Journal of Engineering Science and Technology
Vol. 1, No. 2 (2006) 176- 191

ANALYSIS OF EFFECT OF APPROXIMATING OUTPUT SAMPLES USING RANDOM SUB-SAMPLING OF INPUT FOR COMPUTATION REDUCTION IN FILTERING OPERATION

AMANPREET SINGH [1*], SANJAY MARWAHA [1], ANUPMA MARWAHA [2]

[1] Dept. of Electrical and Instrumentation Engineering,
[2] Dept. of Electronics and Communication Engineering BHSBIET
Sant Longowal Institute of Engineering and Technology, Longowal (Sangrur)
148106 INDIA
*Corresponding Author: amanpsliet@yahoo.com

Abstract

Communication plays a significant role in today's life. Integration of computing and communicating devices, wide-spread internet access through World Wide Web (WWW), and wireless links are an increasing demand for mobile cellular services at the consumer end, so it has led to new signal processing technologies. Signal processing and communications are tightly inter-woven and immensely influence each other. As the need for sophisticated signal processing algorithms and hardware increase, their potential to make contributions to the communication revolution appears unbounded. Digital signal processing (DSP) technology is widely used in numerous familiar products, peripherals of computers and the electronics world. This paper deals with the optimization of DSP environment for communication applications. Emphasis is given to the receiver part of the communication system; more specifically the channel separation aspect is discussed. No such algorithm for the computational saving for receiver part of communication system has been reported earlier. In this paper an attempt has been made to optimize the filtering operation.

Keywords: FIR Filter, Computation reduction, Frequency Translation, Auto Corelation, Cross Corelation.

Nomenclatures

f_{clk}	Clock frequency
C_L	Equivalent Load Capacitance
V_{dd}	Power Supply Voltage
S_n	Sum of Indexed sequence
I	Interfering Signals
n	Integer number
P, q	Prime number (i.e for Prime Code)
i	Sequence numbers
W	Code weight
Z	Elements in Hadamard Matrix
n_n	Random Noise Component
r_n	Input to digital receiver
h	Coefficient of Filter
M	Order of the Filter
D	Decimation Factor
Greek Symbols	
Ω	Frequency domain parameter
λ	Cross- correlation
λ_a	Auto- correlation
λ_c	Cross- correlation

1. Introduction

The advent of portable computing has led to a significant increase in research work targeting the reduction of power consumption in high throughput digital signal processor (DSP) devices. The power dissipation in digital CMOS circuits is the sum of three contributions i.e. The static power dissipation P_s, the short-circuit power dissipation P_{sc}, and the dynamic dissipation P_{dyn}. The dominating term in the DSP applications is the dynamic power dissipation, P_{dyn}, which is due to the charging of the capacitive load C_L to voltage level V_{dd} through the CMOS transistor. The rate for which the capacitive load is charged, is given by the clock frequency f_{clk} and the switching activity factor (the average number of 0 – 1 transitions per clock period) [1-3].

The dynamic power consumption of a CMOS circuit is the dominating part of the total power consumption. This part can be approximated by the well known formula (Eq. 1)

$$P_{dyn} = 1/2 \alpha f_{clk} C_L V_{dd}^2 \tag{1}$$

where α is the switching activity of the circuit, f_{clk} is the clock frequency of the circuit, C_L is the equivalent load capacitance of the circuit, and V_{dd} is the power supply voltage.

Therefore, for achieving low power in CMOS circuits one must target to minimize one or more of the parameters α, V_{dd}, C_L and f_{clk}.
Researchers have targeted reducing the switching activity within DSP systems. This includes:

1. The use of coding techniques
2. The design and manipulation of arithmetic units such as adders and multipliers
3. The use of various techniques to exploit bit correlation in coefficients.
4. A low-power technology decomposition procedure, which produces a decomposed network with minimum switching activities [2].

Other techniques include dynamic minimization of filter orders following a differential approach to process coefficients and use of multirate architectures.

But in all these techniques only coefficients are reordered or hamming distance of coefficients are minimized. No such technique has been stated in which the effect of approximating the output samples through the use of sampling of the input for the minimization of computation is used. This technique is used for minimization of computation, which will further lead to energy optimization in the DSP domain.

A communication device does its entire digital signal processing in user level software on a general-purpose workstation. Wireless networks are statically specified by their built-in link and physical layer functions. Conventional communication devices effectively use improved analog to digital converters and digital signal processing on a dedicated hardware i.e. network interface card (NIC). Implementing all the links and most of the physical layer functions in user level software, increase the flexibility and make it possible to dynamically modify the functions such as modulation techniques and multiple access which are otherwise fixed in traditional NICs. The signal processing involved in these layers has been lumped into one layer because of its implementation in dedicated hardware. But to interpolate with different network, it is necessary to change small parts of the existing layers. For instance, two different computers may employ the same modulation and coding but at the same time they may use different multiple access protocols. To facilitate this flexibility and modularity, maximum possible processing is brought under software control. New network interface can now be created by changing only a small amount of code [4-5, 7-10].

