NARROWBAND CHANNEL EXTRACTION FOR WIDEBAND RECEIVERS

Matthew L. Welborn

Software Devices and Systems Group
MIT Lab for Computer Science
mwelborn@lcs.mit.edu

Abstract

One of the most computationally intensive processing stages of a wideband digital receiver is the extraction of a narrowband channel from a wideband input signal. In implementations that compute the convolutional sum, the computation is proportional to the bandwidth of the input signal. This paper shows how to break this dependence, reducing the limiting factor to the requirement to maintain a sufficient output signal-to-noise ratio (SNR).

This paper describes two complementary algorithms for efficient channel extraction in wideband receivers. The first allows the required frequency translation to be performed at the lower sample rate of the channel filter output. The second algorithm decouples the effect of interference rejection from SNR improvement and improves the computational efficiency of filtering by using only a subset of the input samples. Additionally, we present a simple model to quantify the effects of this technique and experimental verification using a wideband software radio receiver.

1. INTRODUCTION

Flexibility is an important characteristic in many wireless communication systems employing DSP. The work in this paper is motivated by a desire to use an extremely flexible DSP system, a software radio, to improve computational efficiency relative to conventional systems by using novel computation structures and algorithms. This work specifically focuses on the multirate problem of extracting a narrowband signal from a wideband sample stream in a wideband digital receiver.

A first-order estimate of the computational complexity of a wideband digital receiver is the work required to extract the individual channels from the output of the ADC [7, 11]. Several techniques have been used to make this high complexity task more manageable. One idea is to use dedicated digital filtering hardware [1]. Digital down-converter chips are available that provide this function using cascaded integrator-comb (CIC) filters while maintaining some amount of programmability, see for example [4]. For some situations, polyphase techniques can be used to efficiently separate multiple, equally-spaced signals. [5].

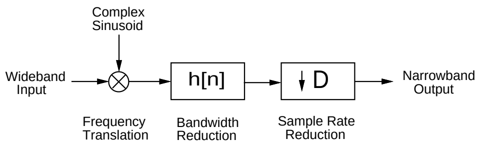

Figure 1: Typical processing for narrowband channel selection.

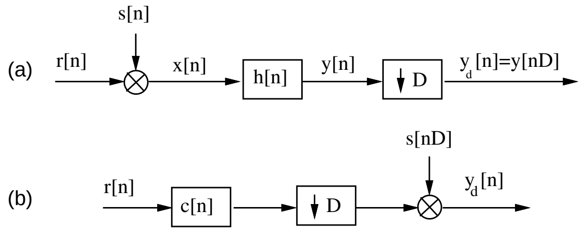

Figure 2: Block diagram showing (a) frequency translation of desired signal before filtering and decimation and (b) interchange of frequency translation and bandwidth reduction steps.

When we wish to extract the channels independently, however, the typical approach is to implement a narrowband digital filter to extract the channel as shown in figure 1. In this direct implementation the computational complexity is proportional to the high input sample rate. This paper presents techniques to remove this dependence and reduce the complexity of the channel extraction process. In section 2 we present an algorithm which performs frequency translation with computational complexity proportional to the lower *output* sample rate. Section 3 describes how we separate the bandwidth reduction step into two distinct effects: interference rejection and SNR improvement. This decoupling of effects is used in section 4 to reduce the complexity of the bandwidth reduction step, allowing us to perform less work and still achieve the desired result using *random sequence modulation*. We also present experimental results which validate the algorithms using a software radio implementation.

2. FREQUENCY TRANSLATION

If we assume that the frequency translation of figure 1 will be followed by a decimating FIR filter, then we have the situation shown in figure 2a. Here $r[n]$ is the received wideband sample sequence, $h[n]$ is the order-M channelization filter, $y[n]$ is the filter output and $y_d[n]$ 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 baseband. The output of the cascaded translation and filtering steps is:

$$y[n] = \sum_{m=0}^{M} h[m]x[n-m] = \sum_{m=0}^{M} h[m]r[n-m]e^{-j2\pi f_c(n-m)T} \tag{1}$$

where $e^{-j2\pi f_c nT}$ is the complex sinusoid ($s[n]$ in figure 2), f_c is the original carrier frequency and T is the sample interval. We now combine steps to define a new set of filter coefficients:

$$y[n] = e^{-j2\pi f_c nT} \sum_{m=0}^{M} c[m] r[n-m] \tag{2}$$

where $c[m] = h[m]e^{j2\pi f_c mT}$ are new composite filter coefficients. Figure 2b shows that the steps of frequency translation and filtering have effectively been reversed and the multiplication required for the translation now occurs at the lower output rate. This technique can also be applied when a a cascade of multiple FIR filters is used.

