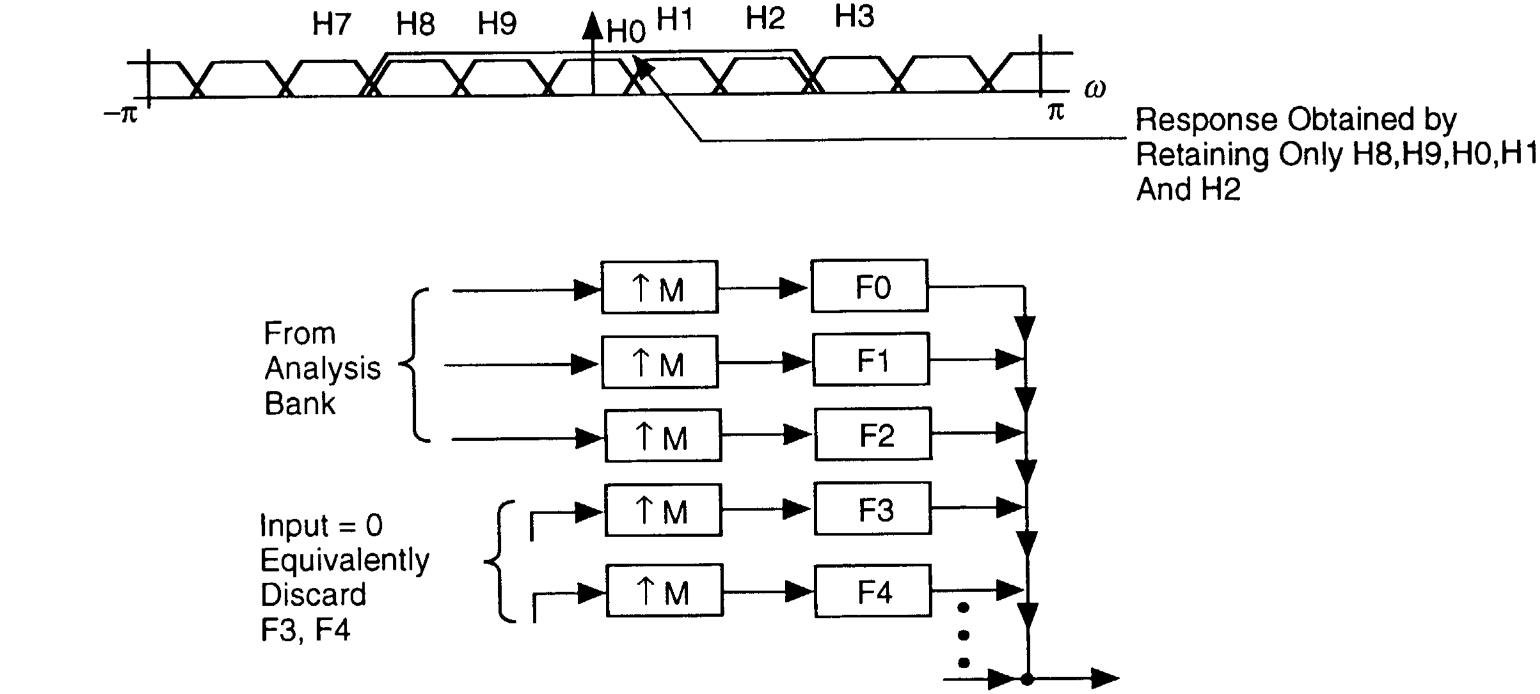

Figure 21. Discarding of the Synthesis Filters for Filtering Operation

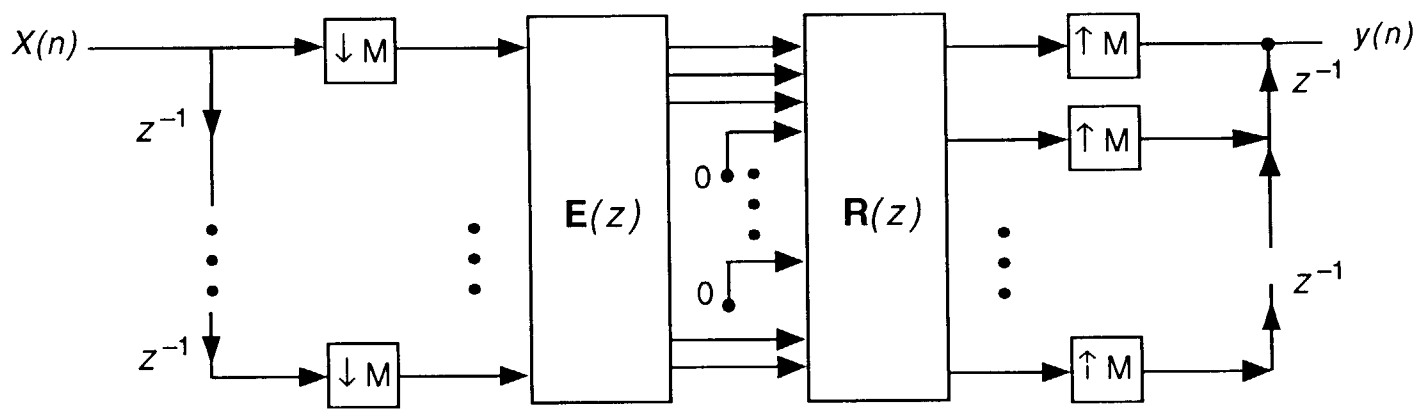

Figure 22. Effects of Discarding in the Structure of DFB

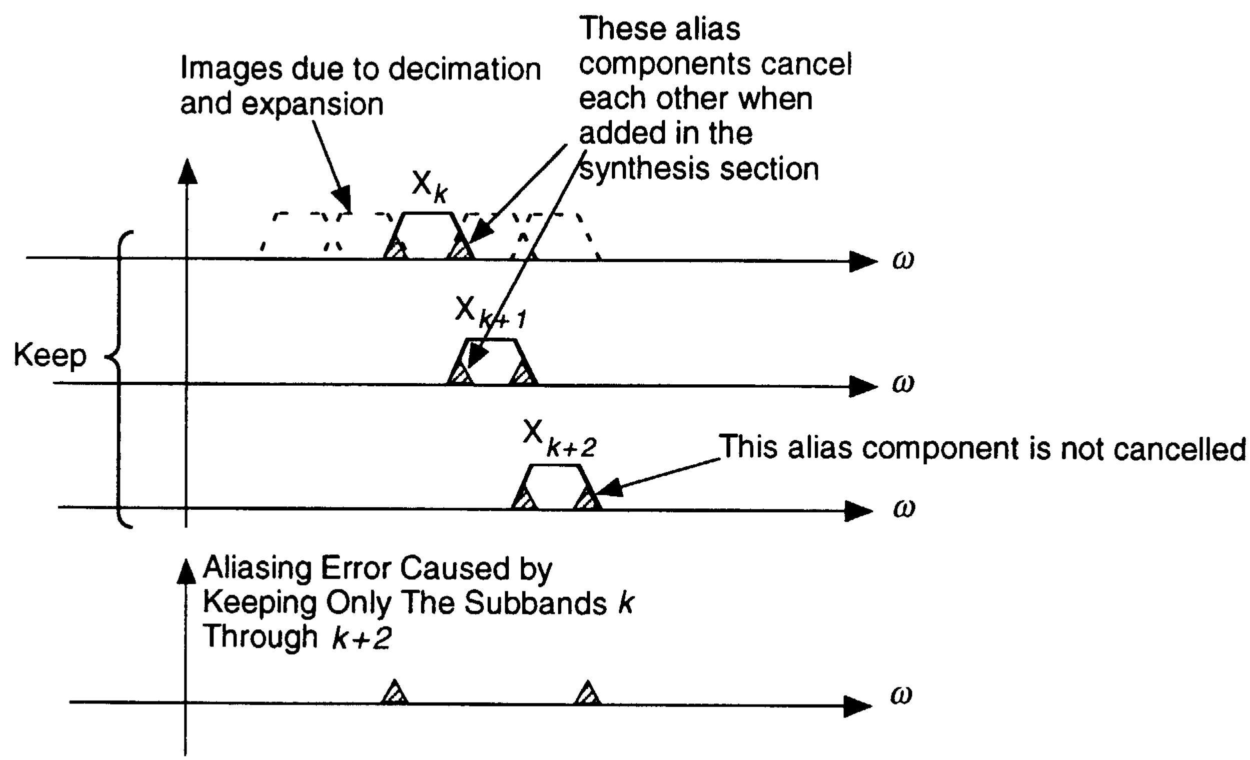

Figure 23. Aliasing Error Due to Dropping of Synthesis Filters

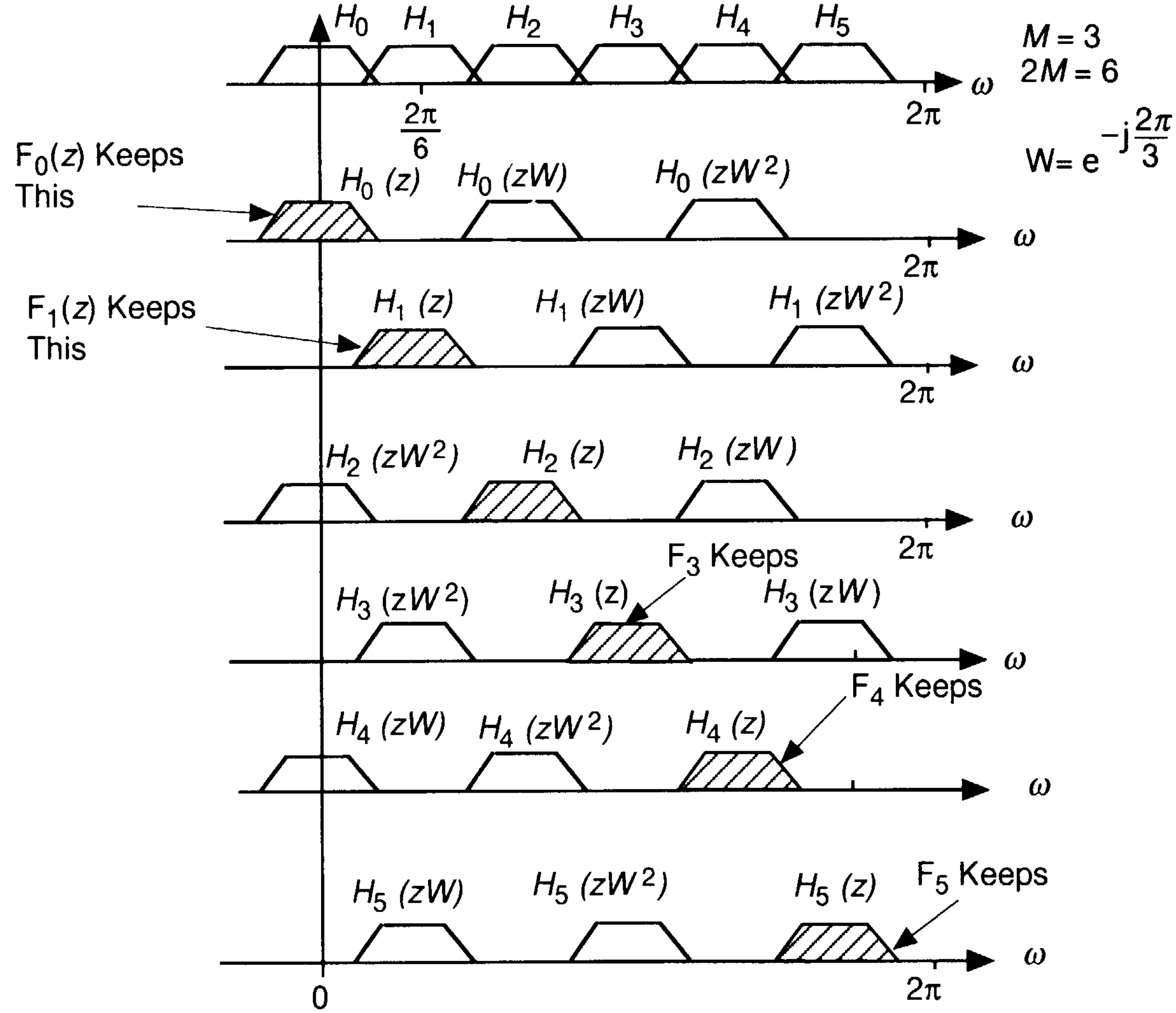

Figure 24. Subbands' Spectrums After the Decimation-Expansion in a Non-Maximally Decimated Filter Bank, *M* = 3

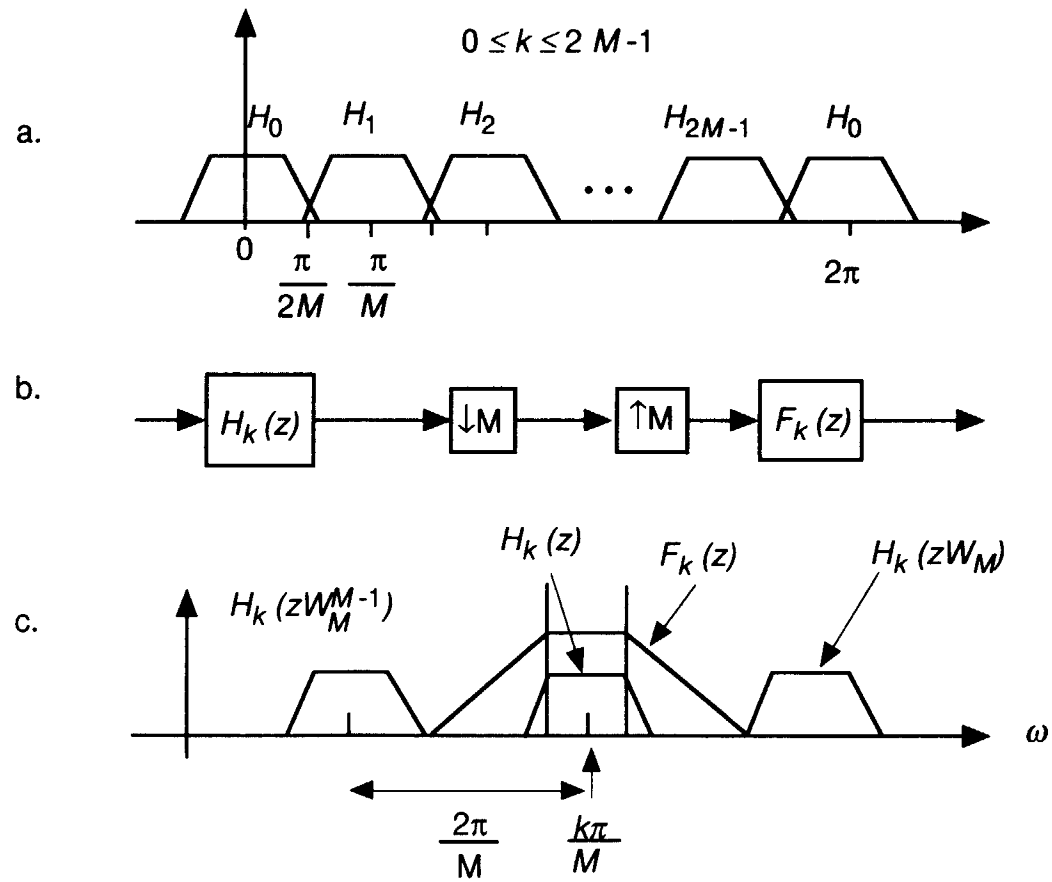

Figure 25. Non-Maximally Decimated Filter Bank: a. Analysis Filter Bank Frequency Support, b. Subband Channel, c. Typical Frequency Response of Synthesis Filter with Relation to the Signal and its Images