In any communication system, receiver recovers the original data bit stream. After channel decoding and source decoding, the original source information is reproduced. Figure 1 shows the processing steps performed in the receiver. This figure also shows the decomposition of channel decoder into several specific functions. The first two functions are channel separation and symbol detection. The combined task of these two functions is to reproduce the error-encoded bit stream passed to the digital modulator in the transmitter. Of course, the sequence of bits produced at the output of the detector may contain errors. As the original encoded sequence contains some controlled amount of redundancy, the error decoder can detect the presence of errors and correct them to take other appropriate action to handle the corrupted data [12].

In this work, initially the process is examined for the required functions of channel separation and symbol detection. Key objective is to develop a more fundamental understanding of the functionality required by these two processing steps in the digital domain. This understanding can be used to develop algorithms that provide both flexibility and improved efficiency over conventional techniques in a wide range of situations.

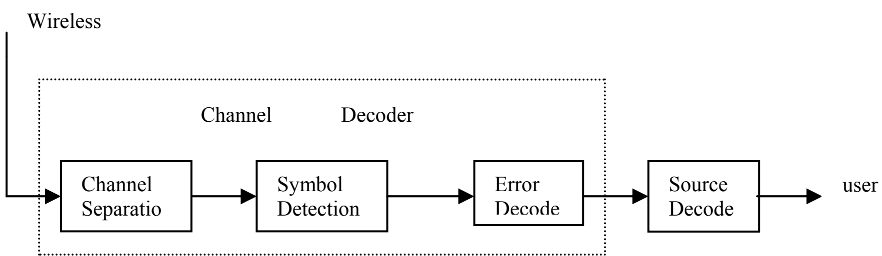

Fig.1. Required Processing Steps in a Digital Receiver

The channel decoder is conventionally divided into specific steps of channel separation and detection because the wireless channel is a shared medium that combines many signals. The distortion caused by the channel takes on several forms: additive interfering signals, random noise, delayed versions of the same signal, etc. So the input is modeled to a digital receiver as a sum of an indexed sequence S_n, representing the signal of interest, with I interfering signals from other nearby transmitters and a random noise component, n_n

$$r_n = s_n + \sum_{i=0}^{I} s_{i,n} + n_n \tag{2}$$

The overall process of channel decoding requires the recovery of the transmitted data from received signal. Algorithms for this type of analysis are sensitive to interference in the input signal, but often perform well in the presence of only additive random noise. We therefore divide the processing of the received wideband signal into two stages: first, the channel separation stage removes interfering signal, transforming the input sequence into another sequence that is simply a noise version of the original signal, from which the detector produces an estimate of the original data encoded by the digital modulator.

2. Channel Separation - Conventional Approach

To understand the level of computation required separating a narrowband channel in a wideband digital receiver, it is helpful to study the specific operation that are required.

Digital filtering (FIR filters) is advantageous whenever there is large decimation factor [6, 11 and 13] as shown in Fig. 2.

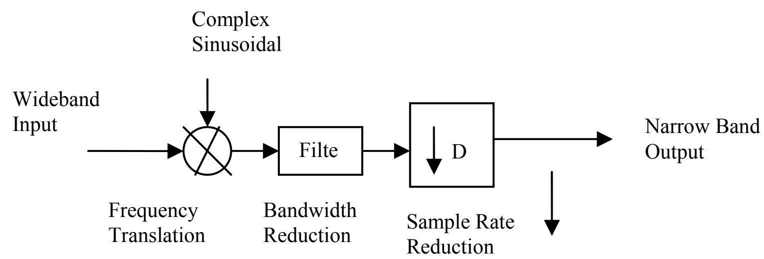

Fig.2 Typical Processing for Narrow Band Channel Selection

In a direct implementation, as shown in Fig. 3, both the frequency translation and filtering steps require computation proportional to R_{in}, which is itself twice the bandwidth of the wideband input measured in Hertz. This minimum sample rate requirement is a consequence of the nyquest sampling theorem which describes the lowest sample rate required to represent the band limited continuous signal using discrete samples without inducing distortion (12-13). In Fig. 3, r_n is the received wideband sample sequence, h_n is the order-M channel filter (with M+1 coefficients), y_n is the filter output and y_n' is the decimated filter output. To perform the translation, we multiply r_n by a complex exponential sequence to get x_n, with the desired signal at complex base band. The output of the cascaded translation and filtering step is

$$y_n = h_m x_{n-m} = h_m r_{n-m} e^{-j2\prod f_c (n-m) T_s} \tag{3}$$

Where $s_n = e^{-j2\prod f_c (n-m) T_s}$ is a complex sinusoidal sequence with frequency f_c (the original carrier frequency) and T_s is the interval between samples.

