Improvements to Spread Spectrum Communications Systems via the Wavelet Transform

Jon DeShazo
University Undergraduate Research Fellow, 1994-95
Texas A&M University
Department of Electrical Engineering

APPROVED
Undergraduate Advisor Andrew Chan
Exec. Dir., Honors Program

Bibliography

Robert C. Dixon, *Spread Spectrum Systems with Commercial Applications,* Third Edition, John Wiley & Sons, New York, 1994.

Jerry D. Gibson, *Principles of Digital and Analog Communications,* Second Edition, Macmillan, New York, 1993.

M. Medley, G. Saulnier and P. Das, "Applications of the Wavelet Transform in Spread Spectrum Communications Systems," *SPIE* Vol. 2242 *Wavelet Applications,* pp. 54-68, 1994.

J. Proakis and M. Salehi, *Communications Systems Engineering,* Prentice Hall, Englewook Cliffs, New Jersey, 1994.

O. Rioul and M. Vetterli, "Wavelets and Signal Processing," *IEEE Signal Processing Magazine,* pp. 14-38, October 1991.

B. Vidakovic and P. Müller, "Wavelets for Kids," Duke University, submitted to AMS 1991.

Introduction

Over the past several years, spread spectrum radio communication systems have been increasingly viewed as attractive alternatives to more common methods of radio communication. The military has long valued spread spectrum communication techniques, but commercial vendors are now embracing this technology. This research hoped to combine spread spectrum communications with a relatively new approach to information processing, *wavelet technology*, to improve the performance of today's spread spectrum technology.

Spread Spectrum Communication Technology

Spread spectrum communication is an extension of the types of digital communication used widely for many years. The military has used spread spectrum techniques in its secure communication equipment. Communications equipment designers are only now applying spread spectrum technology to commercial applications, as wireless communication devices use increasingly precious frequency space to connect people and computers everywhere.

Spread spectrum techniques literally spread the frequency content of normal signal (see Figure 1). Spreading the signal lowers the average energy density of the radio signal; this provides several benefits over non-spread signals, such as selective addressing, multiple access, low probability of intercept, signal encryption, and high-resolution ranging.

Selective addressing and multiple access allow many radio devices to use the same general frequency space without interfering with each other and without receiving unnecessary or unauthorized information. In selective addressing, a spread spectrum transmitter dispatches a signal with a unique code at the beginning of the intended user's information. This pattern identifies the intended receiver, allowing

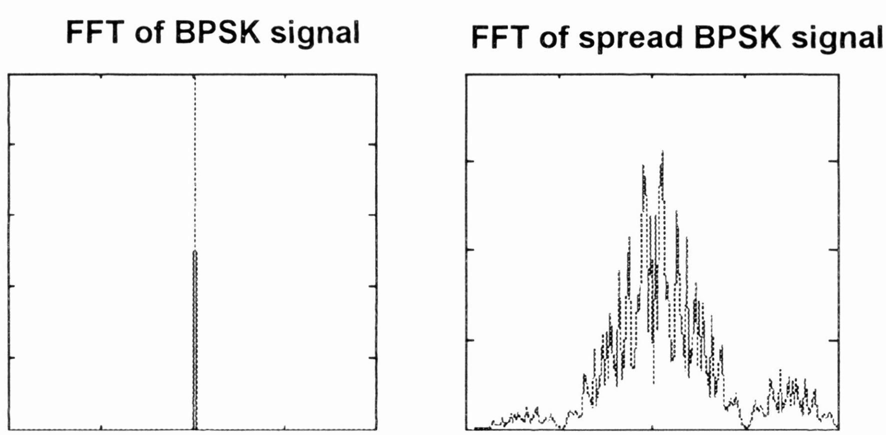

Figure 1: FFT spectra of normal BPSK and spread BPSK signals
(note: peak of BPSK signal is over 6 dB higher than spread signal)

(or requiring) other receivers to ignore the rest of the message and continue searching other frequencies and time spaces for its intended message. Spread spectrum techniques allow several communications devices to use the same or nearly the same frequency space for individual communication.

In this project, the spread spectrum system is based on simple binary phase-shift key (BPSK) modulation, which will be used to show how one form of spread spectrum, direct sequence, works. In BPSK modulation, a transmitter interprets bit sequences as phase shifts on a sinusoidal carrier wave. It does this by simply mixing the bit stream values with the carrier wave. A phase shift of 0 degrees is normally the "1" bit, and a phase shift of 180 degrees is the "-1" ("0") bit. The receiver mixes this incoming signal with an identical (or nearly so) carrier wave. The resulting signal has two parts: a DC signal carrying the bit information, and a high-frequency signal that is usually filtered out.

The direct sequence spread spectrum (DSSS) technique further modulates the transmitted signal with a sequence of pseudorandom bits whose rate is much higher than the information bit rate. This spreads the signal in the frequency plane and scrambles the information carried in the signal. The receiver mixes an identical sequence with the incoming signal to despread the signal. If the receiver mixes the wrong pseudorandom sequence, the incoming signal will be indistinguishable from random noise. DSSS techniques make a communications system much more complicated than simple BPSK systems, but its resistance to casual eavesdropping and interference more than outweigh its complexity.