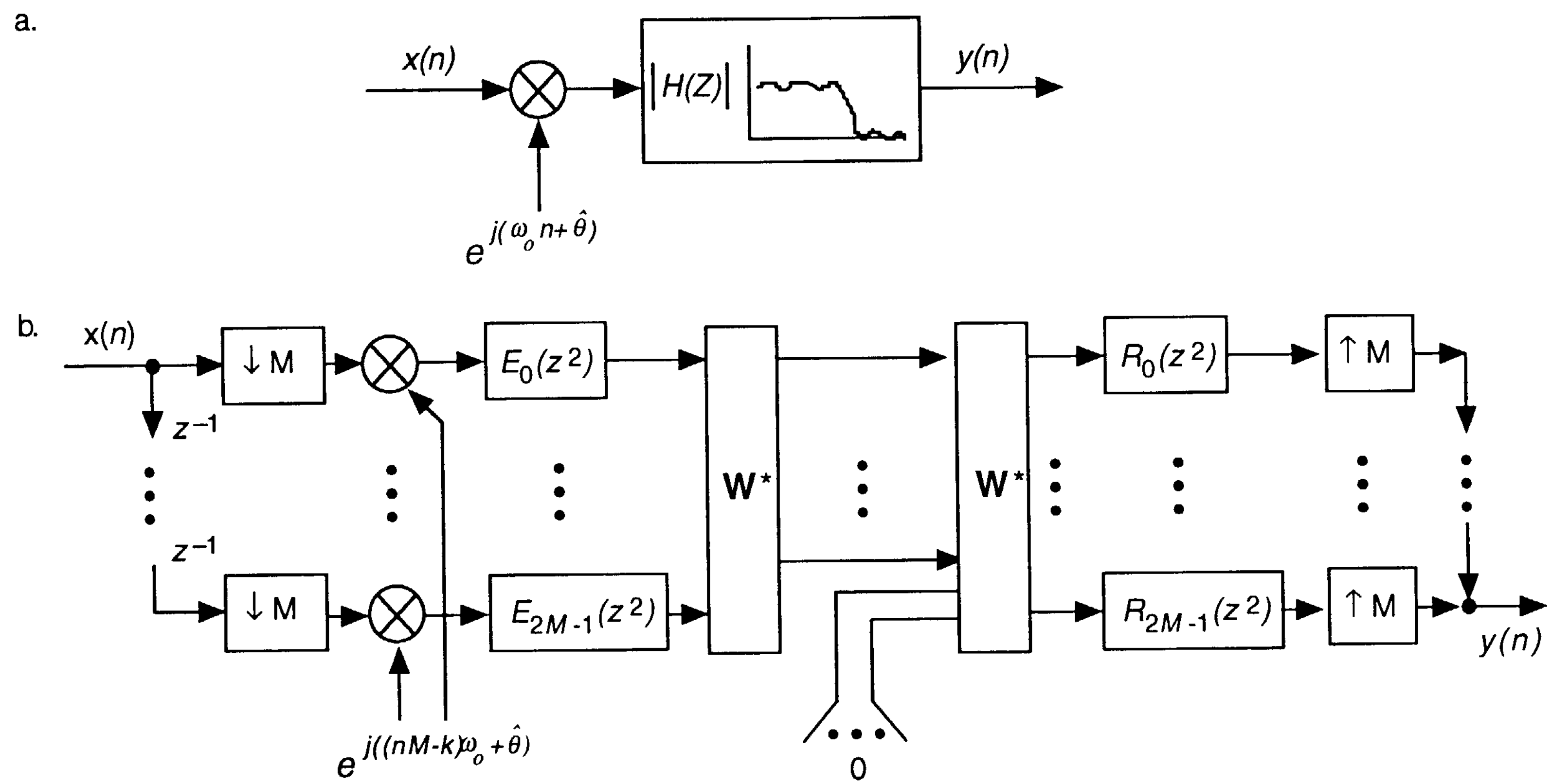

Figure 26. Demodulator Design:
a. Basic Model,
b. Parallel Model

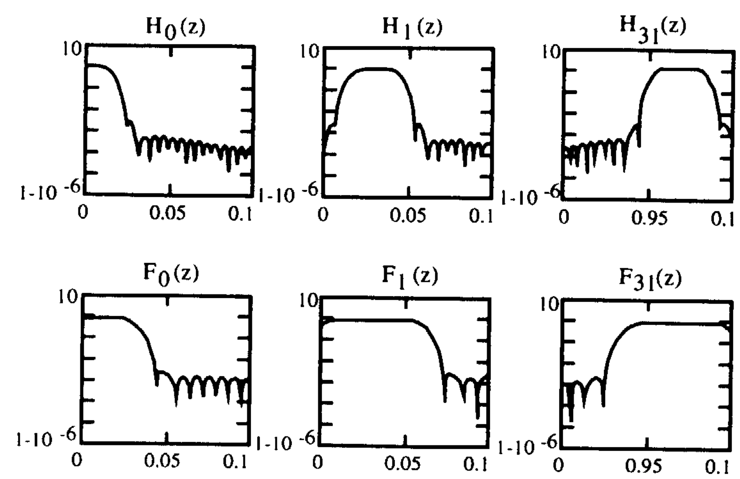

Figure 27. Example of Filter Bank Frequency Response for 2*M* = 32

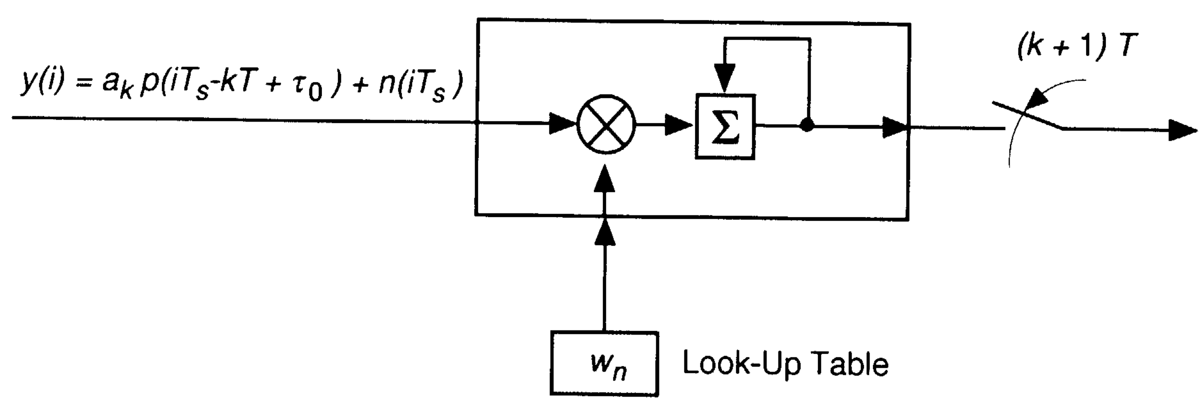

Figure 28. Digital Matched Filtering for Signals in AWGN

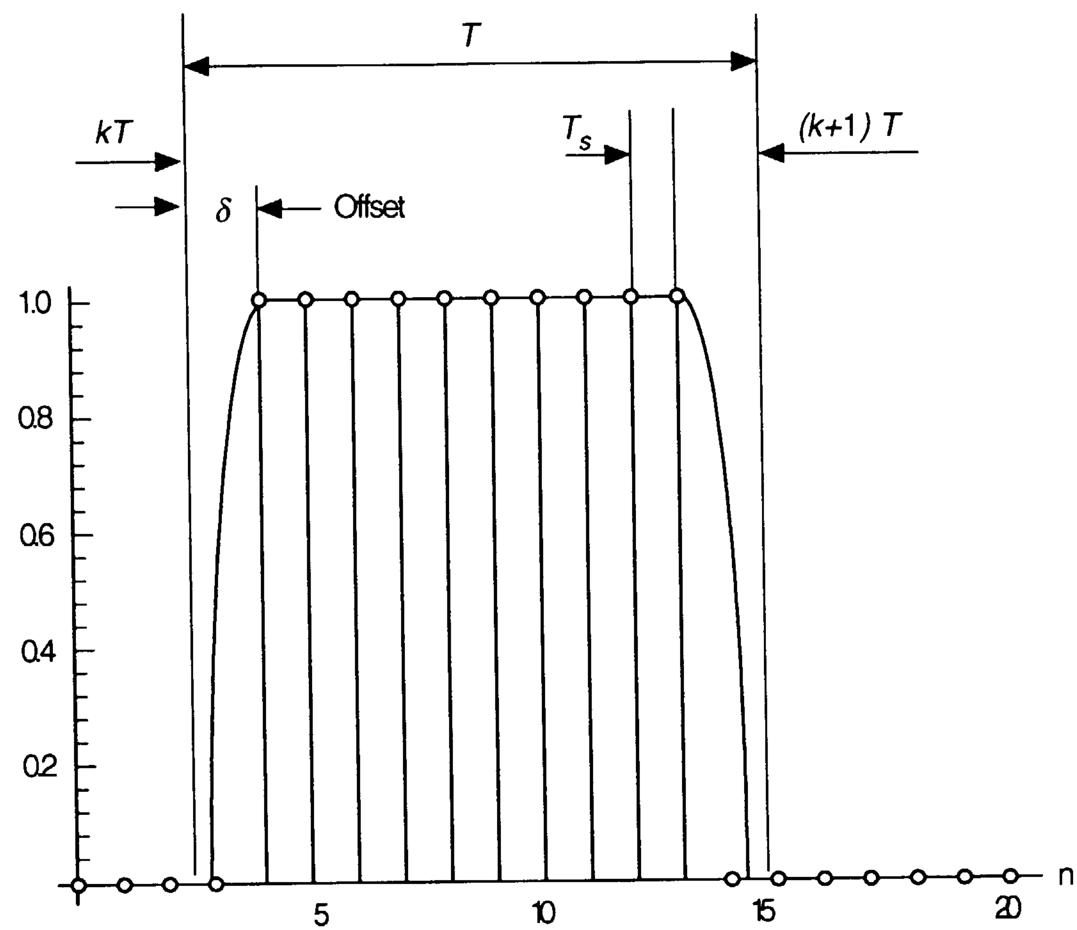

Figure 29. Illustration of Offset in Sampling a Symbol

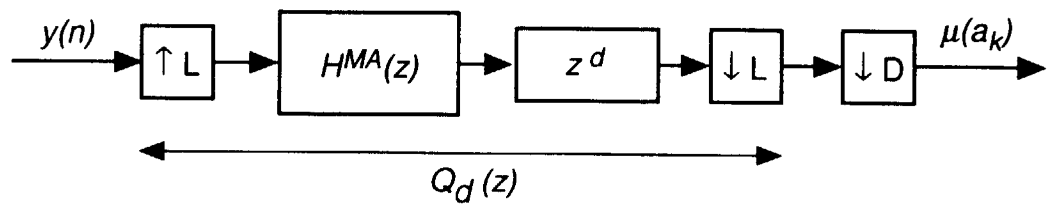

Figure 30. Multirate Digital Filter for Matched Filtering

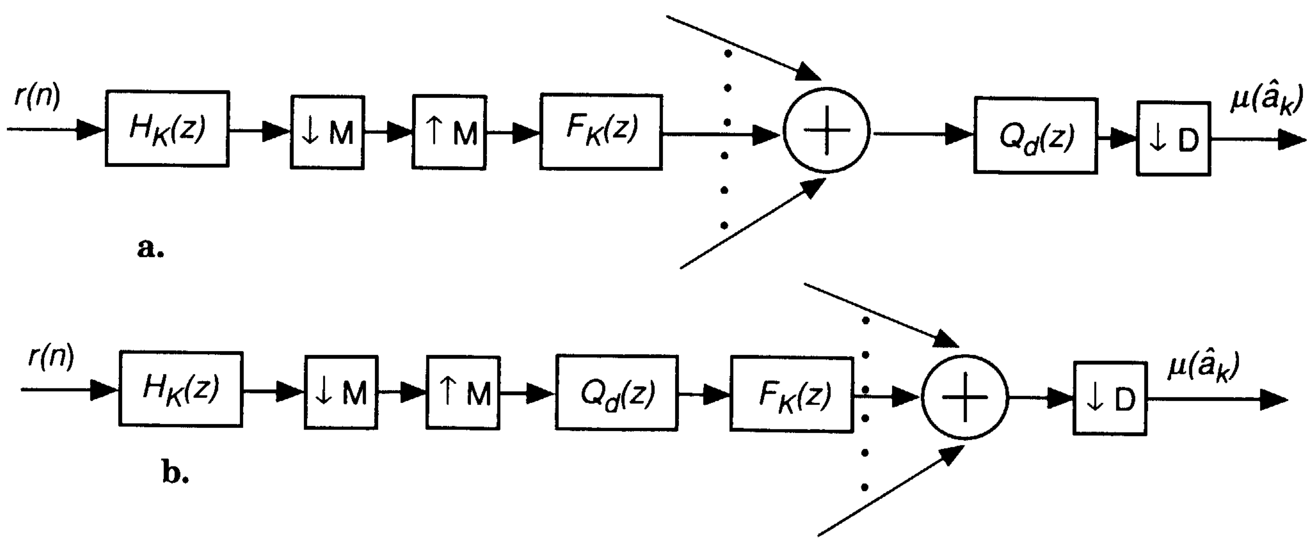

Figure 31. Combining the Matched Filtering and Synthesis Filter Banks:
a. Concatenation of the Filter Bank with the Matched Filter
b. Commuting $Q_d(z)$ with $F_k(z)$