Cascaded frequency translation and filtering is a very common approach and has been used in many digital receiver implementations [14-15]. It provides the flexibility of modifying the amount of frequency translation without redesigning the entire digital filter. In particular, the technique of BBS is often used to provide precise and flexible frequency translation. One drawback of this approach is that the frequency translation stage requires a complex multiplication to be performed for every sample prior to filtering (in addition to generate the complex sequence s_n). This computational load can become quite high, especially as we consider the design of receivers with wider input bandwidths: the wider input bandwidth requires proportionally higher sample rate and higher computation for frequency translation. For filtering, M+1 multiply-accumulate operations are required for each output samples, where the output rate is $R_{out} = R_{in}/D$. An example of a wideband digital receiver, with typical values for these parameters helps to make these relationships more concrete.

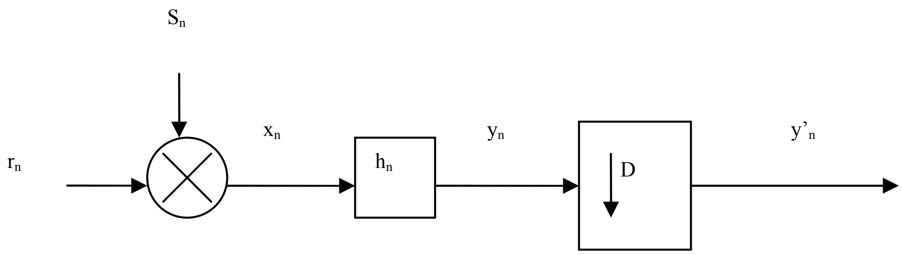

Fig. 3. Block diagram showing frequency translation of desired signal before filtering & decimation

A wideband receiver for a cellular base station needs to access 12.5 MHz of spectrum, so we will use Rin = 30M samples/second. We assume that a narrowband voice channel of 30 KHz would require a filter with about 1800 coefficients and we will use a decimation factor of D = 600 to produce a complex valued output sequence with Rout = 50 Ksamples/second. Using these numbers, the computation require each second to generate the output sequence for a single narrowband voice channel is 50,000X1800 = 90million complex real multiply-accumulate (for filtering), plus at least 30 million complex X real multiplications for frequency translation [16].

Many techniques have been used to make this computationally intensive filtering task more manageable. We now examine a few of these techniques and how each improves efficiency over the conventional approach.

One common technique is to use a cascade of multiple FIR filters that perform the bandwidth reduction in several stages. Using this approach, a lower average number of operations per output sample can be achieved [14,17]. The basis for this improvement is that the each of the multiple stages (except the last) computes an intermediate result that is used in computing multiple output samples in order to amortize the computational costs.

Another approach, the filter bank, is appropriate multiple independent narrowband channels are to be extracted from the same wideband input sequence simultaneously. If each of the desired output channels has identical bandwidths and response (just different center frequencies) then techniques exist that can exploit the special relationship between the multiple sets of filter coefficients. These techniques can compute the multiple output sequence at a lower average cost than multiple independent single channel filters [16, 18]. This approach is similar to the recursive decomposition approach used by the Fast Fourier Transform, which can be viewed as a bank of uniformly spaced frequency selective filters.

A third approach, often used in dedicated digital filtering hardware, is a special filter structure known as cascade-integrator-comb (CIC) filter or Hogenauer filter [4]. This technique uses a special filter structure with cascaded stages of accumulators and combs that can implement a pass band filter using no multiplication operations. This approach is effective where multiple addition operations are economical than a single multiplication.

These approaches have a number of characteristics is common such as:

1. Each approach statically specifies the filter pass band and stop band.
2. In each approach, filter complexity is proportional to input sample rate; as receivers are designed with wider input bandwidths, the cost of extracting a constant-width channel increases as well.
3. Finally each approach needs a larger number of input samples, relative to the direct approach of Eq. 2, to compute a particular output sample. In a sense they are less efficient in their use of each input sample relative to the single high order filter whose coefficients are optimized to provide a desired filter response. This increased input-output dependence is ameliorated by the fact that each input sample is used in the computation of multiple output samples.

3. Random Sub-Sampling

The goal of the channel filter is to remove adjacent channels from the wideband signal so that sample rate can be reduced without causing the aliasing of the other interfering signals into band of interest. This is accomplished by designing the filter to reject all potential interfering signals and only then reducing the sample rate in order to reduce the computational load in subsequent stages. It is assumed that the input to the channel filter is a sequence of uniformly spaced real-valued samples of a signal with bandwidth W_0.We wish to generate an output sequence that contains only those components of this signal that lies within a certain narrow frequency band $W_N << W_0$.