The disadvantage of this algorithm is that we must compute the composite filter coefficients before we begin filtering or if we tune the filter to a different carrier frequency. In a software-based system, such as a software radio, we have the computational capability to perform this pre-processing and sufficient memory to store multiple filter definitions if desired.

3. BANDWIDTH REDUCTION

Although it is relatively easy to understand the modification of the frequency translation step in the previous section, it is more difficult to see how to modify the bandwidth reduction step to remove the dependence on the input sample rate. To understand why the dependence arises, it is useful to look more closely at two specific effects of filtering: interference rejection and SNR improvement. Separating these two effects will allow us to design a filter sufficient to reject adjacent channel interference while performing the minimum work required to improve or maintain the output SNR.

In a receiver where we extract a narrowband signal from a wideband sample stream, it is often necessary to specify a sharp transition between passband and stopband to reject adjacent channel interference. This requirement leads to a large number of taps in the resulting FIR filter, a direct implementation of which will have a computational complexity proportional to the input sample rate [6]. A second consequence of using a high-order FIR filter is that we can get a significant improvement in the SNR of the output signal relative to the input signal. In fact, with appropriate filtering we can improve the SNR by 3 dB for every halving of the sample rate as we extract a narrow channel [11].

So the capability to reject adjacent channels is related to the *length* (in time) of the impulse response of the channel filter, while the improvement in SNR due to sample rate reduction (as well as the amount of computation required) depends on the *number* of input samples used to compute each output sample. To decouple the two effects, we will compute the filter output using only a subset of the available samples. We will implement a filter with a sufficiently long time response to provide the sharp transition while using only as many samples over that interval as necessary to produce or maintain the required output SNR. In the following section we present one simple approach to determine which subset of samples can be used in this computation.

4. RANDOM SEQUENCE MODULATION

Using only a subset of the wideband sample stream to compute the channel filter output will result in some distortion of the wideband signal. The object of this work, is to determine how we can minimize this effect within the narrow band of interest. The technique presented here is based on ideas from the area of randomized signal processing presented in [2]. However, we use random choice

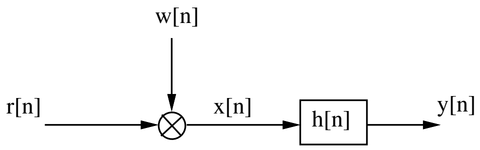

Figure 3: Diagram of model used to determine the effect of using a subsequence of input sample stream to compute channel filter output.

in a different way, introducing the randomness while processing a stream of uniformly spaced samples, as opposed to introducing the randomness while performing quantization or sampling of the original analog signals.

4.1. Model for Analysis

From the previous section we can see that the goal of channelization is to remove adjacent channels from the wideband signal so that sample rate can be reduced without causing the aliasing of other signals into band of interest. We will use the simple model shown in figure 3 to study the effect of using only a subset of the available samples to compute the filter output. In this model $r[n]$ is the received wideband sample sequence, $h[n]$ is the channelization filter and $y[n]$ is the filter output. (For simplicity, this model does not include the step of frequency translation, but this will not affect the results here.) To specify which samples are used in the filtering operation, we will multiply $r[n]$ by the sequence $w[n]$ to get $x[n]$, where the values of $w[n]$ will be either zero (indicating that the sample is not used) or one (sample is used). Such a model could be efficiently implemented by only performing the multiplications corresponding to non-zero values of $x[n]$:

$$y[n] = \sum_{\substack{m=0 \\ w[n-m]\neq 0}}^{M} h[m]x[n-m] \tag{3}$$

This model can be used to select any desired subsequence of the wideband signal. For example, we could immediately perform decimation of the received sample stream by the integral factor D *before* filtering by choosing $w[n]$ as a sequence of uniformly spaced ones and then retaining only the non-zero product samples:

$$w[n] = \begin{cases} 1, & n = kD, k = 0, \pm 1, \pm 2, ... \\ 0, & otherwise \end{cases} \tag{4}$$