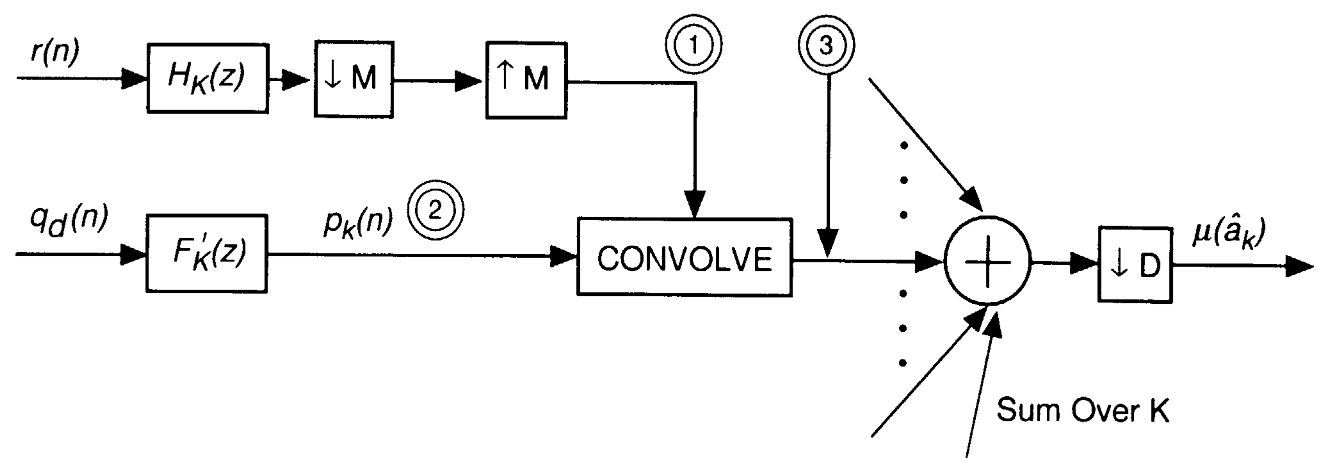

Figure 32. Equivalent Model Using DFB Convolution Theorem

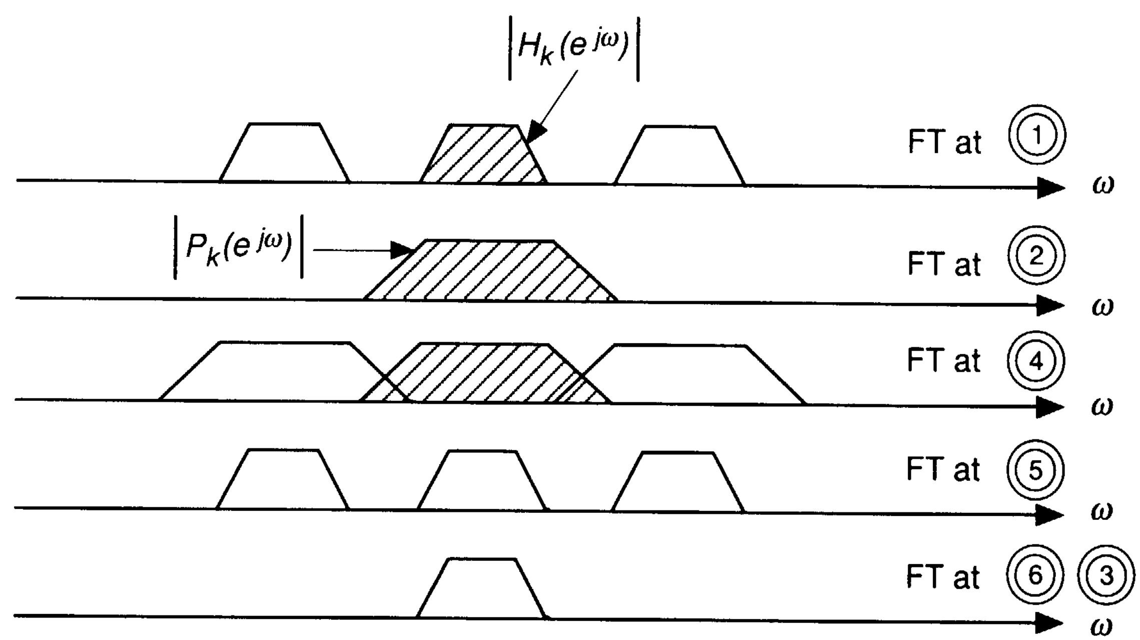

Figure 33. Fourier Transform Support of Signals in Figure 32

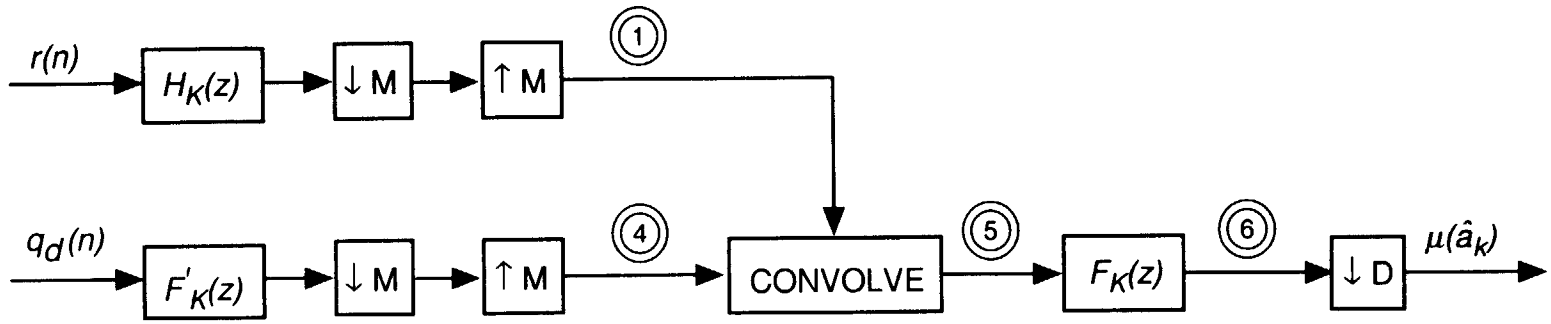

Figure 34. k-th Signal Path of Figure 31

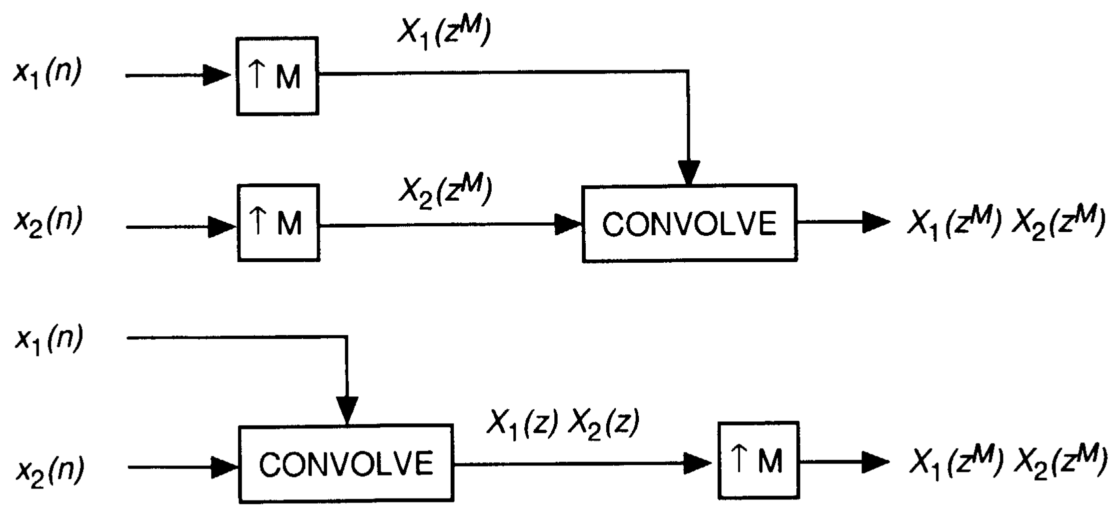

Figure 35. General Identity for Multirate Convolution

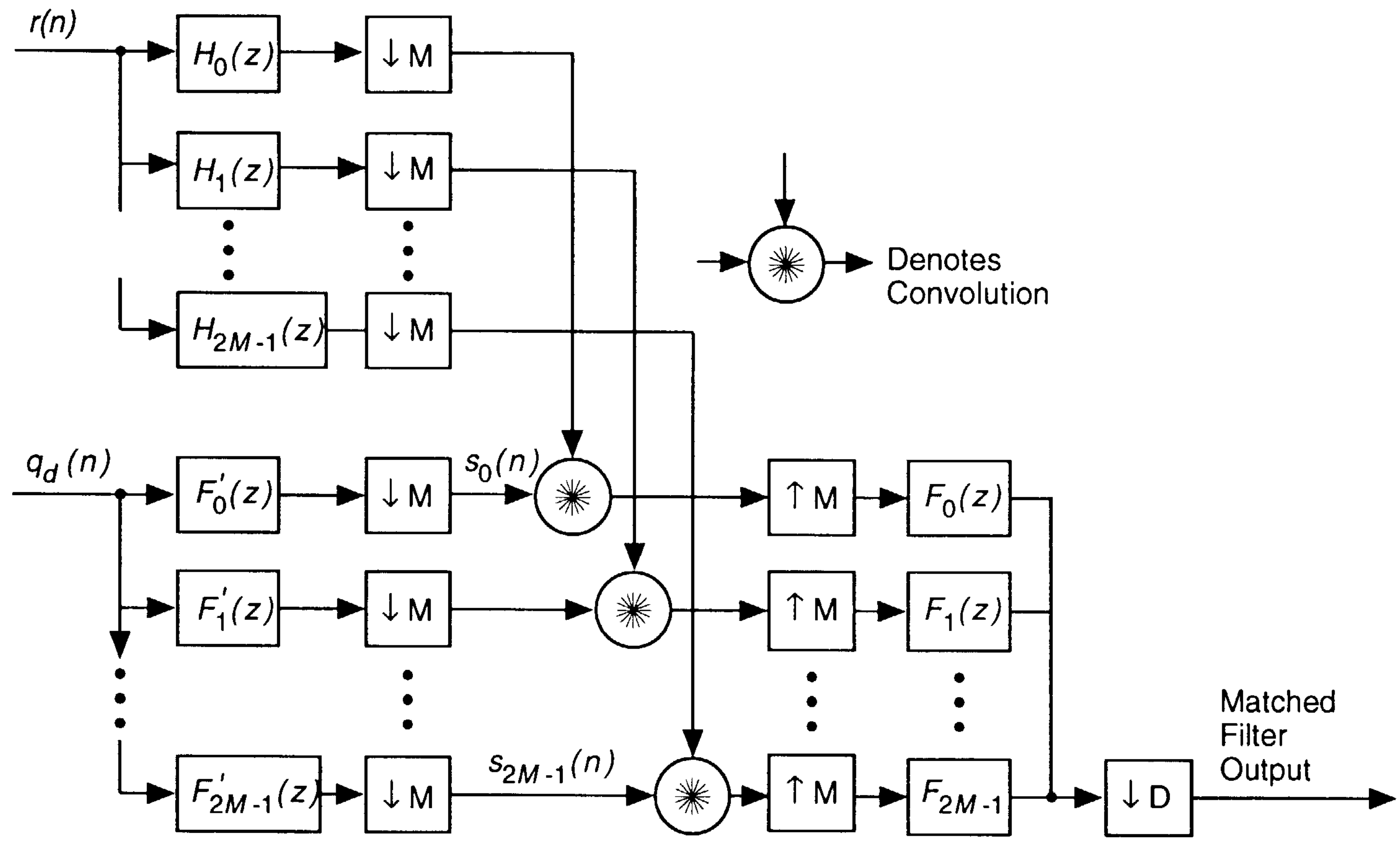

Figure 36. Subband Version of Matched Filter

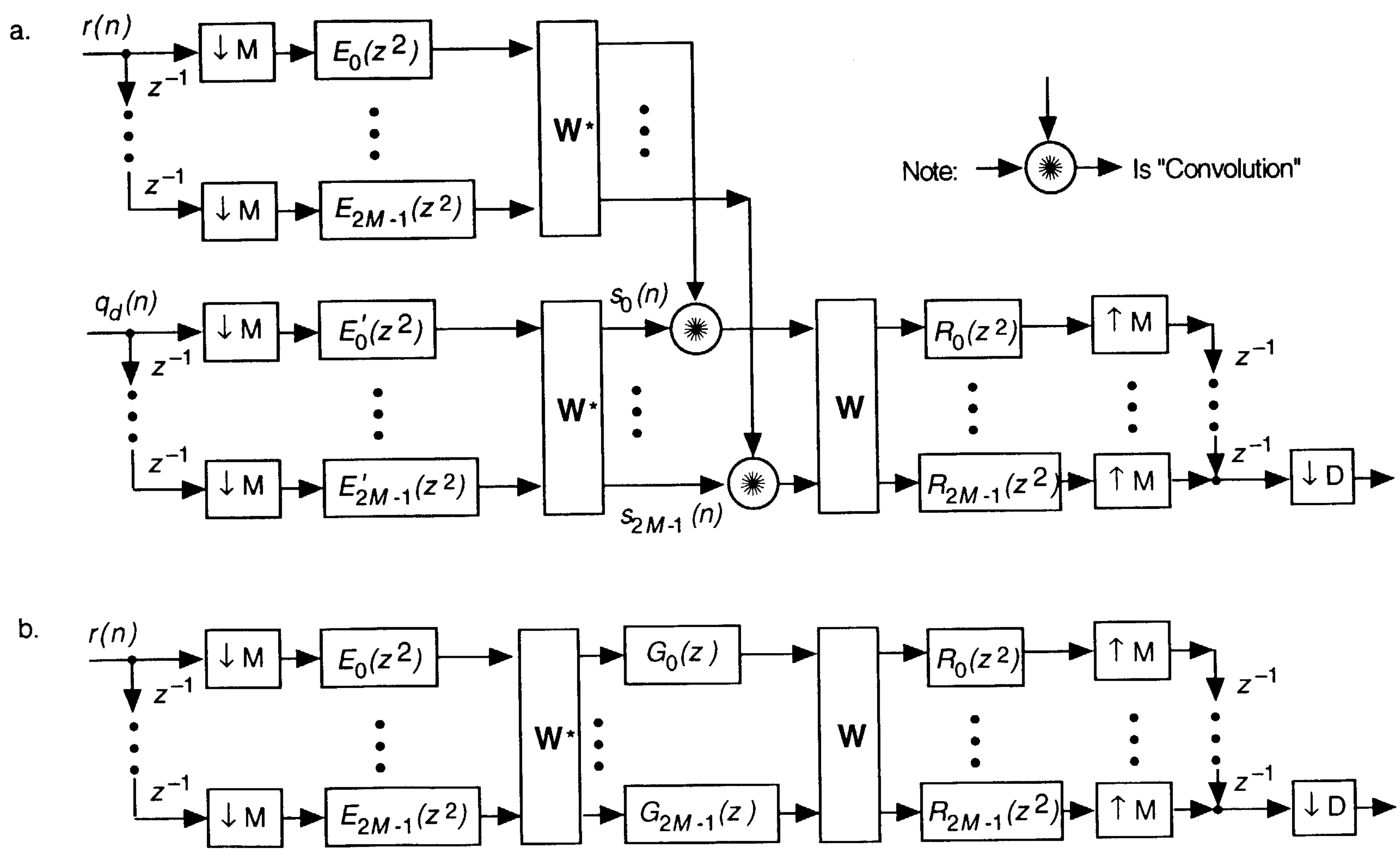

Figure 37. Final Structure of Combined Filtering and Matched Filter:
a. Basic Model,
b. Implementation Model

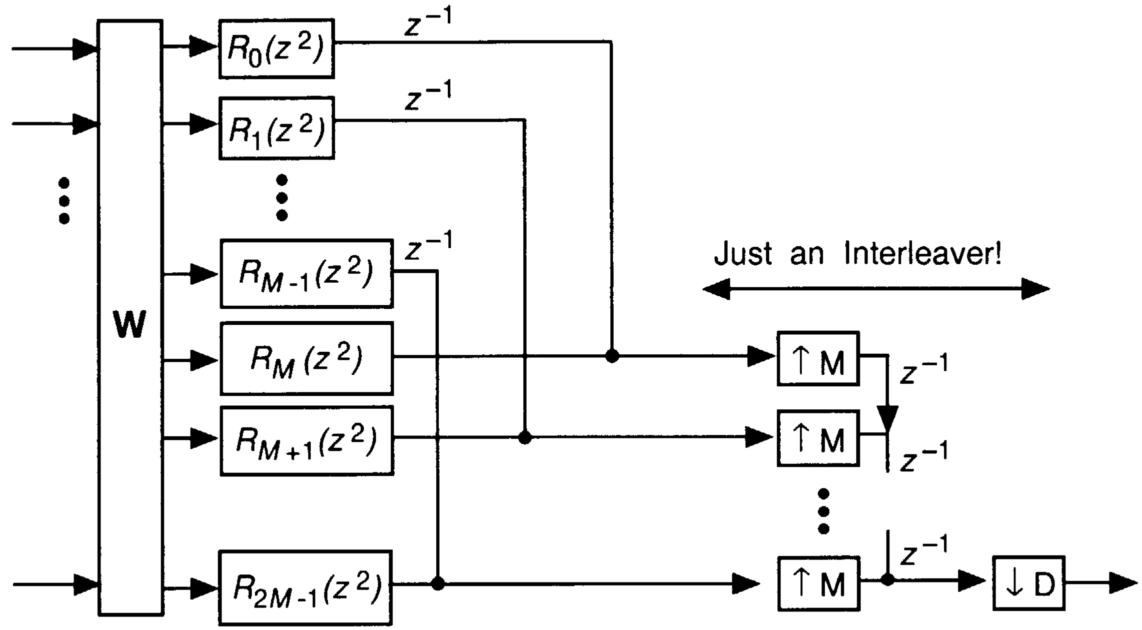

Figure 38. Synthesis Filter Section with Addition Operation at the Lower Rate

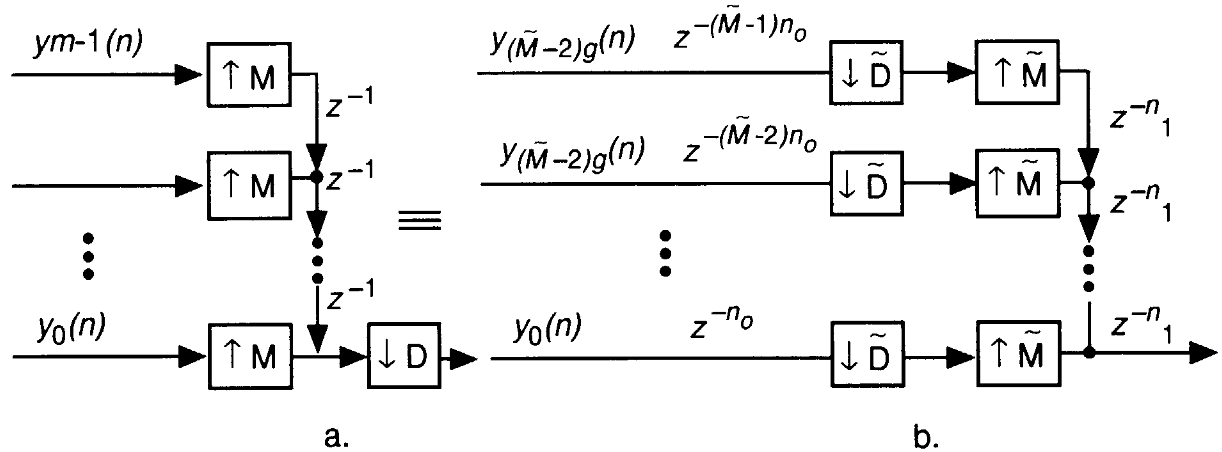

Figure 39. Multirate Identity for Exchanging Decimation and Expansion
a. Original Model, b. Equivalent Model