This is often accomplished using a decimating FIR filter that passes only the band of frequencies we desire. In such a filter, the values of the output sequence are computed from the input using discrete-time convolution:

$$y_n = \sum_{m=0}^{M} h_m r_{n-m} \tag{4}$$

Here h_m is the length - (M+ 1) sequence whose elements are the coefficients of the order-M FIR filter. The input sequence r_n is assumed to be infinite and n is the time index for the sample r_n. The order M of the filter has been chosen to sufficiently attenuate all out-of-band signal components so that, after filtering, we can compress our representation of the output signal by the decimation factor of D. This need to resolve and remove out-of-band components is the primary factor that drives the determination of the minimum length of the filter response. The compression of the output representation of decimating FIR filter is accomplished as we compute the output samples only for times n = kD. Each of these output samples will, of course, require M+ 1 multiplication and M additions to compute. It is clear that the filter order M is not chosen specifically to provide some required level of output SNR in the channel filter. Our idea is to produce output samples y'_n that only approximate the Y_n to the extent that they still provide the desired level of SNR at the output while requiring fewer operations to compute. We will compute these approximate samples by only partially evaluating the summation shown in 3 for each sample:

$$y'_n = \sum_{m \in s_n} h_m r_{n-m} \tag{5}$$

Where the selection set $S_n \subset \{0,1,....,M\}$ is the subset of the indices of the filter coefficients used to compute y_n. We would like to find a way to choose this subset that will adequately approximate the original output sequence y_n while using the smallest amount of computation, that is, using the smallest expected number of terms in each sum, $E\{|S_n|\}$.

Our investigation of this problem starts with the development of tools to provide a quantitative understanding of the effect of discarding input samples. We first introduce a new model that allows us to analyze the effect of discarding different sets of input samples. We also present an expression that represents the distortion caused by this operation of discarding samples while bounding the distortion measured at the filter output. This is equivalent to reducing the computation required in channel separation while maintaining some minimum level of SNR at the filter output.

The technique we present has two forms. The first makes no assumptions about the input signal: it simply discards samples randomly to reduce computation. The other form demonstrates that we can use knowledge about the input signal (not actual samples, but rather in terms of the expected distribution of energy at different frequencies) to further reduce computation for the same level of output distortion. This approach to reduce computations can be generalized in the number of ways, but in this work we restrict our consideration to a scheme that induces distortion that has a flat spectrum, that is, a type of distortion in which the error at each point in the sequence is uncorrected with the error at other sample points. This white distortion is often easier to deal with in subsequent processing stages, such as a detector.

4. Model for Analysis

We will use the model shown in Fig. 4 to analyze the effect of approximating the output samples through the use of random sub-sampling of the input.

In Fig.4, we see the original filtering operation, shown in part (a), as well as the model that produces the approximate output samples in part (b). In part (b), the block labeled "Sample Selector" controls which samples will be used in the computation of the approximate output, y'_n. This selection process is modeled as multiplication by the sequence Z_n: when a particular $Z_n = 0$, the corresponding r_n will not contribute to the computation. The non-zero Z_n can, in general, take on any value that will help us to reduce the approximation error; we discuss how the values of the non-zero Z_n are chosen later. We can now re-write the expression for the y'_n as:

$$y'_n = \sum_{m=0}^{M} h_m r_{n-m} z_{n-m} = \sum_{m=0, z_{n-m} \neq 0}^{M} h_m r_{n-m} z_{n-m} \tag{6}$$

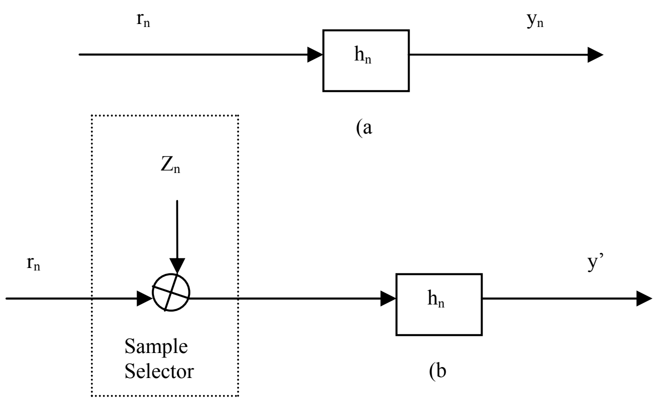

Fig.4. Diagram Showing Conventional FIR and Model for approximating output samples.

Before we proceed with a complete analysis of this model for sub-sampling and its effects on the filter output, it is helpful to present a short example of one simple random sub-sampling scheme. We perform a simple analysis of this example case to build some intuition for sub-sampling before analyzing a more general rule for discarding samples in the sections that follow.

4.1 Discarding samples using biased coin flips

One simple way to pick the values of Z_n in Fig. 4 is to use a biased coin that gives probability p of retaining a sample:

$$z_n = \begin{cases} 1/p - with - probability - p \\ 0 - otherwise \end{cases} \tag{7}$$

Here the samples that are retained are multiplied by the constant 1/p. To understand the effect on the filter output of this discarding of samples, we use standard results from signal processing that describe the effect of passing a random signal through a linear filter like the one in Fig. 4. Before showing these results, however, we need a few definitions.