Clearly this is not an appropriate choice for $w[n]$ because it causes aliasing of adjacent narrowband signals into the band of interest. Aliasing of interfering signals will be a problem whenever we select a subsequence that is uniformly spaced, or even a non-uniform but periodic sequence. If we instead choose $w[n]$ as a random sequence, we see that the effect of the sequence multiplication is very different. For example, let us make the decision of whether to use each sample according to the outcome of an independent biased coin flip; let us choose $w[n]$ as an *i.i.d.* Bernoulli sequence:

$$w[n] = \begin{cases} 1, & \text{with probability } p \\ 0, & \text{with probability } (1-p) \end{cases} \tag{5}$$

To see the effect of this choice on the product sequence, we find the autocorrelation sequence, $R_W[k]$, and its discrete-time Fourier transform (DTFT), the power spectrum density (PSD), $S_W(\Omega)$:

$$R_W[k] = E\{w[n+k]w^*[n]\} = p^2 + (p - p^2)\delta_k = \begin{cases} p, & k = 0 \\ p^2, & k \neq 0 \end{cases} \quad (6)$$

$$S_W(\Omega) = \sum_{k=-\infty}^{+\infty} R_W[k]e^{-j\Omega k} = (p - p^2) + 2\pi p^2 \sum_{l=-\infty}^{+\infty} \delta(\Omega - 2\pi l) \quad (7)$$

In (6) δ_k is a unit pulse at $k = 0$ and in (7) $\delta(\cdot)$ is the Dirac delta function (a unit impulse). The argument Ω is used in the PSD to indicate that this is the transform of a discrete sequence. We see in (7) that the PSD of $w[n]$ has both a constant portion due to the independence of the $w[n]$ samples and an impulsive part due to its non-zero mean. When we perform the sequence multiplication to produce $x[n]$ we can quantify the effect of the random sequence by performing the periodic convolution of the PSDs of the two original sequences [9]:

$$S_X(\Omega) = \frac{1}{2\pi}\int_{2\pi} S_R(\theta)S_W(\Omega - \theta)d\theta \quad (8)$$

Here $S_R(\Omega)$ is the PSD of the received sequence $r[n]$ which we will treat as a random sequence for this analysis. If we substitute from (7) and carry out the convolution we get:

$$S_X(\Omega) = \frac{p - p^2}{2\pi}\int_{2\pi} S_R(\theta)d\theta + p^2 \sum_{l=-\infty}^{+\infty} \int_{2\pi} S_R(\theta)\delta(\Omega - \theta + 2\pi l)d\theta \quad (9)$$

This can be written as simply a scaled version of $S_R(\Omega)$ plus additive noise which is proportional to the average power in the original signal $r[n]$:

$$S_X(\Omega) = p^2 S_R(\Omega) + \left(\frac{p - p^2}{2\pi}\right)\int_{2\pi} S_R(\theta)d\theta \quad (10)$$

In contrast to the narrowband signal aliasing of the *uniform* sub-sampling case discussed above, here we see that the effect of the random sequence multiplication is a much more benign addition of wideband uncorrelated noise. If we then complete the processing by filtering as indicated in figure 3, we will be able to extract the desired channel with some additional amount of additive noise. In addition, by controlling the probability, p, of selecting each sample for use, we can control the amount of computation expended in computing to output while ensuring that we meet any required specification for output SNR.