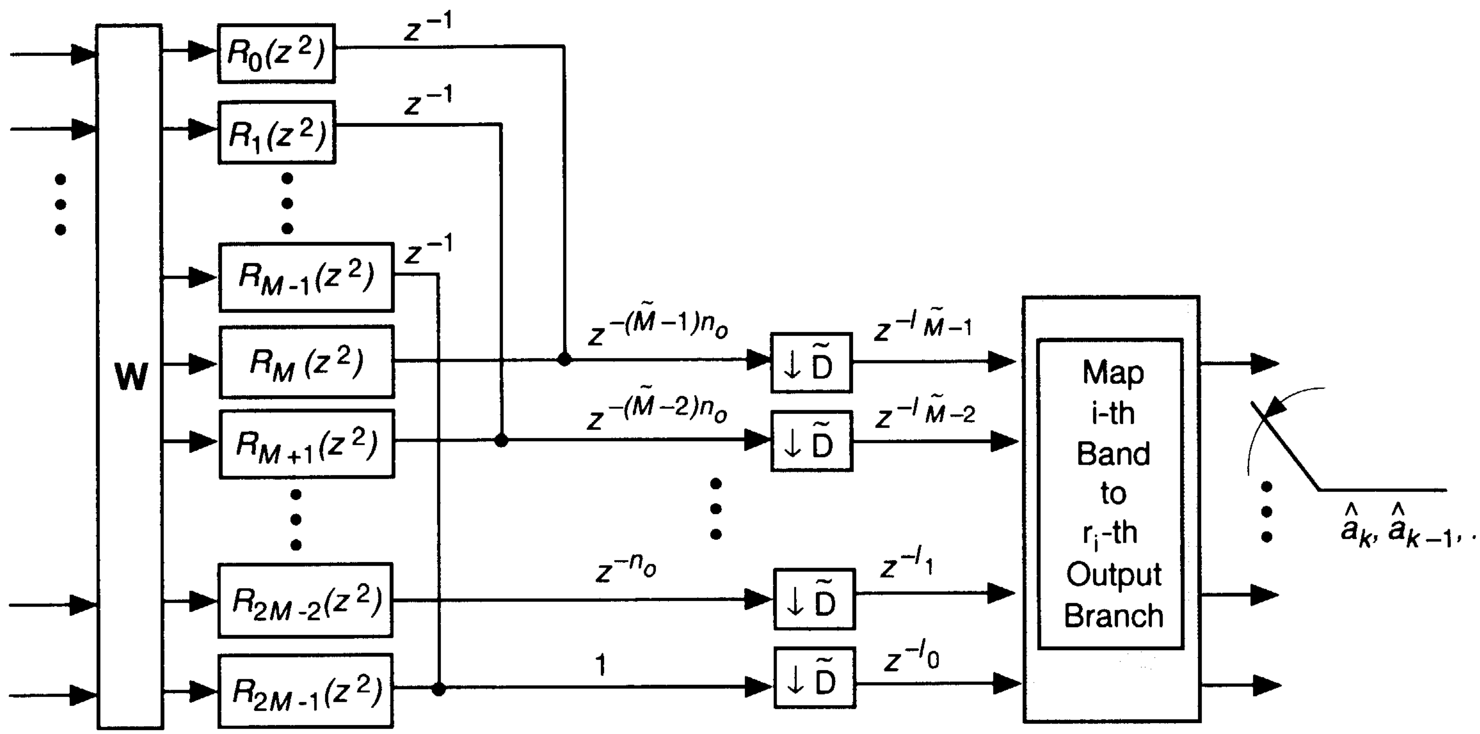

Figure 40. Symbol Stream Selection from the Subband

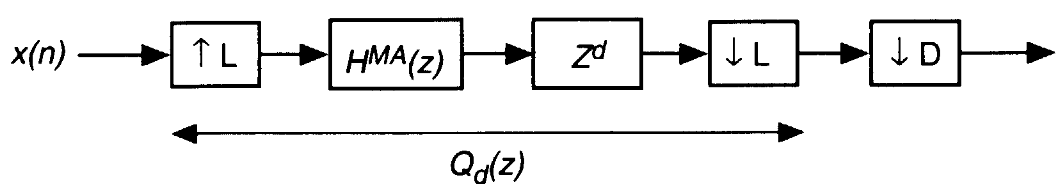

Figure 41. Digital Matched Filter Model

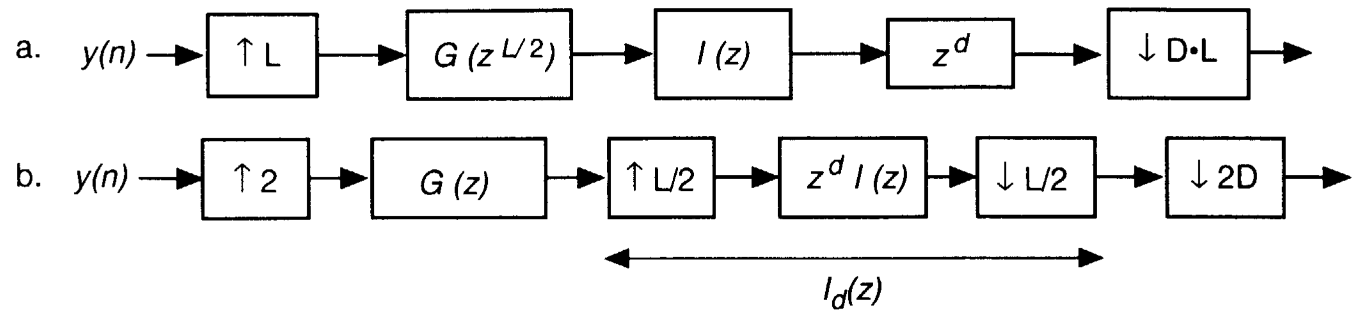

Figure 42. Interpolated Finite Impulse Response
a. Matched Filter,
b. Dividing the Expansion Rate

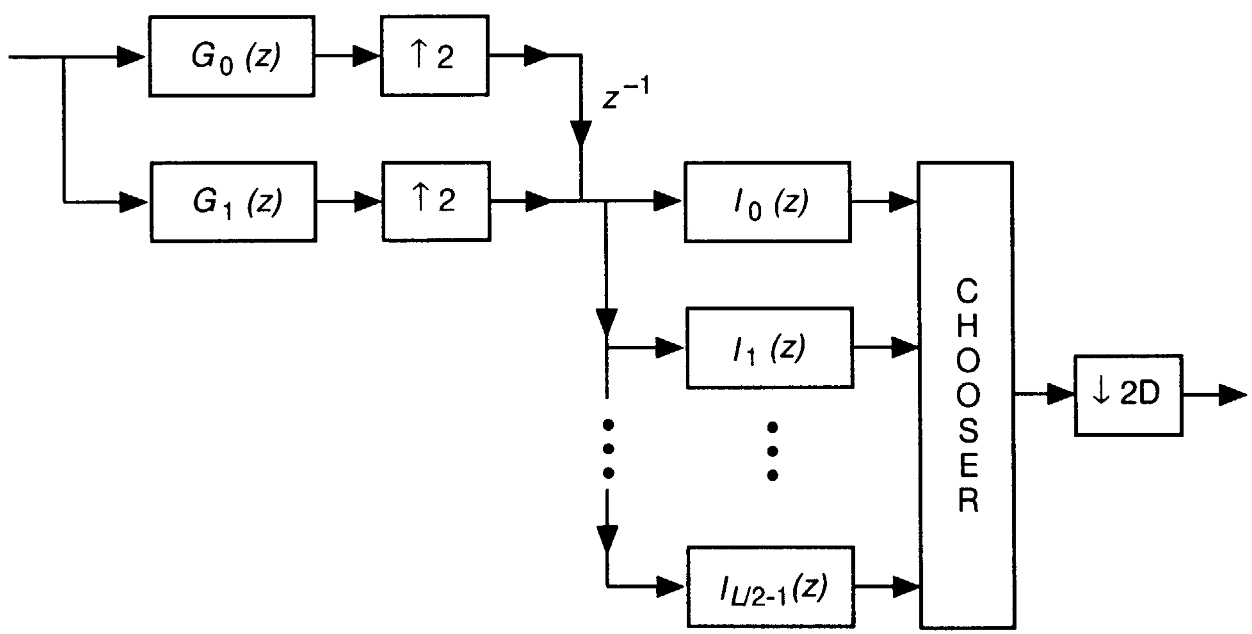

Figure 43. IFIR Implementation of Matched Filtering

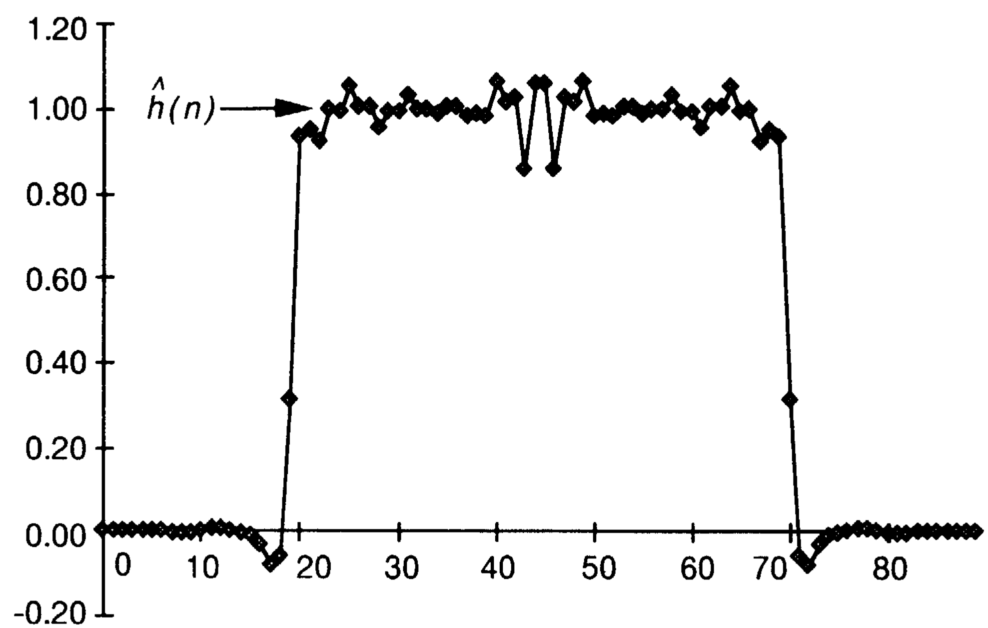

Figure 44. Example of Matched Filter Approximation Using IFIR

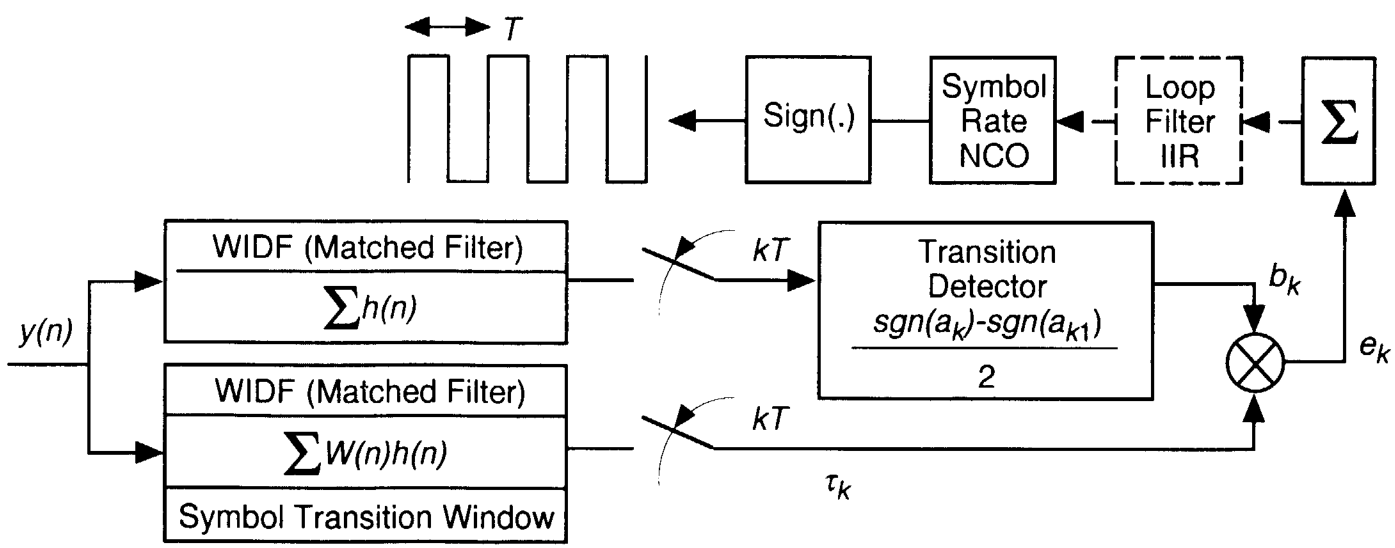

Figure 45. Digital Transition Tracking Loop Block Diagram

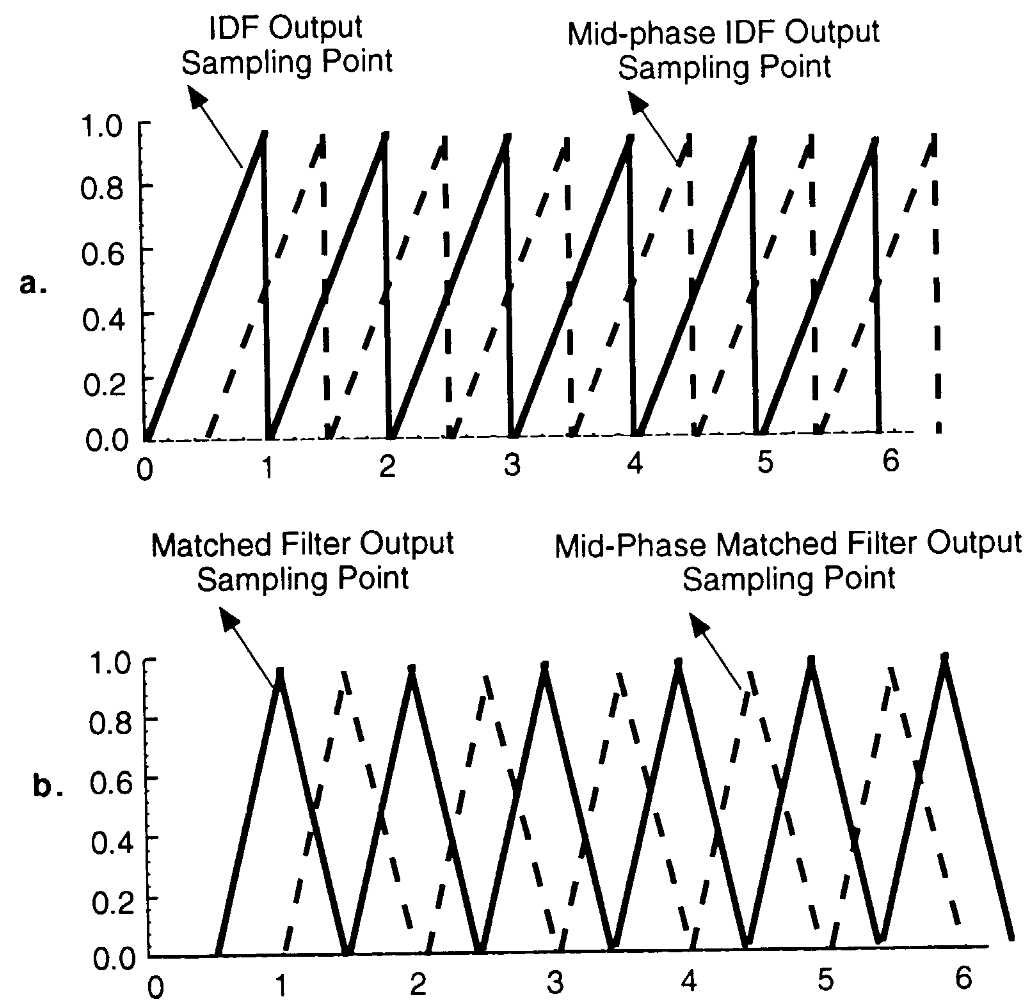

Figure 46. Comparison of Sampling Points: a. Output of IDF and Midphase IDF Output, b. Output of Matched and Mid-Phase Matched Filter