We define the auto-correction sequence (ACS) of a real-valued random sequence S_n as the expectation:

$$R_s[n,m] = E\{s_n s_m\} \tag{8}$$

A random sequence S_n is wide-sense stationary (WSS) if:

1. The expected value, E $\{s_n\}$ is independent of time, and

2. The ACS can be written as a function only of the difference, k = m-n, between the samples in the expectation: R_s {n,m] = R_s [k] = E {s_n S_{n+k}}

When a sequence is WSS, we can also define the power spectrum density (PSD) as the discrete time Fourier transform (DTFT) of the ACS:

$$S_s(\Omega) = DTFT\{R_s[k]\} = \sum_{k=-\infty}^{\infty} R_s[k] e^{-j\Omega k} \tag{9}$$

The subscript s in both R_s [k] and S_s (Ω) refer to the original sequence S_n. We also note that whereas R_s [k] is a discrete sequence, S_s (**Ω**) is a continuous function in the frequency domain. We use **Ω** as the frequency domain parameter (as opposed to ω) to indicate that this is the transform of a discrete sequence and is therefore periodic in the frequency domain with period 2Π. The sequence of filter coefficients h_n is a finite-length deterministic sequence, and we will write its DTFT as:

$$H(\Omega) = DTFT\{h_n\} = \sum_{k=-\infty}^{\infty} h_k e^{-j\Omega k} \tag{10}$$

When a WSS random sequence is passed through a digital filter such as h_n, its output is also a WSS random sequence, and we can write the PSD of the output sequence in terms of, in the input PSD and the filter. For the filter shown in Fig. 4 (a), the output PSD is [Oppenheim and Schafer, 1989]:

$$S_y(\Omega) = DTFT\{R_Y[K]\} = |H(\Omega)|^2 S_R(\Omega) \tag{11}$$

For the case of our simple coin - flipping sample selector, we can now analyze the effect of discarding samples. The ACS and PSD of the selection sequence z_n defined in Eq. 6 are:

$$R_Z[k] = E\{z_n z_{n+k}\} = \begin{cases} 1/p, k = 0; \\ 1, k \neq 0 \end{cases} \tag{12}$$

$$S_z(\Omega) = \sum_{k=-\infty}^{\infty} R_z[k] e^{-j\Omega k} = (1/p - 1) + 2\prod \sum_{l=-\infty}^{+\infty} \delta(\Omega - 2\prod l) \qquad 13)$$

The PSD of the input to the filter in Fig. 4 (b) is the PSD of the product r_n z_n, which is the periodic convolution of S_r (Ω) and S_z (Ω) [6]:

$$S_{\{rz\}}(\Omega) = 1/2\prod \int_{2\Pi} S_r(\theta) S_z(\Omega - \theta) d\theta \tag{14}$$

Substituting from Eq. 13 and carrying out the convolution yields

$$S_{\{rz\}}(\Omega) = 1 - p/2\Pi\, p \int_{2\Pi} S_r(\theta) d\theta + \sum_{l=-\infty}^{+\infty} \int_{2\Pi} S_r(\theta) \delta(\Omega - \theta + 2\Pi l) d\theta \tag{15}$$

The second term in the right-hand side of this result reduces to simply S_r (**Ω**) because of the shifting property of the integration with the impulses and the periodic spectrum of the PSD. This PSD at the filter output can now be written as simply the original filter output from Eq. 10 plus a second additive term:

$$S_{y'}(\Omega) = |H(\Omega)|^2 S_r(\Omega) + |H(\Omega)|^2 \left(1 - p/2\Pi\, p \int_{2\Pi} S_r(\theta) d\theta\right) \tag{16}$$

This result in Eq. 15 helps us to understand the effect of discarding samples according to simple biased coin flips. The first term in Eq. 16 is equal to the PSD of the original output signal when no samples were discarded. The second term is additive and represents the distortion caused by discarding some samples. Note that when p=1 (all samples are used) the distortion is zero and the distortion increases as p decreases. In Fig. 5, the distortion is plotted as a function of the probability p of retaining each sample for typical values of h_n and $S_r(\Omega)$. The results shown in this figure are developed more fully in the next few sections, but we can see that as p decreases from one to near zero, the level of distortion increases to levels that exceed the signal of interest (at 0 dB in the plot, the power of the distortion equals the power of the signal itself).

4.2 Analysis of the error sequence

The example of choosing the values of z_n using a biased coin helped us to gain some intuition about the effect of discarding input samples as we compute the output of a narrowband channel filter. Discarding samples led to additive distortion in Eq. 16 in which PSD increased in magnitude as samples were discarded. In order to understand how to choose the selection sequence z_n in a manner that will allow us to more carefully control the induced distortion, we now provide a more general analysis to the effect of discarding samples.