4.2. Correlated Random Sequences

We notice in (10) that as we reduce the number of samples used (by decreasing p), the spectral density of the original signal decreases in proportion to p^2 while the spectral density of the additive noise only decreases as p. This might cause the noise level to overwhelm the signal with only a small reduction in computation. If we look instead at the more general case where $w[n]$ is wide-sense stationary (WSS) and $w[n] \in \{0, 1\}$, but where the samples of $w[n]$ are not necessarily independent, its PSD can be written as a continuous part plus an impulsive part:

$$S_W(\Omega) = S_{W_0}(\Omega) + 2\pi\beta \sum_{l=-\infty}^{+\infty} \delta(\Omega - 2\pi l) \quad (11)$$

Here we use $S_{W_0}(\Omega)$ to indicate the PSD of the same sequence $w[n]$ if we subtracted the expected value ($E\{w[n]\}$) from each sample and the magnitude of the impulsive part, β, is simply the square of the same expected value($E\{w[n]\}^2$). When we look at the resulting output of the channelization filter we will get:

$$S_Y(\Omega) = |H(\Omega)|^2 \beta S_R(\Omega) + \frac{|H(\Omega)|^2}{2\pi}\int_{2\pi} S_R(\theta)S_{W_0}(\Omega - \theta)d\theta \quad (12)$$

Here $H(\Omega)$ is the DTFT of the channel filter shown in figure 3. In this more general result the first term on the RHS of (12) is a filtered and scaled version of the original wideband signal: this is the desired output signal. The second term in the RHS of (12) is a filtered version of what was white noise in (10) but which is now, in general, non-white. This is the term we want to minimize to get the best possible output SNR. Because our high-order channel filter will already have good stopband attenuation, we can choose $w[n]$ to minimize the value of $\int_{2\pi} S_R(\theta)S_{W_0}(\Omega - \theta)d\theta$ in the passband of the filter $H(\Omega)$ for a given β. This is a more useful result because it shows that with knowledge of the wideband signal we can potentially design a random sequence which will cause more of the aliased signal components to be outside the filter passband, thus improving the output SNR relative to (10).

To summarize the results of this section, we have shown that the computational complexity of the channel extraction process need not depend on the high input sample rate of the wideband receiver frontend. The *random sequence modulation* technique presented here allows us to understand the effect of using only a subset of the wideband input sample stream to compute the output of the channel filter. Although the channel filter is a high-order filter, we need not evaluate all of the taps in order to attain the sharp transition from passband to stopband. The number of terms that we need to evaluate is instead determined by the desired SNR of the output signal. In the special case of choosing $w[n]$ as an *i.i.d.* Bernoulli sequence, (10) shows that discarding some samples produces wideband uncorrelated noise at the output. For the more general case in (12) we see that the result of not using some samples depends upon both the PSD of the input signal and the PSD of the random sequence $w[n]$. To minimize the power of the signal components aliased into the band of the desired signal, we should minimize the result of the convolution of the two PSDs above within the passband of the channel filter.

4.3. Experimental Verification

The algorithms developed in this work have been implemented in a software radio system designed and built as part of the SpectrumWare research project at the MIT Lab for Computer Science. A simple block diagram of the implementation is shown in figure 4. This system uses a wideband RF frontend that is designed to receive 10 MHz of RF spectrum and produces a stream of 12-bit samples at 25.6 Msamples/sec. The sample stream is transferred into the memory of the host processing system (in this case a 200 MHz Pentium Pro PC) and all of the signal processing is performed using software running on a Linux operating system. More details on the system are available in [3, 10].

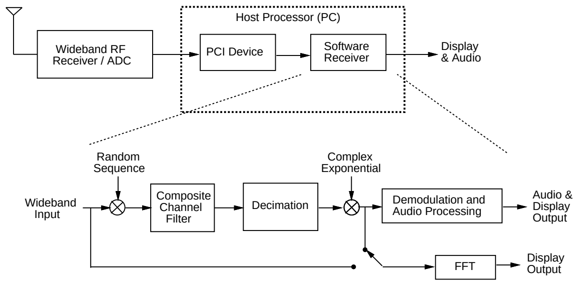

Figure 4: System block diagram for wideband FM receiver

The implementation of both the frequency translating filter and the random sequence modulation can be combined as shown in figure 4. The software receiver performs demodulation of the FM signals in real-time (not possible using a direct implementation without these algorithms) and can provide frequency domain display of the entire 10 MHz band or any filtered sub-band.