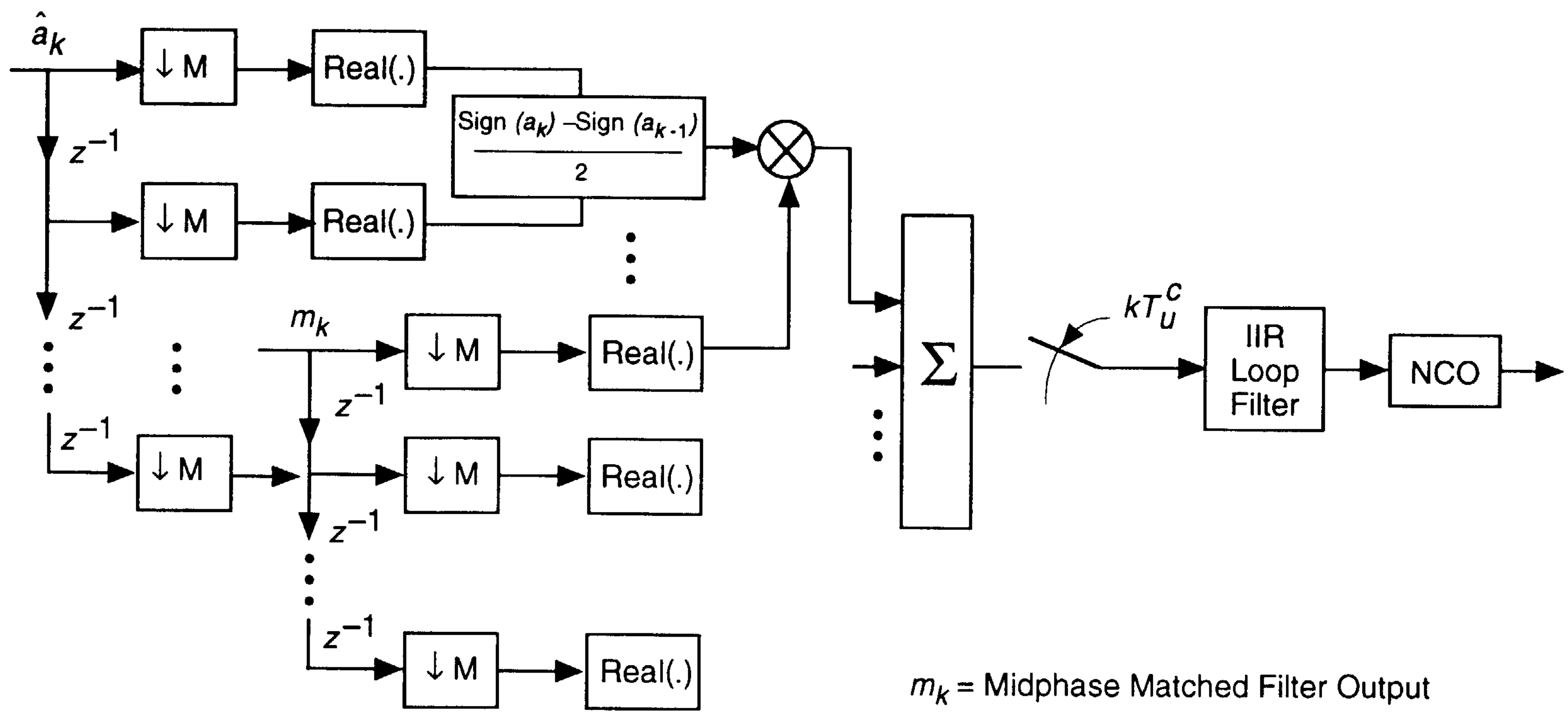

Figure 47. Parallelized DTTL

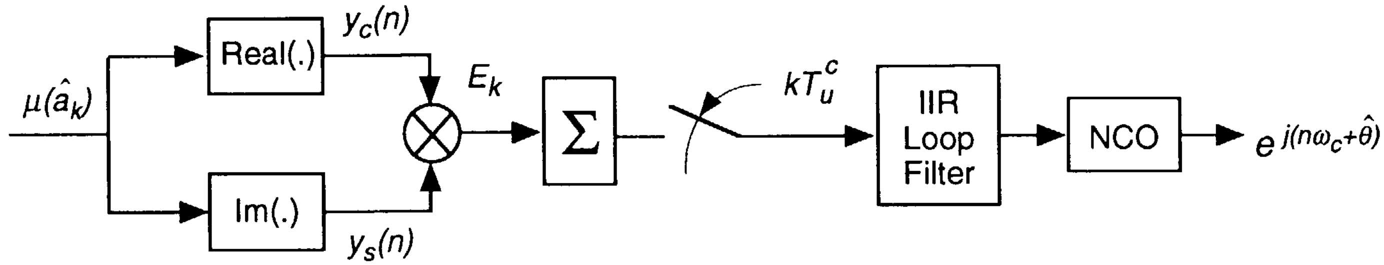

Figure 48. Classical Costas Loop

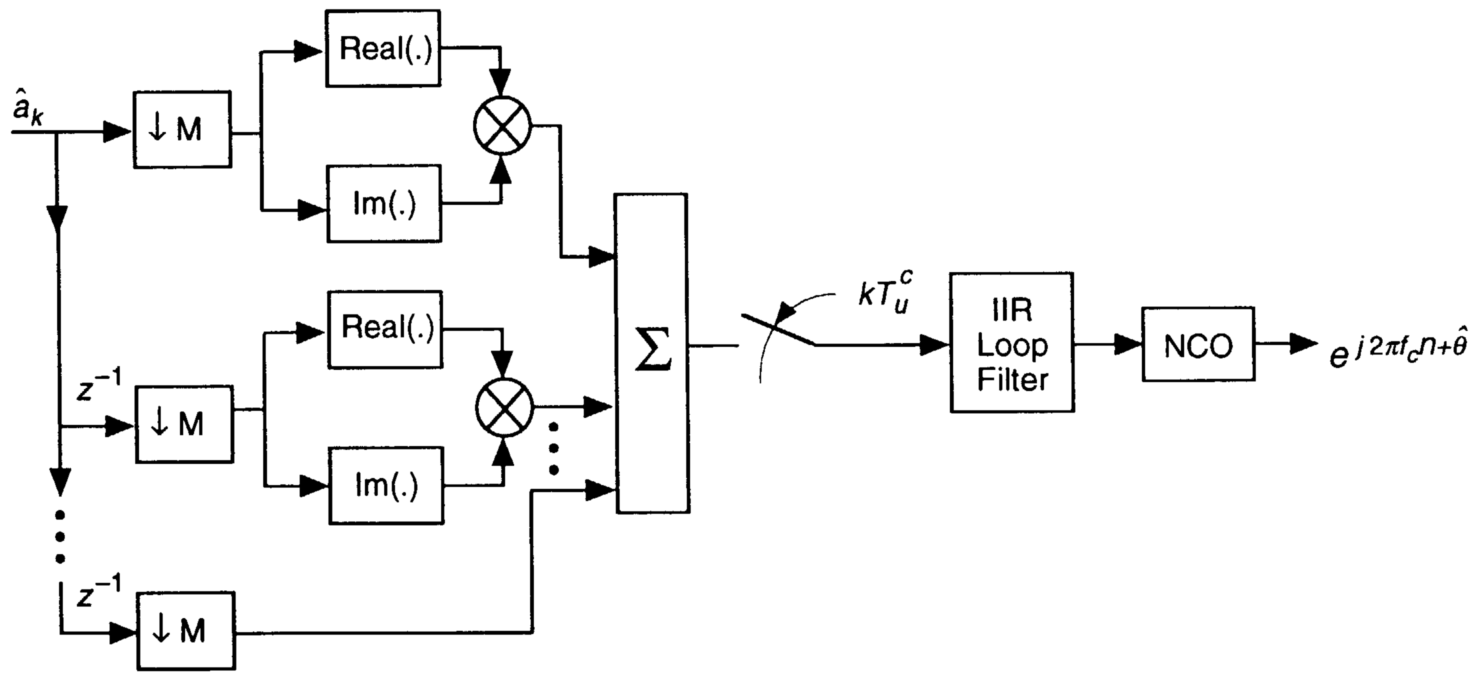

Figure 49. Parallelized Costas Loop

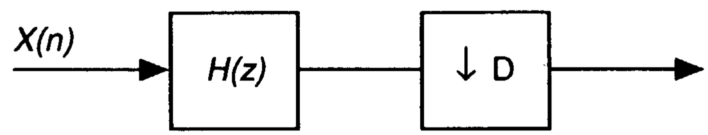

Figure 50. Basic Decimation and Filtering

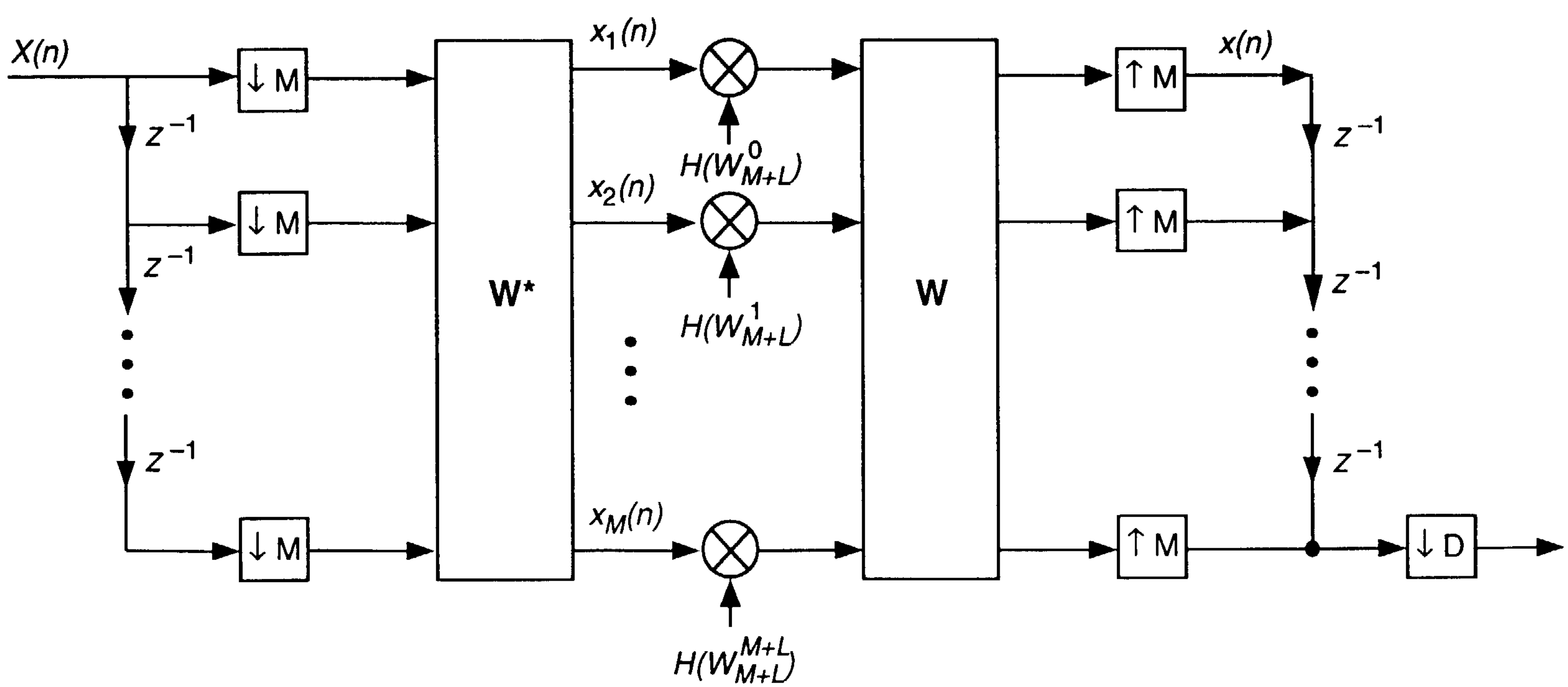

Figure 51. Option II: Frequency Domain Calculation

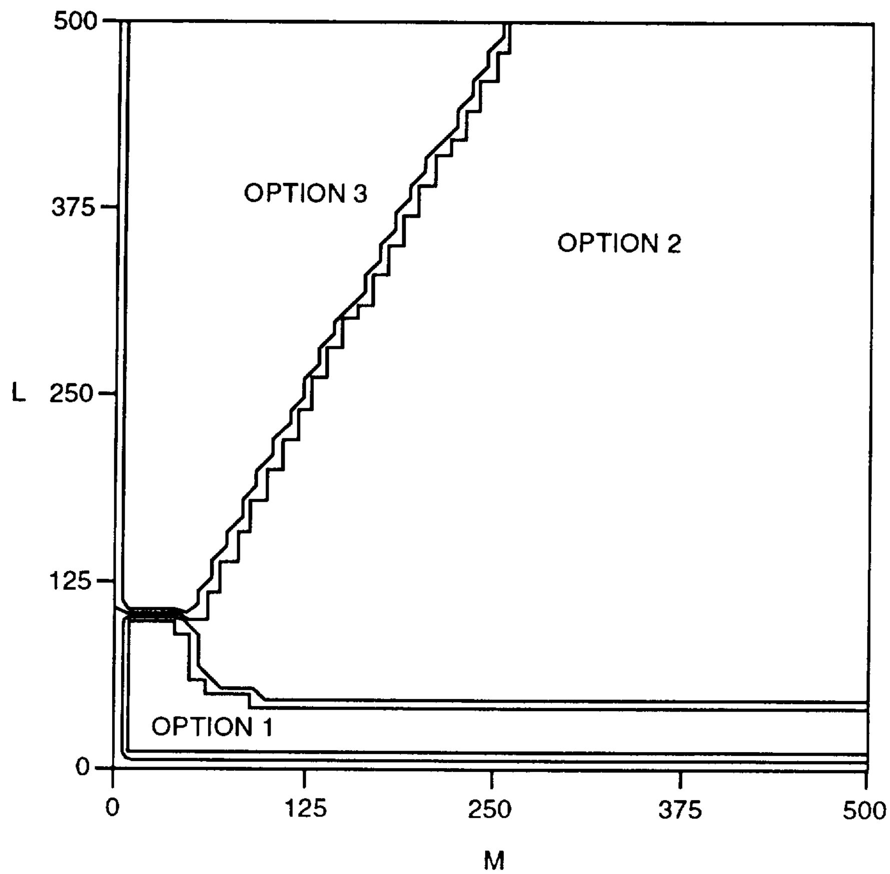

Figure 52. Complexity of Various Options for parallelization of Filtering Operation versus L (Filter Order) and M (Number of Banks)

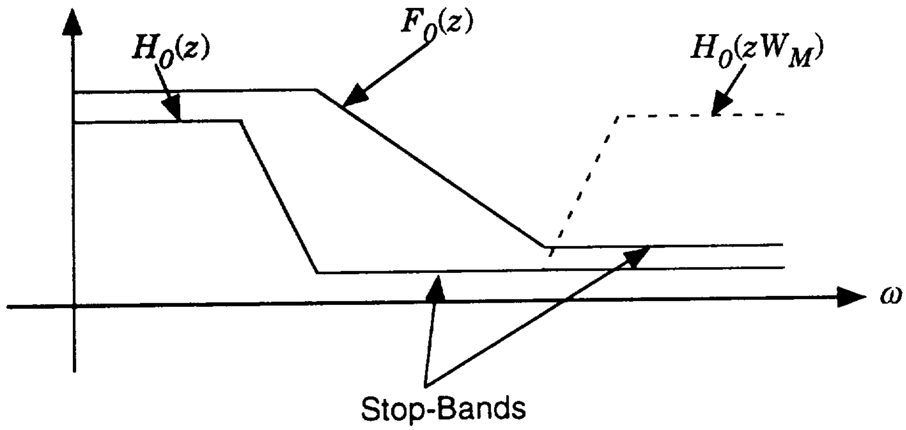

Figure 53. Filter Design Specifications

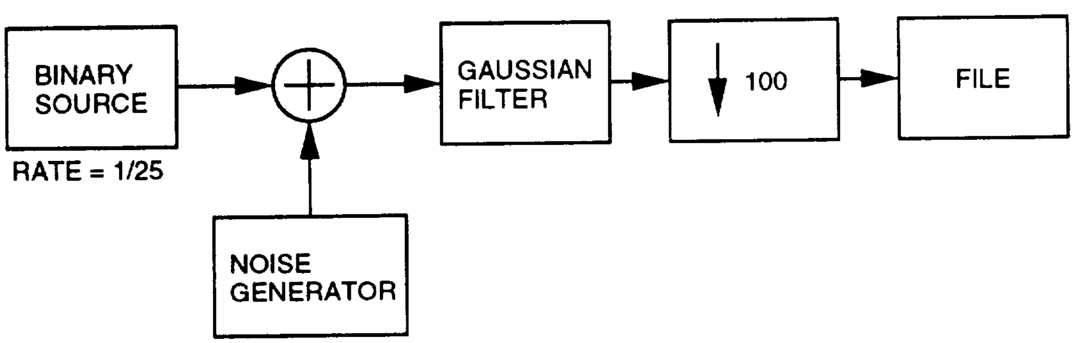

Figure 54. Input Generation For Simulation

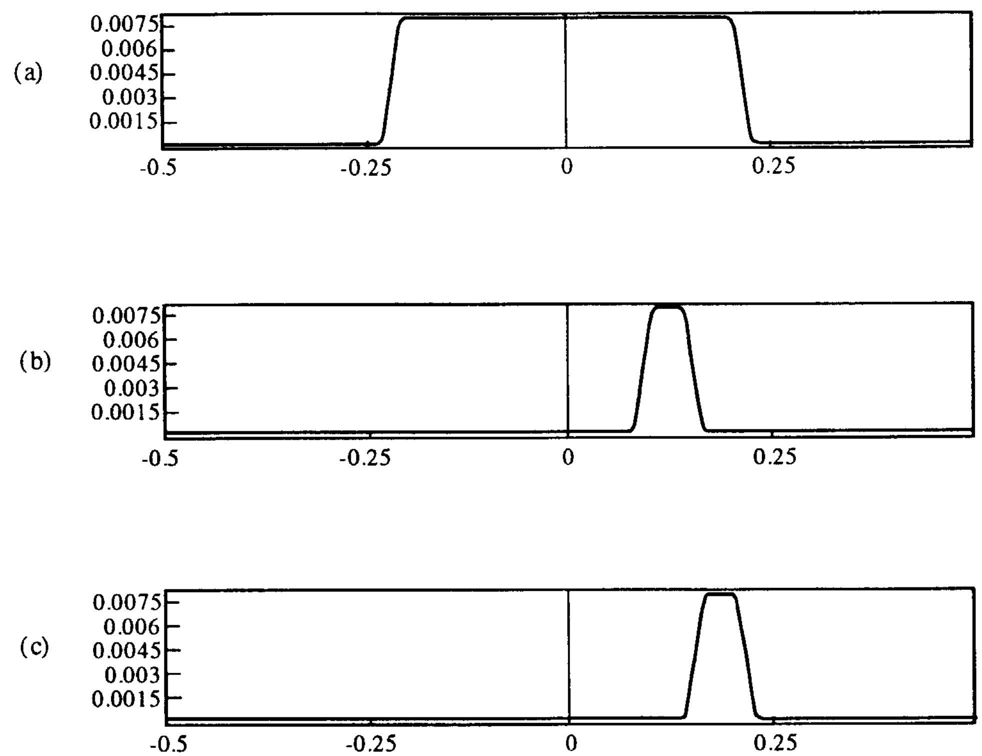

Figure 55. Composition of a Frequency Band Out of Sub-bands: a. 7 Sub-bands Create One Continuous Frequency Band. b., c. Individual Sub-bands.