The error between the approximate filter output y'_n and the original output sequence is defined in Eq.(17)

$$e_n = y'_n - y_n \tag{17}$$

To provide an unbiased approximation, the error sequence, e_n, should have zero mean () and for given choice of the sequence z_n. And itt is desirable to compute its variance, i.e., the mean squared error (MSE) of the distorted output sequence relative to the original output as shown in Eq. 18

$$Var(e_n) = E\{e_n^2\} \tag{18}$$

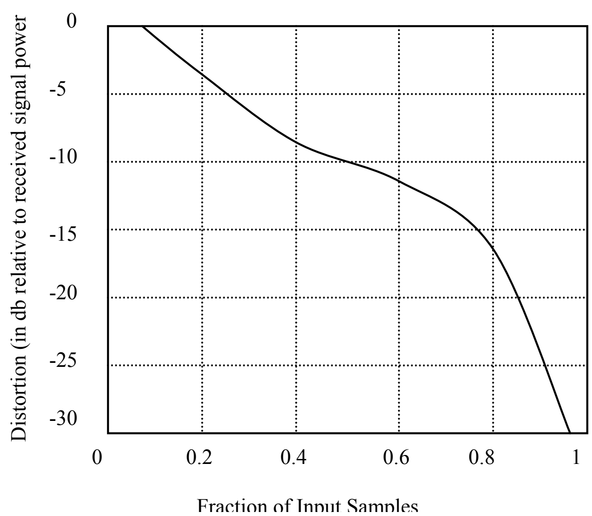

Fig.5. Distortion due to discarding input samples according to random coin flips.

In Fig. 6, the two schemes of Fig. 4 are combined to produce the error sequence for the purpose of analysis. Before using this combined model to derive the relationship between the input sequences and the output variance, we need to state a few assumptions that will simplify the derivation.

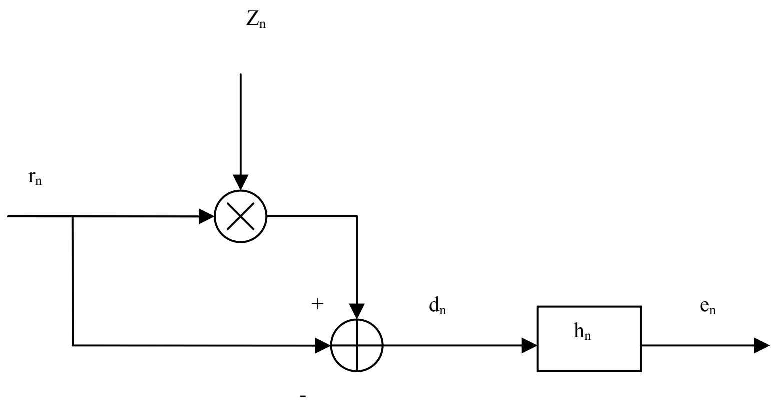

Fig. 6. Model for Analysis of Error Variance due to Random Sub-Sampling

We again assume that the input sequence r_n is wide-sense stationary. We also assume that the Z_n are chosen independently of the values of r_n. This allows us to fully realize the computational savings of discarding some input samples without examination, as well as allowing us to pre-compute z_n. It is important that the sequence z_n have a non-zero mean; we will see later that this determines the amplitude of the desired signal in the output sequence. Without loss of generality, we assume that E $\{z_n\} = 1$. This specific choice prevents problems with scaling factors later but does not limit our choice of sequences, as long as we scale them appropriately. We will also define $v_n = z_n$-l to simplify the notation in out analysis (So $E\{v_n\} = 0$). We can now write the ACS for the filter input sequence, $d_n = r_n z_n - r_n$ (the distortion sequence), as:

$$R_d[k] = E\{d_n d_{n+k}\} = E\{r_n r_{n+k} v_n v_{n+k}\} = R_r[k] R_v[k] \tag{19}$$

Our model for approximating the filtering operation has several desirable characteristics. First, if the length of the filter (M+1 coefficient) is greater than the decimation factor, then some input sample will be required in the computation of multiple output values. When this is the case, our model will ensure that these samples are used in every such computation or in none of them. This is useful in a real implementation where much of the cost of the computation is in retrieving a sample from memory, not in performing the actual arithmetic operation ($h_m r_{n-m}$).

In a sense, we are approximating the input sequence as opposed to approximating the filter. The selection set S_n can be viewed as the set of filter coefficients that are used to compute each output. If this set were the same for every n, it would simply define a new filter that is an approximation of the original filter. This approximation approach could be evaluated using standard filter response techniques.

4.3 Problem statement

Using the model shown in Fig. 6 and the definitions above, we now state our problem more precisely:

Find the sequence Z_n that minimizes the expected amount of computation required to approximate the filter output while ensuring that the error variance is less than or equal to a bound B:

$$\text{Min}[\Pr\{z_n \neq 0\}] \text{ such that } E\{e_n^2\} \leq B \tag{20}$$

4.4 Error variance

To analyze the effects of z_n on the variance of the error sequence, we first write the power spectrum density of the output of the filter in Fig. 6. The output PSD can be written in terms of the input PSD and the frequency response of the filter, similar to Eq. 11:

$$S_e(\Omega) = |H(\Omega)|^2 S_d(\Omega) \tag{21}$$

The variance of e_n can be written as the inverse DTFT of this PSD evaluated at k=O:

$$Var(e_n) = R_e[0] = \left[1/2\Pi \int_{2\Pi} S_e(\Omega) e^{-j\Omega k} d\Omega\right]_{k=0} = 1/2\Pi \int_{2\Pi} S_e(\Omega) d\Omega \tag{22}$$