In order to test the performance of the random sequence modulation, the receiver was used to extract a 30 kHz-bandwidth FM signal using a random set of subsamples to compute the filter output. Figure 5 shows a plot of both the predicted and measured SNR degradation at the filter output as the percentage of samples used was varied from 100% to 10%. The predicted values are computed using (10) and the measured SNR is estimated by using FFTs of the output sequence with the signal both inside and outside the filter passband. The amount of computation required to perform the channel filtering was measured by counting CPU cycles and was determined to decrease linearly as the number of samples processed was reduced from 100% to 10%, with some small constant filter overhead.

5. CONCLUSION

In a wideband digital receiver, narrowband channel extraction can be decomposed into two separate steps: frequency translation and bandwidth reduction. This paper has shown that the first step can be performed with a computation rate that is proportional to the lower, output sample rate by pre-computing a set of composite filter taps. The second step, bandwidth reduction, has been shown to encompass two distinct effects: rejection of out-of-band signals and SNR improvement. The random modulation technique presented in this paper allows us to decouple these effects and compute an output signal by evaluating only a random subset of the taps of an FIR filter. This allows us to perform only the minimum amount of computation required to produce a satisfactory output SNR while still preventing narrowband aliasing into the output signal.

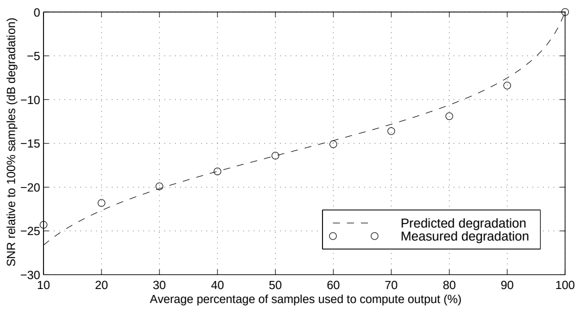

Figure 5: Predicted and measured filter output SNR (relative to using 100% of the samples) versus percentage of samples used to compute the filter output.

The results presented in this paper also highlight some of the advantages of a flexible, software-based receiver implementation. This flexibility allowed us to use more sophisticated algorithms in order to provide decreased computational complexity. This is clearly seen in several places:

- The composite filter design shown in (2) is able to incorporate part of the frequency translation function and improve overall efficiency.
- The random sequence filtering algorithm is able to separate the interference rejection and SNR improvement aspects of the filtering in order to provide more efficient channel extraction.

This capability to incorporate more flexible and sophisticated algorithms in a real-time implementation will be the subject of future work as we seek to extend the present results. One approach will be the incorporation of wideband signal information, as indicated in section 4, to improve the performance of the random sequence modulation algorithm.

6. REFERENCES

[1] Rupert Baines. The DSP Bottleneck. *IEEE Communications Magazine*, 33(5):46–54, May 1995.

[2] Ivars Bilinskis and Arnolds Mikelsons. *Randomized Signal Processing*. Prentice Hall, 1992.

[3] Vanu G. Bose and Alok B. Shah. Software Radios for Wireless Networking. In *Proceedings of Infocomm '98*.

[4] Harris Semiconductor Corporation. *Digital Signal Processing Data Book*. 1994.

[5] Fred Harris. Tutorial on Polyphase Transforms. San Diego State University - unpublished note.

[6] Leland B. Jackson. *Digital Filters and Signal Processing*. Kluwar Academic Publishers, 1989.

[7] Joe Mitola. The Software Radio Architecture. *IEEE Communications Magazine*, 33(5):26–38, May 1995.

[8] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[9] Alan V. Oppenheim and Alan S. Wilsky. *Signals and Systems*. Prentice Hall, 1983.

[10] M. Welborn J. Guttag V. Bose, M. Ismert. Virtual Radios. *to appear in JSAC issue on Software Radios*, 1998.

[11] Jeffery A. Wepman. Analog-to-Digital Convertors and Their Applications in Radio Receivers. *IEEE Communications Magazine*, 33(5):39–45, May 1995.