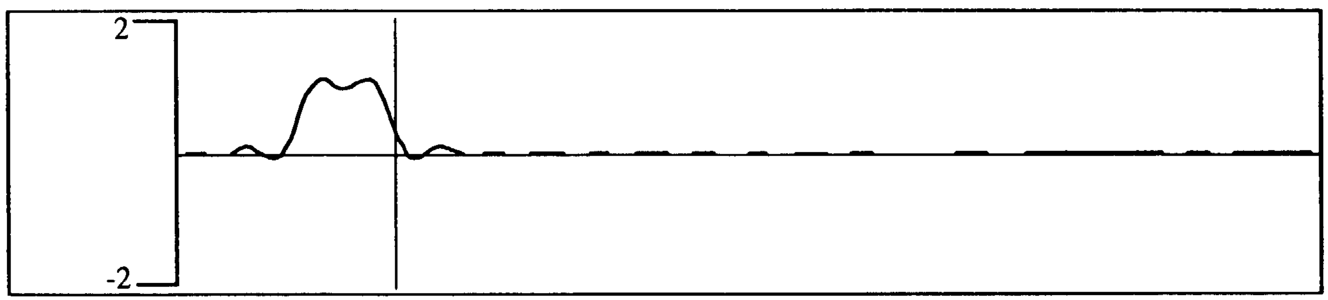

Figure 56. Integrate & Dump Impulse Response, with 7 Lower Frequency Sub-bands Connected

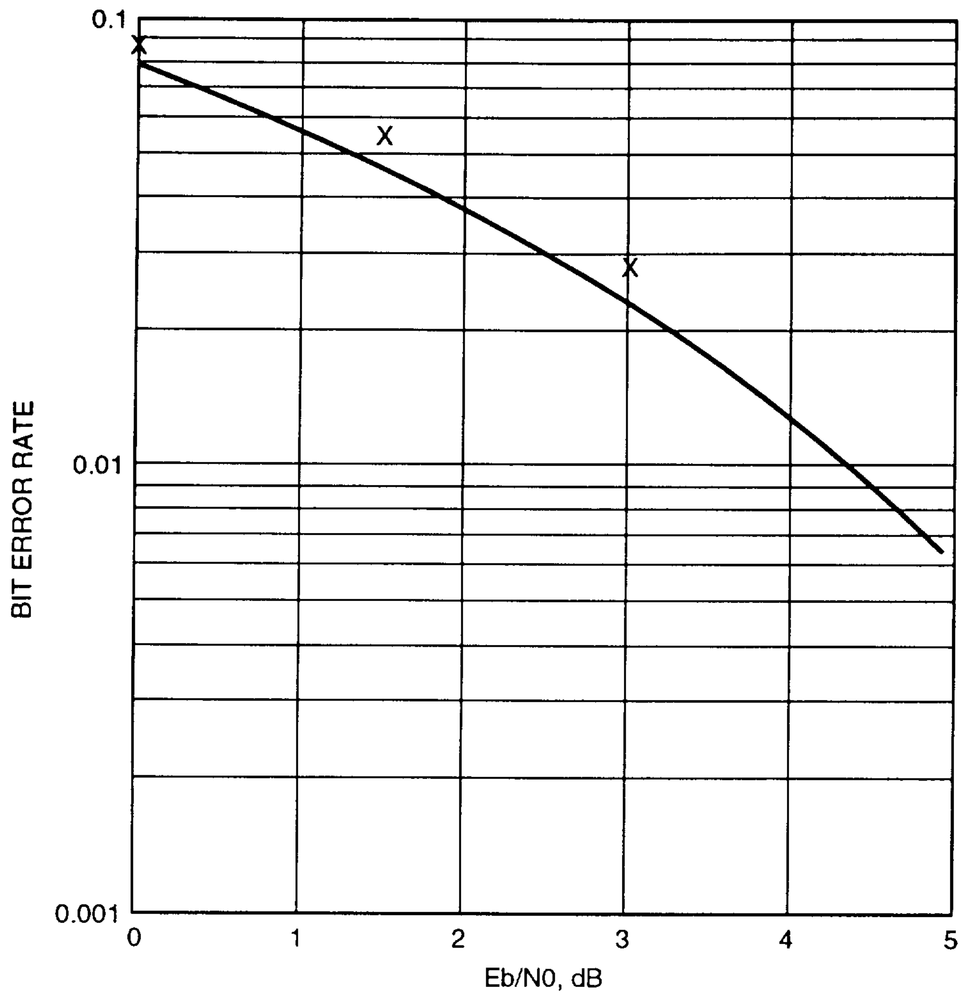

Figure 57. BER Performance of Filter-bank Receiver With 4 Samples Per Symbol and IF Sampling

Appendix A. MathCad™ Software for Generation of Filter Bank Coefficients

Appendix A: Filter Bank Receiver -- Generation and Testing.

A.1 Filter Design.

$B := 16$ Number of filters in the filter bank

$M := 8$ Decimation ratio (B=2M)

Analysis filter design:

$N_h := B \cdot 6 - 1$ Length of analysis filter. 6 can be replaced by any even number, else fix the delays.

$\mathrm{setwindow}(5) = 5$ Set the window to Hamming

$h := \mathrm{lowpass}\left(\frac{1}{2 \cdot B}, N_h\right)$ This makes a B-band filter. The function lowpass generates a FIR by the windowing method.

$i := 1 .. \mathrm{length}(h)$

$k := 0 .. B - 1$

$N_h = 95$

$h0_i := h_{i-1}$ Position the center of the impulse response at a multiple of B, for obtaining a delay which is a multiple of B.

$h0_0 := 0$

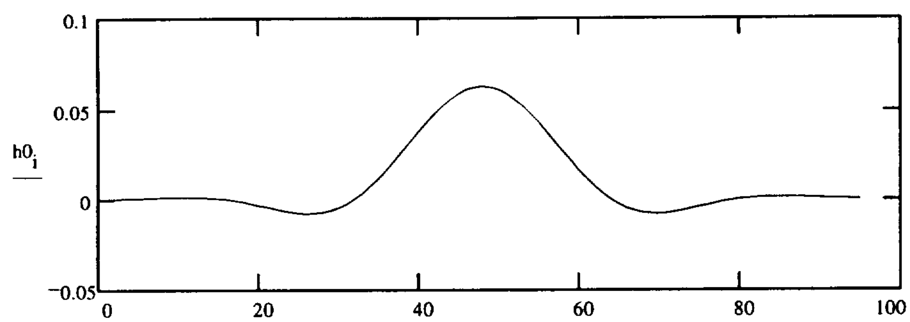

Figure A.1: Analysis filter H0(z) impulse response.

$n := 0 .. 99$ $f_n := \frac{n}{100}$

$W_B := \exp\left[j \cdot \left(\frac{2 \cdot \pi}{B}\right)\right]$

$H0(z) := \sum_i z^{-i} \cdot h0_i$ $H(k, z) := H0\left(z \cdot W_B^{-k}\right)$

$\alpha := 1.45 + 0.52j$ verifying that the filters sum to $z^{-3 \cdot B}$ so they are indeed Nyquist filters. α is an arbitrary number.

$\sum_k H(k, \alpha) = -6.71718 \cdot 10^{-10} + 7.19988 \cdot 10^{-10} i$ $\alpha^{-3 \cdot B} = -6.71718 \cdot 10^{-10} + 7.19988 \cdot 10^{-10} i$

Synthesis filter design:

$\mathrm{setwindow}(5) = 5$

$N_f := B \cdot 4 - 1$ Length of synthesis filter, same remarks as above.

$ff := \mathrm{lowpass}\left(\frac{1.9}{2 \cdot B}, N_f\right)$

$N_f = 63$ $j := 1 .. \mathrm{length}(ff)$

$f0_j := ff_{j-1}$ Put the middle at a multiple of B, for having a delay which is a multiple of B.

$f0_0 := 0$

$F0(z) := \sum_j z^{-j} \cdot f0_j$ $F(k, z) := F0\left(z \cdot W_B^{-k}\right)$

Save prototypes in file for SPW simulation:

WRITEPRN(f) := ff
WRITEPRN(h) := h0

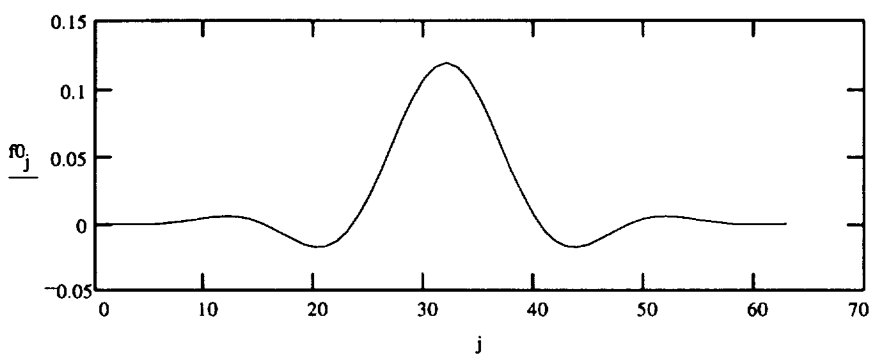

Figure A.2: Synthesis filter F0(z) impulse response.

Generating Gk(z):

$i := 0 .. 7$

$x_i := 1$ x is the desired bank response. In this case x is an integrate&dump of 8 samples.

$A_{k,j} := f0_j \cdot W_B^{-j \cdot k}$ Generate all Fk(z) coefficients

$\mathrm{lng} := \mathrm{length}(f0) + \mathrm{length}(x) - 1$

Convolve Fk(z) with x.

$go^{<k>} := \mathrm{response}\left[x, \mathrm{Re}\left[\left(A^T\right)^{<k>}\right], \mathrm{lng}\right] + j \cdot \mathrm{response}\left[x, \mathrm{Im}\left[\left(A^T\right)^{<k>}\right], \mathrm{lng}\right]$

$m := 0 .. \frac{lng}{M}$

$gd_{m,k} := go_{m \cdot M,k}$ Decimate the result.

$WRITEPRN(gi) := Im(gd)$ Write to file.

$WRITEPRN(gr) := Re(gd)$

$z_n := \exp(j \cdot 2 \cdot \pi \cdot f_n)$

A.2 Frequency domain tests.

The frequency response of several analysis filters and one synthesis filter is shown in Figure A.3.

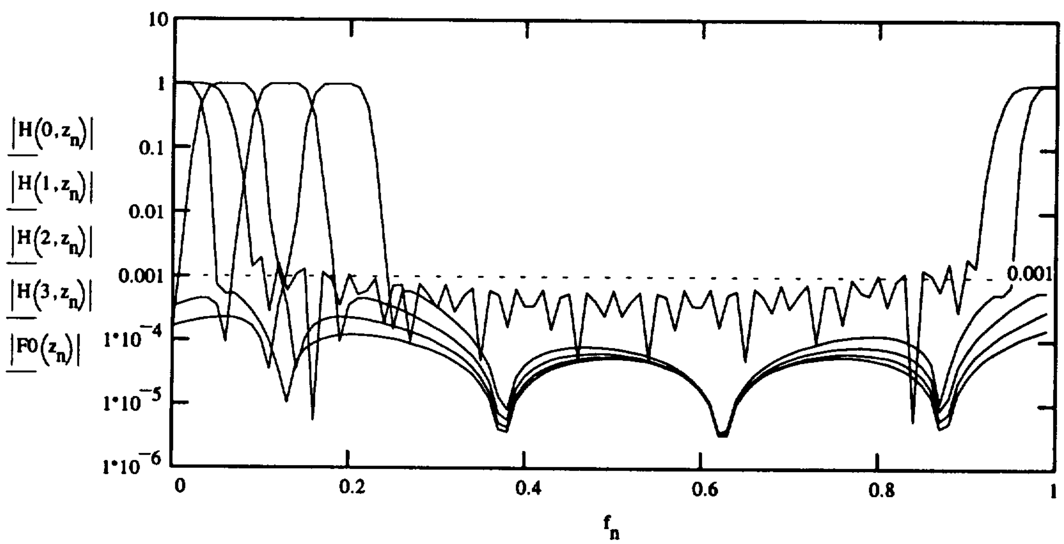

Figure A.3: Frequency response of H0(z)...H3(z) and of F0(z)

A test that the combined filters Hk(z)*Fk(z) do sum to a constant (the filters Fk(z) distort the Nyquist property of the filters Hk(z)) is shown in Figure A.4. In this figure, we also demonstrate how a frequency band is constructed out of a few subbands.

$V(k,z) := F(k,z) \cdot H(k,z)$

$p := 0 .. 4$

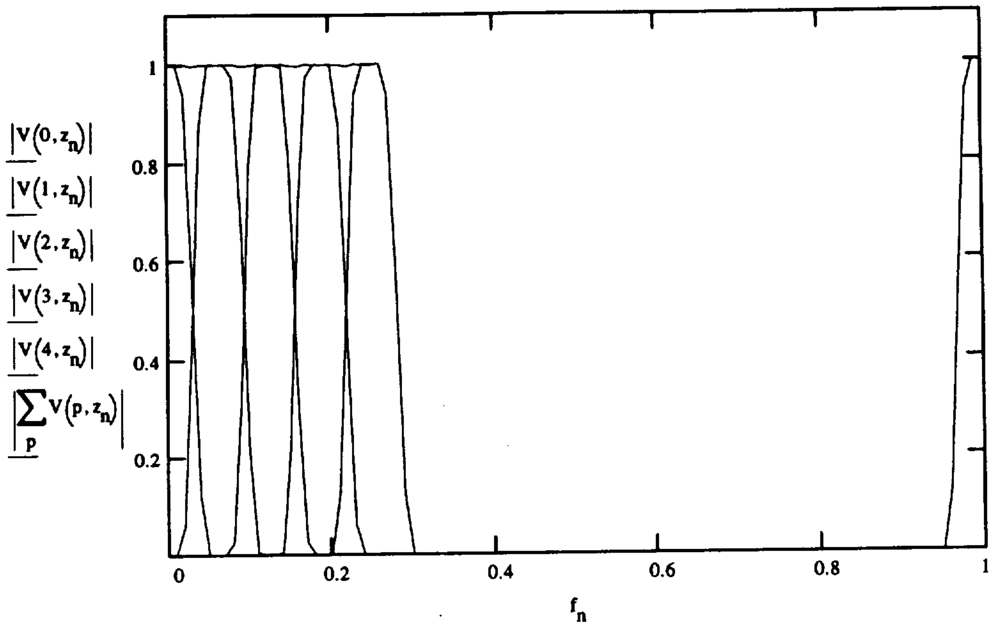

Figure A.4: Frequency band composition from several subbands.

Adding decimation-interpolation to the picture.

$W_M := \exp\left(i \cdot \frac{2 \cdot \pi}{M}\right)$ $\qquad$ $m := 0 .. M - 1$

Computing the frequency response of Hk(z) and Fk(z) followed by decimation by M and then interpolation by M.

$HH(k, z) := \sum_m H\left(k, z \cdot W_M{}^m\right)$ $\qquad$ $FF(k, z) := \sum_m F\left(k, z \cdot W_M{}^m\right)$

$HHFF(k, z) := HH(k, z) \cdot FF(k, z)$ $\qquad$ The combined response with all the images.

$HHFF\_F(k, z) := HH(k, z) \cdot FF(k, z) \cdot F(k, z)$ $\qquad$ The filter Fk(z) removes all the undesired images.

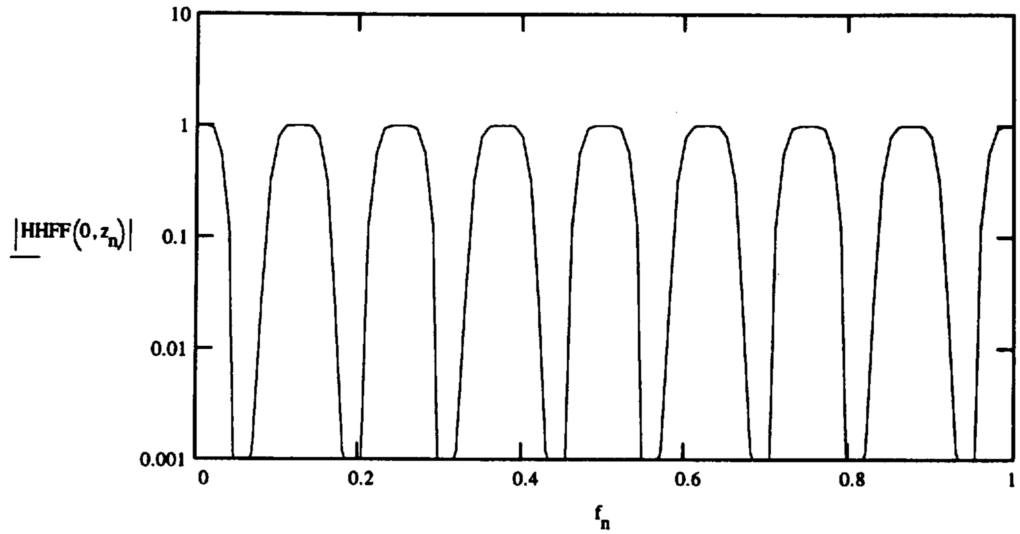

Figure A.5: Frequency response of a subband after decimation-interpolation by M.

The response of the complete system. The system is supposed to compute the convolution of X and Q, both may be arbitrary signals. In reality X is the input to the receiver and Q is the matched filter response. We set here F`k(z)=Fk(z).

Arbitrary choice of X and Q

$Q(z) := 3 - z^{-1} + 5\cdot z^{-2} + 8j\cdot z^{-3}$

$X(z) := 1 + z^{-1} - 3\cdot z^{-4}$

$HX(k,z) := H(k,z)\cdot X(z)$ X passed through each one of Hk(z).

$FQ(k,z) := F(k,z)\cdot Q(z)$ Q passed through each one of Fk(z).

$HX1(k,z) := \sum_m HX\left(k, z\cdot {W_M}^m\right)$ Decimation-interpolation by M.

$FQ1(k,z) := \sum_m FQ\left(k, z\cdot {W_M}^m\right)$ Decimation-interpolation by M.

$Od(z) := X(z)\cdot Q(z)$ The desired response (convolution of X with Q).

$O3(z) := \sum_k F(k,z)\cdot HX1(k,z)\cdot FQ1(k,z)$ The filter bank output. Here each subband is convolved, and then passed through the synthesis filter Fk(z). The result is summed to provide the output.