Writing this variance in terms of the input sequence PSD results in

$$Var(e_n) = 1/2\Pi \int_{2\Pi} |H(\Omega)|^2 S_d(\Omega) d\Omega \tag{23}$$

The PSD of the input sequence, S_d (Ω), represents the distribution in frequency of the expected distortion (the squared difference,) caused by discarding samples according to the sequence z_n. To reduce the variance at the output we would ideally like this distortion to occur at frequencies for which the amplitude of H(Ω) in Eq. 22 is small: the stopband of the filter. If d_n is a white sequence then S_d (Ω) will be constant for all (as d_n must also be zero -mean). This implies, from Eq. 22 that the output variance will be simply proportional to the input variance, E $\{d^2{}_n\}$ and that is proportionality factor will depend only on the response of the bandpass filter, H(Ω).

If we substitute in Eq. 22 using the definition of the DTFT of R_d [k],

$$Var(e_n) = 1/2\Pi \int_{2\Pi} |H(\Omega)|^2 \left[\sum_{k=-\infty}^{\infty} R_d[k] e^{-j\Omega k}\right] d\Omega \tag{24}$$

Exchanging the order of summation and integration gives Eq. 25

$$Var(e_n) = \sum_{k=-\infty}^{\infty} R_d[k] \left[1/2\Pi \int_{2\Pi} |H(\Omega)|^2 e^{-j\Omega k} d\Omega\right] \tag{25}$$

The expression in the square brackets above is the inverse DTFT of the squared response of the filter. This can be written as the ACS of the deterministic, finite length sequence of filter coefficients h_n, which we will call C_h [k]:

$$C_h[k] = \sum_{m=-\infty}^{\infty} h[m] h[k+m] = 1/2\Pi \int_{2\Pi} |H(\Omega)|^2 e^{-j\Omega k} d\Omega \tag{26}$$

All of these results can now be combined to show that the output error variance is simply the sum of a product of three sequences:

$$Var(e_n) = \sum_{k=-\infty}^{\infty} R_r[k] R_v[k] c[k] = \sum_{k=-\infty}^{\infty} R_r[k] (R_z[k]-1) c_h[k] \qquad (27)$$

To check this result, consider the case of zero output distortion $Z_n = 1$ for all n results in R_v [k] = 0 for all k, giving zero error.

This result in Eq. 27 is significant for several reasons. First, it shows that the error variance of our approximation scheme depends on the sequence v_n (and hence z_n) only through its ACS. Second, it shows that the error variance depends linearly on all three of the key parts of the system: the ACS of the input sequence, the ACS of the selection sequence and the coefficients of the channel selection filter.

4.5 Selection sequence

Equation 27 can be used to choose the selection sequence z_n, minimizing the amount of computation to produce an output with a specified bounded variance. This sequence will have the following properties:

1. It provides a bounded error variance, given by Eq. 26.
2. It allows us to discard as many as possible that is to maximize the probability that Z_n =0, subject to property (1) above.
3. It produces a distortion with uncorrected error values at each point in the sequence.

We now identify two separate cases as we try to decide which samples to discard. In Eq. 27, the induced error variance depended only on the ACS of the received wide band sequence and the ACS of the channel filter. Although it is conceivable that in some cases we may have a good idea of the spectral distribution of the received signal (and therefore its ACS), we may not always have this information. We therefore identify two cases as we try to find a good choice for the selection sequence, Z_n.

Although it is well known how to generate a random sequence with a desired ACS (e.g. by generating 'shaped noise' [21], we found no prior work on how to directly generate such a sequence with a relatively high probability that z_n=0. Instead, we will start with a candidate selection sequence, that has a desirable ACS and perform a sequence transformation to convert it to a sequence that has more zero elements and an ACS that remains "close" to that of x_n, i.e. This sequence transformation approach is shown in Fig. 7, which also depicts the creation of the initial sequence by the filtering of a white noise sequence with the shaping filter. We will describe how the filter is chosen in a later section.

In terms *of our* transformation scheme, the error variance of Eq. 26 can be written as two components: one component due to choice *of* the original sequence (the first term below), and another to the transforming effect that introduces more zeros and produces z_n.

$$Var(e_n) = \left[\sum_{k=-\infty}^{\infty} R_r[k] (R_x[k]-1) c_h[k]\right] + \left[\sum_{k=-\infty}^{\infty} R_r[k] (R_z[k] - R_x[k] c_h[k])\right] \qquad (28)$$

A similar decomposition to that of Eq. 18, which gives ACS for the distortion sequence, can be performed. This decomposition also has two terms: the first term is the portion of the autocorrelation due to the initial choice of, the second term corresponds to the part of the distortion due to the transformation process:

$$R_d[k] = R_r[k]R_v[k] = R_r[k](R_x[k]-1) + R_r[k](R_z[k]-R_x[k]) \tag{29}$$

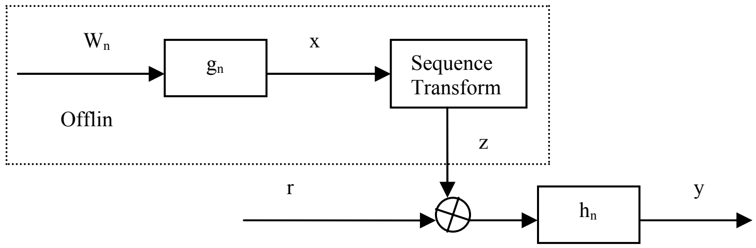

Fig.7. Random sub-sampling using a transformed sequence for sample selection

Because we want the distortion to be uncorrelated at each sample (white), we would like $R_d[k]$ to be non-zero only for k=0. To achieve this goal requires that for all $k \neq 0$:

1. First, choose the sequence to make the first term in the right-hand side of Eq. 28 equal to zero, and
2. Then, choose the transformation such that $(R_z[k]-R_x[k]) = 0$ in the second term.

5. Conclusions

In this paper authors have analyzed the effect of approximating the output samples through the use of random sub-sampling of the input for the minimization of computation in the filtering operation. The study is useful for the energy optimized design of digital filter for channel separation in the receiver useful for communication applications.

Acknowledgement

Authors are thankful to The Ministry of Human Resources And Development for grant of project of Rs. 12 lack for up gradation of DSP lab for teaching and research.

References

1. Mehendale, M., Sherlekar, S.D., & Venkatesh, G. (1995). Coefficient optimization for low-power realization of FIR filters. *IEEE Workshop on VLSI Signal Processing*, Japan.
2. Merakos, P.K., Masselos,K., Koufopavlou, O., Nikolaidis, S. & Goutis, C. E. (1997). A novel transformation for reduction of switching activity in FIR filter implementation. *Proceedings of Int. conf. Digital Signal Processing*. Santorini, Greece, 653-656.

3. Graham, R., Knuth, D. and Potashnik, D. (1994). *Concrete Mathematics*. Essex: Addison-Wesley.
4. Mitola, J. (1995). The software radio architecture. *IEEE Communication Magazine*, May, 26-38.
5. Wipman, J.A. (1995). Analog to digital converter and their applications in radio receiver. *IEEE Communication Magazine*, 33(5), 39-45.
6. Oppenheim, A V & Schafer, R W. (1989). *Discrete Time Signal Processing*. New Delhi: Prentice Hall India.
7. Lee, E.A. & Messerschmitt, D.G. (1994). *Digital Communication* (3rd ed.). New York: Springer.
8. Frerkering, M.E. (1994). *Digital Signal Processing in Communication System*. Kentucky: Van Nostrand Reinhold.
9. Zangi, K. & Koilpillai, R.D. (1999). Software radio issue in cellular base-stations. *lEEE lSAC Issue on Software Radio*, 20-35.
10. Orfanidis, S. (1996), Introduction to Signal Processing. Prentice Hall.
11. Crochiere, R. & Rabiner, L. (1981). Interpolation and decimation: A tutorial review. *Proceedings of IEEE*. 18-45.
12. Baines, R. (1995). The DSP bottleneck. *IEEE Communication Magazine*, 33(5), 46-54.
13. Hogenaver, E. (1981). An economical class of digital filters for decimation and interpolation. *IEEE Transaction on Acoustia, Speoch and Signal Processing*, (6)18-32.
14. Singh, A. & Singh, J. (2002). Computer aided analysis of FIR filter for software radio. *Journal of Punjab Academy of Sciences*, 1, 43-48.
15. Singh, A. & Singh, J. (2001). Computer modeling and analysis of random sub-sampling in software radio receiver. Systems, *Proceedings of Man, and Cybernetics, 2001 IEEE International Conference*. Tucson, AZ, USA, 3484 - 3488.
16. Singh, J., Singh, A., Sharma, D., Bhatti, S.S. & Kaur, K. (2000). Information processing in digital domain. *Proceedings of Electronics, Circuits and Systems (The 7th IEEE International Conference)*. Jounieh, Lebanon, 579 -582.
17. Singh, J., Singh, A., Sharma, D., Bhatti, S.S. & Kaur, K. (2000). Digital signal processing in digital domain – Recent trends in development. *Proceedings International Conference Asia Pacific Telecom – 2000 on Advances in Telecommunication and Information Technology*, 395 – 401.
18. Singh, A. & Singh, J. (2001) Flexible signal processing in software radio receiver. *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*. Salt Lake City, UT, USA.
19. Singh, A. & Singh, J. (2003). Complex FIR filter for channel separation in software radio receiver. *Journal of the Institution of Engineers (India)*, (83), 76-81.
20. Singh, A., Marwaha, S. &, Marwaha, A. (2006). Management of power reduction techniques in DSP domain. *Proceedings of Asian Conference on Intelligent Systems & Networks*. Jagadhri, Haryana, India, 24-25.
21. Stark, H. and Woods, J.W. (1986). *Probability, Random Processes, and Estimation Theory for Engineers*. New Jersey: Prentice Hall.