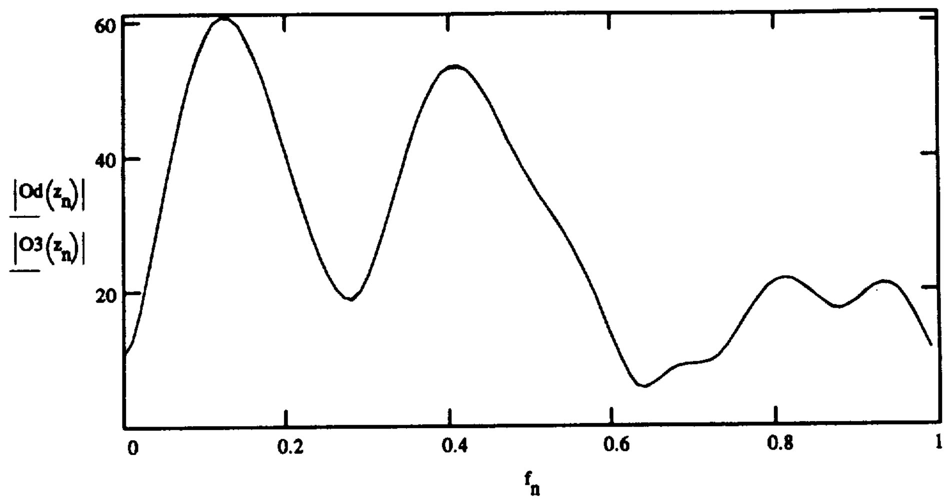

Figure A.6: Comparison of desired output and bank output (the two curves are almost exactly one atop the other).

Appendix B. SPW block diagrams

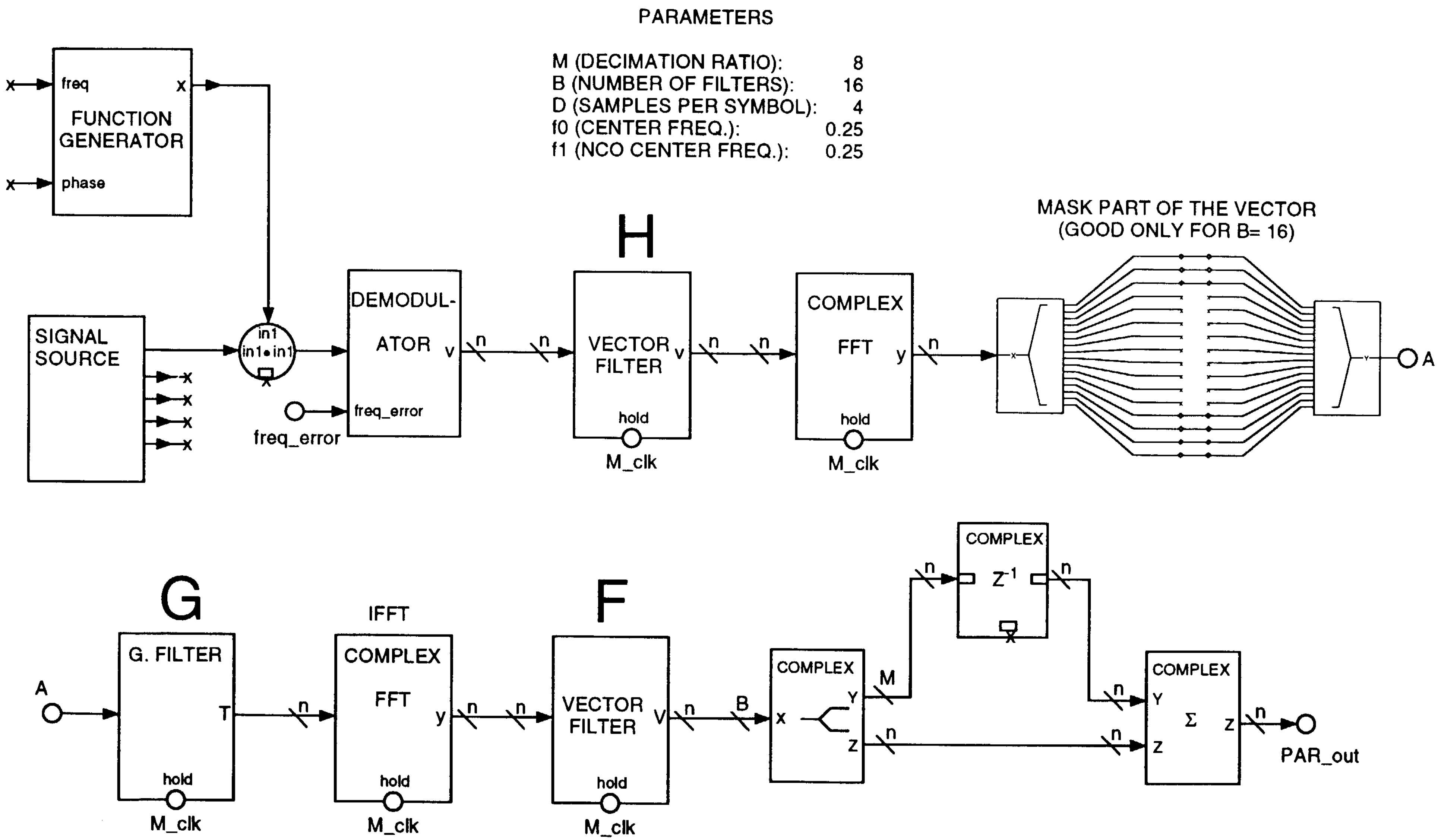

Figure B-1. Filter Bank Receiver Block Diagram in SPW

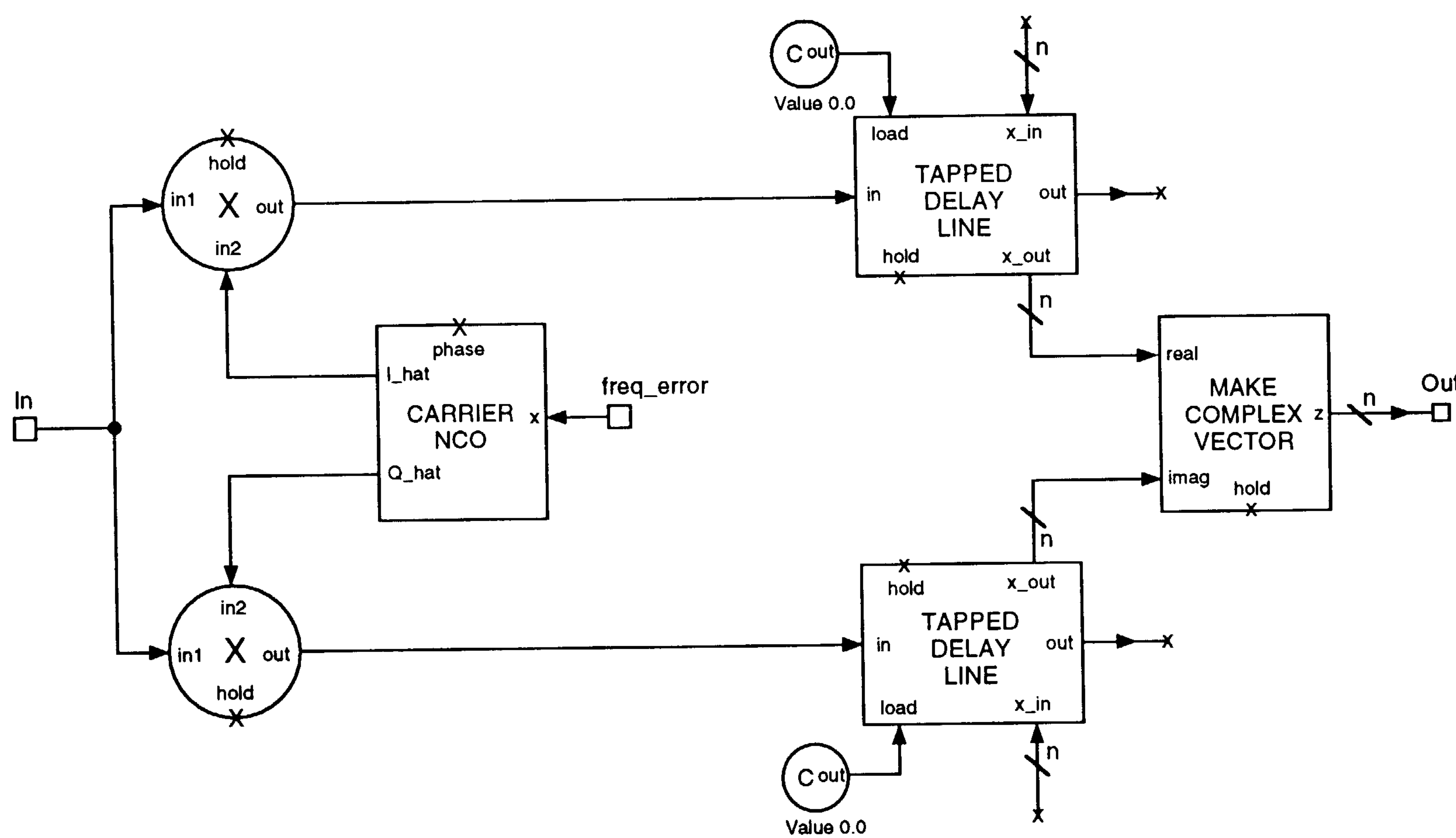

Figure B-2. Parallel Demodulator Block

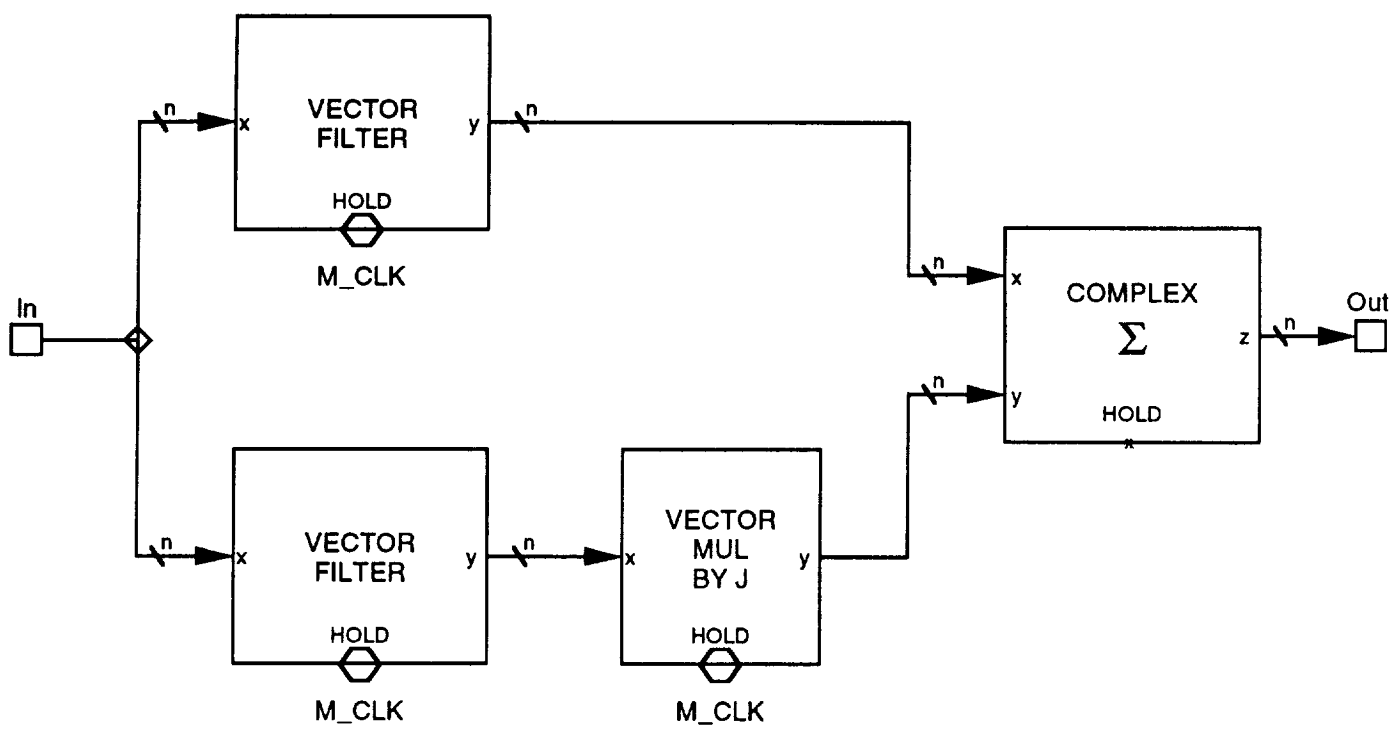

Figure B-3. Complex Vector Filter Block

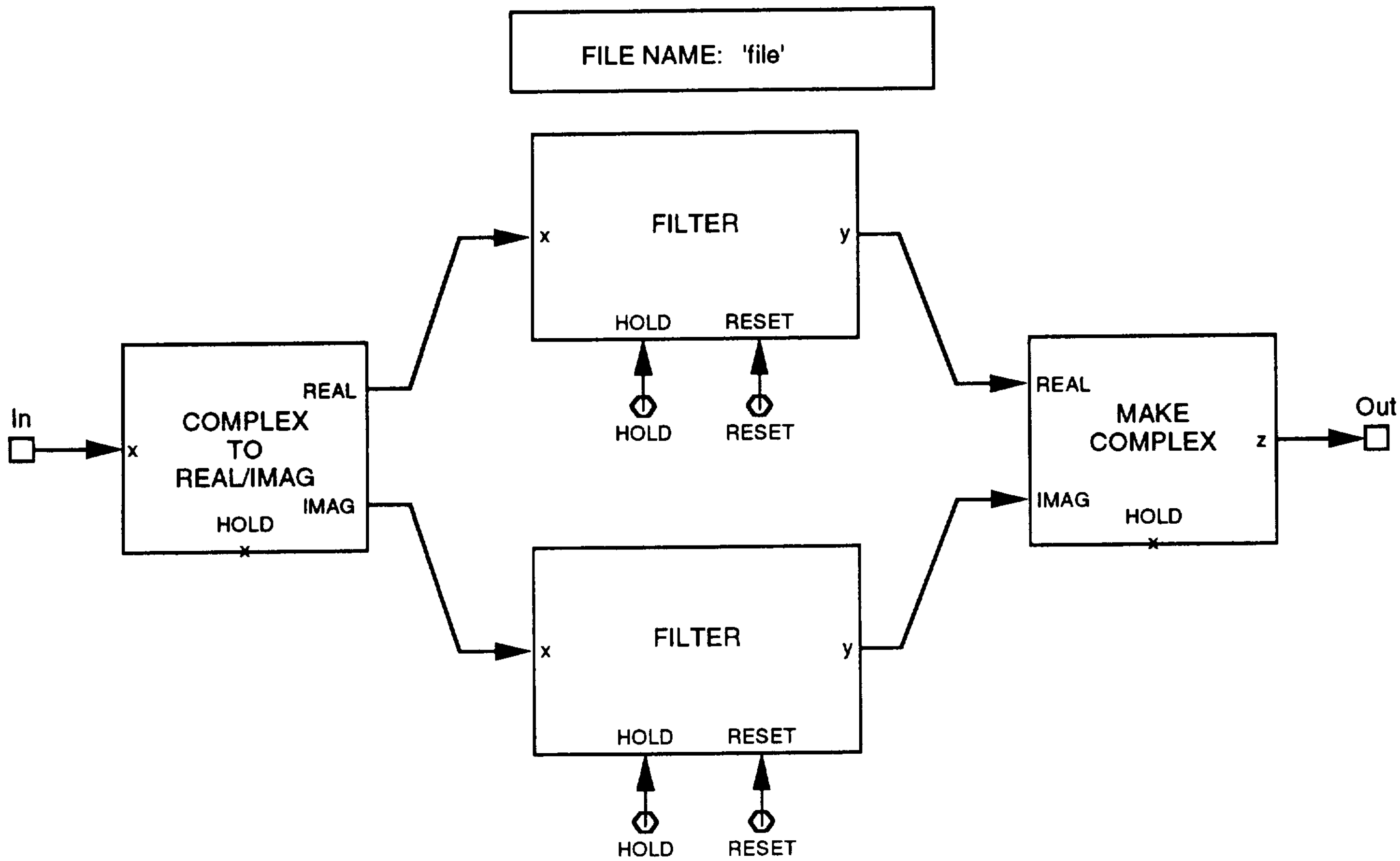

Figure B-4. Complex FIR Filter Block

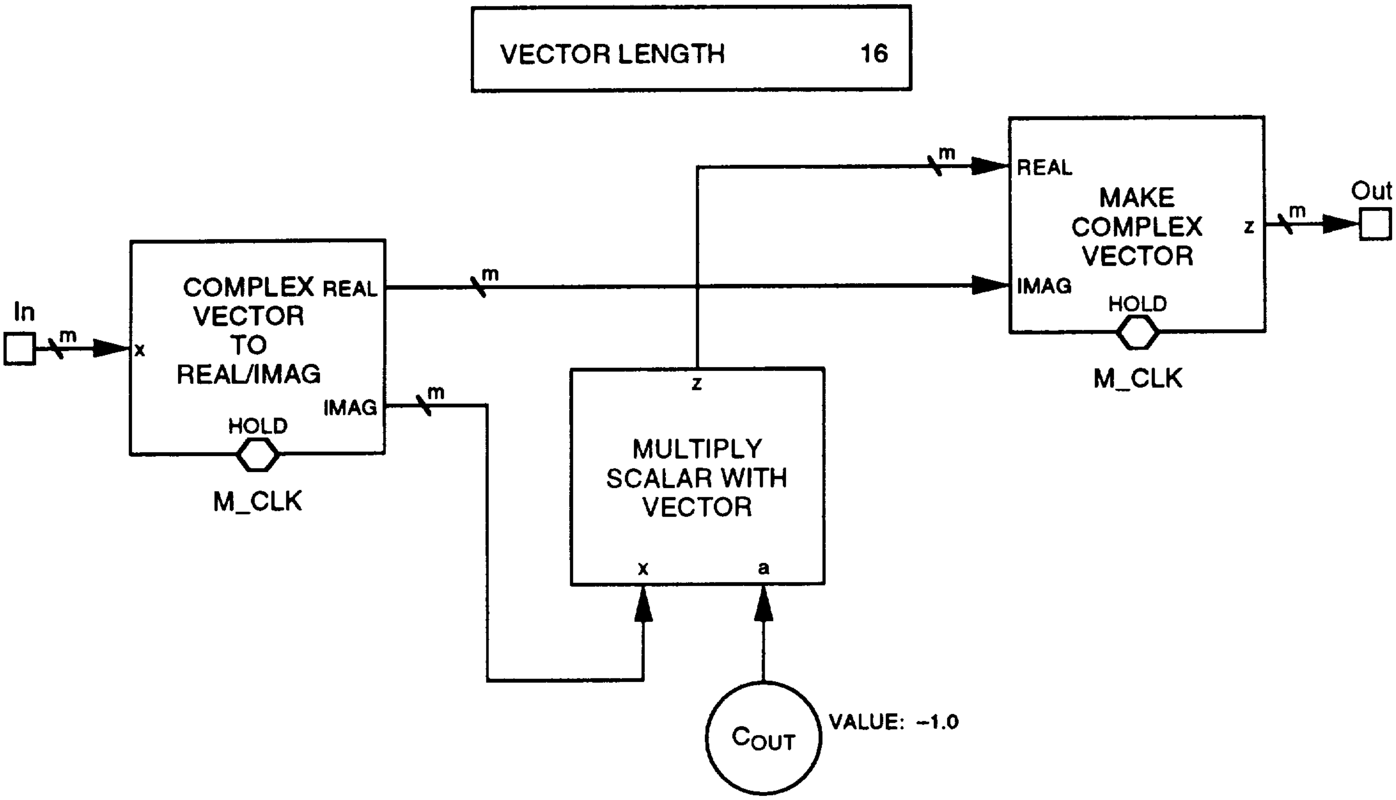

Figure B-5. Vector Multiply by $\sqrt{-1}$ Block

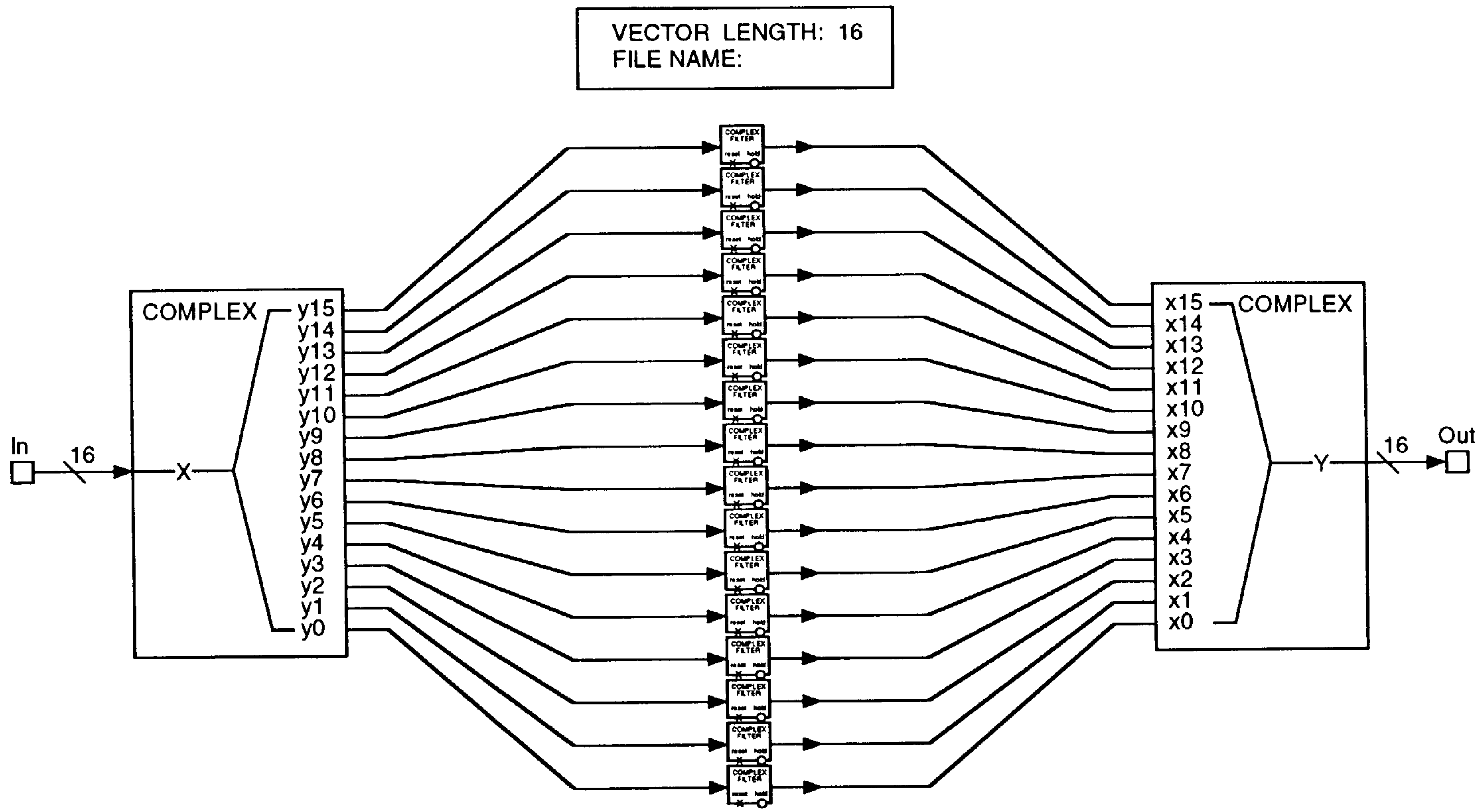

Figure B-6. Vector Filter Block

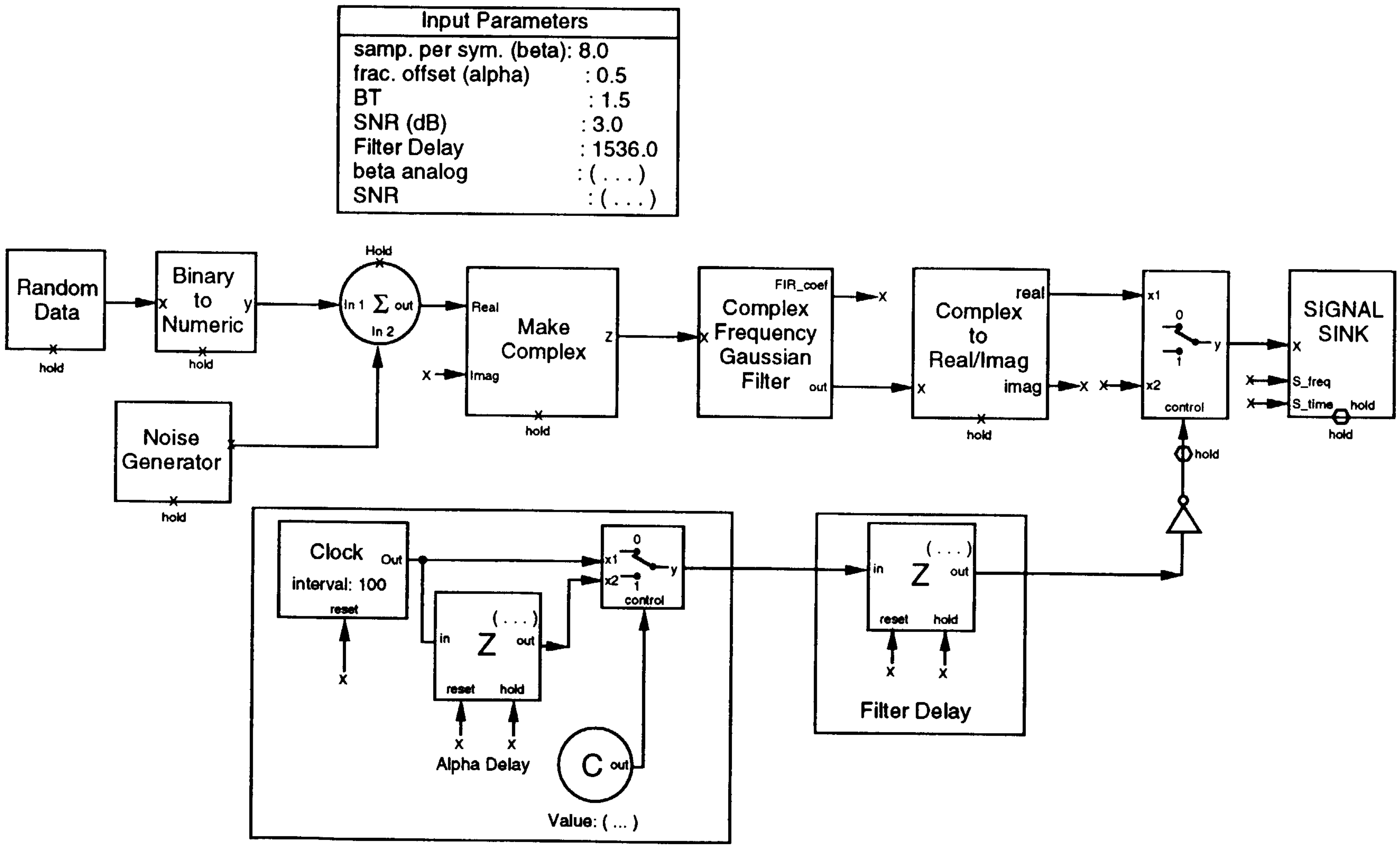

Figure B-7. Input Generation SPW System Block Diagram

Appendix C. Receiver Block Diagram

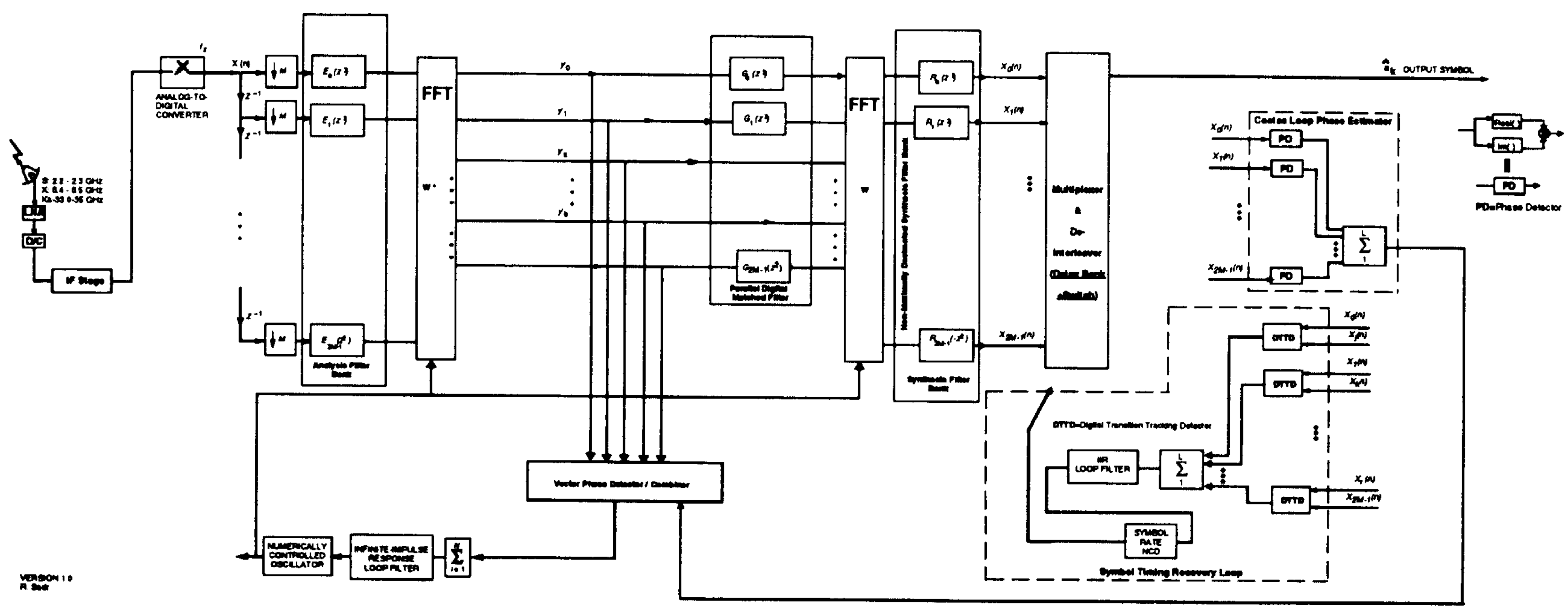

Figure C-1. PRX Block Diagram

Appendix D. C-Programs for Designing PRX Filter Banks

Appendix D.1 Generating the Polyphase filters

*I. '**poly_gen.c**', C program for generating the polyphase filters' files for SPW from the ASCII files of the coefficients of H0(z) or F0(z).*

Appendix D.2 Generating the Subband Matched filters

*II. '**fil_arr.c**', C program for generating the Gk(z) filters' files for SPW from one ASCII file .*

Appendix D.3 Designing the IFIR Matched Filter

*III . '**ifir.c**', C program for Designing IFIR Filters*

poly_gen.c

```
/* poly_gen.c: Generated spw filter files of the polyphase components of a givenfilter.
The input filter coefficients are given in plain ascii format. For a
non-maximally decimated filter bank, the polyphase filters are interpolated by
the ratio of the number of banks to the decimation ratio (must be an integer). The defau
lt is 2, i.e., the number of banks is twice the decimation ratio, and the polyphase filt
ers are interpolated by 2. */



#include "stdio.h"
#include "stdlib.h"

void main(argc,argv)
int argc;
char ** argv;
{
int i,j,k,B,len;
char name[80];
char ss[100];
double b[1000];
int G,M,g,O,o;
FILE *filter_file,*out_file;

if (argc<4) {
  printf("\nEnter file name (no extention please): "); scanf("%s",name);
  printf("\nEnter number of filters: "); scanf("%d",&B);
  printf("\nEnter polyphase type (1 or 2): "); scanf("%d",&O);
  }
else {
  strcpy(name,argv[1]);
  B=atoi(argv[2]);
  O=atoi(argv[3]);
  }
if (argc<5) {
/*   printf("\nEnter decimation ratio: "); scanf("%d",&M);*/
   M=B/2;
   }
else {
  M=atoi(argv[3]);
  }
G=B/M;
printf("Decimation rate for bank is %d\n",M);
if (G*M!=B) {
  printf("Decimation rate must be a dividor of number of filters\n");
  exit(0);
  }
filter_file=fopen(name,"r");
if (filter_file==NULL) {printf("File not found\n"); exit(0);}
len=0;
while (!feof(filter_file)) {
  fscanf(filter_file,"%le",&b[len++]);
  if (len>=1000) {
    printf("Filter too long");
    exit(0);
    }
  }

if (b[len-1]==0) len--;

printf("Length of original : %d\n",len);
for (i=0;i<B;i++) {
  o=(O==1?i:B-1-i);
  sprintf(ss,"%s%d.ascfilt",name,o);
  out_file=fopen(ss,"w");
  if (filter_file==NULL) {printf("Cannot create file %s\n",ss); exit(0);}
  if (G>1) fprintf(out_file,"FILTER 1.0\nMethod: Polyphase decomposition of given FIR for
nonmaximally decimated filter bank.\n");
  else fprintf(out_file,"FILTER 1.0\nMethod: Polyphase decomposition of given FIR.\n");
  fprintf(out_file,"Comment: Filter No %d out of 0-%d\n",o,B-1);
  fprintf(out_file,"$\nPARAMETERS\n$\nCOEFFICIENTS\nPoly filter\n");
  fprintf(out_file,"%d numerator coefficients (b)\n",((len-i-1)/B+1)*G-1);
  fprintf(out_file,"1 denominator coefficients (a)\n");

  for (j=i,k=0;j<len;j+=B) {
    fprintf(out_file,"b[%d] = %le\n",k++,b[j]);
    if (j+B<len) for (g=1;g<G;g++)  fprintf(out_file,"b[%d] = 0.\n",k++);
    }
  if (i>=len) fprintf(out_file,"b[0] = 0.\n"); /* no filter */
  fprintf(out_file,"a[0] = 1.\n$\n");
  fclose(out_file);
  }
fclose(filter_file);
}
```