Background: Wavelets and Multiresolution Analysis

Wavelet theory is a recent breakthrough in mathematics particularly useful for analyzing signals (radio, sound, video or otherwise) in both the time and frequency domains simultaneously. Early studies of the basics of this type of analysis were made almost 100 years ago, but the lack of sufficiently powerful computers made serious application impossible until about 10 years ago.

The Fourier transform must be compared with the wavelet transform in order to properly understand the way that wavelets work. The Fourier transform operates on the theory that any time signal can be replicated with an infinite number of infinite-duration sine and cosine waves, each at a certain frequency that is, the sine wave is the basis function of the transformed signal.

The Fourier transform works well for *stationary* signals — those signals whose components do not change over the period of analysis. For instance, a signal made up of a mixture of sine waves is a stationary signal and could easily be analyzed with Fourier techniques. However, any fast transitions in a signal are spread throughout the frequency axis, making close observation impossible and rendering frequency analysis useless in such situations.

Two methods attempt to address this shortcoming. One cuts the signal into different time windows in which the signal is relatively stationary. Each window is transformed individually, creating a series of "local" Fourier transforms. Another method modifies the sine wave basis functions to be time-limited.

Gabor first adapted the Fourier transform for the time-frequency plane, creating blocks of information corresponding to the locations of the time-frequency window. The effect is similar to a mosaic painting: the information is accurate to a certain point. In the case of Gabor's short-time Fourier transform, or STFT, accurate time resolution yields poor frequency resolution, and vice-versa. Also, a given STFT fixes the size of the windows; the transform cannot tune its window for different frequency bands.

Wavelet transforms seek to overcome this limitation by allowing the shape of the time-frequency window to vary over the plane. In fact, the frequency window is proportionate to the frequency being analyzed. This creates a multiresolution analysis that allows time resolution to become relatively good at high frequencies (where accurate frequency resolution is less important), while frequency resolution becomes more accurate at low frequencies.

The *Continuous Wavelet Transform* (CWT), similar to the Fourier transform, computes the wavelet coefficient on every point on the time-frequency plane. The basis function used is always localized in time—hence the term wavelet (or *little wave*).

Wavelet transforms are not based on sine waves; they require basis functions that are completely localized or that decay very quickly to zero. Any square integrable function with zero mean can become the basis for the continuous wavelet transform. In either case, wavelet analysis no longer renders frequency information in the conventional (Fourier) sense. CWT analysis produces time-scale analysis rather than pure frequency analysis. The term *scale* is more accurate and is used commonly in wavelet circles.

$\psi(t)$ is the mother wavelet or primary waveform, and it can generate a complete set of wavelets $\psi_{a,b}(t)$ by the following formula, where a is the time-dilation factor and b is the time-shift factor:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right).$$

Just as in Fourier or STFT analysis, the CWT of a signal is created by taking the inner product of the signal and the scaled wavelet:

$$\tilde{X}(b,a) = \frac{1}{\sqrt{|a|}}\int x(t)\psi\left(\frac{t-b}{a}\right)dt.$$

Wavelet analysis results in a set of coefficients that indicate how close a signal is to the wavelet's basis function; that is, a signal is decomposed into wavelets of constant shape but different scale, size, and amplitude. These wavelets combine to recreate the original function. This concept of perfect reconstruction is satisfied whenever the signal being analyzed is of finite energy and the wavelet satisfies the multiresolution analysis.

In discrete time application, the discrete wavelet transform (DWT) takes the place of the CWT. The physical manifestation of the DWT is a bank of complementary highpass and lowpass filters followed by downsample filters (devices that remove every other discrete time element), which double the scale of the output signal at each filter (see Figure 2). Passing the signal through this filter bank creates a

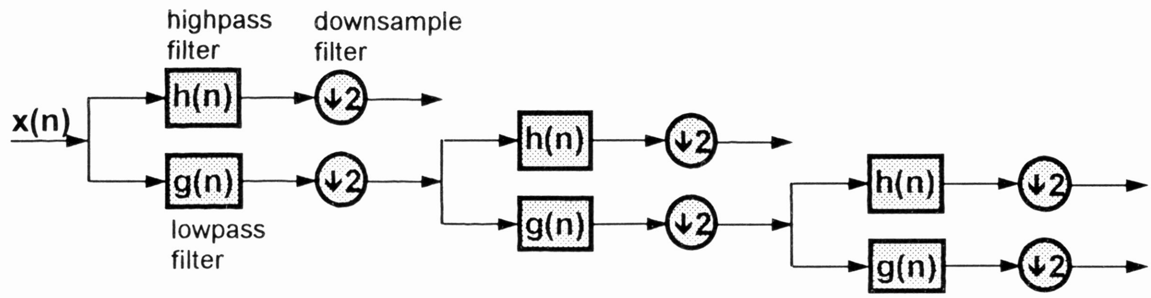

Figure 2: Filter and downsample process for the discrete wavelet transform

coarse approximation of the signal from the lowpass filter and detail information from the highpass filter. Each successively approximated signal is sent through an identical filter bank until no more useful information can be obtained. No information is lost when the signal elements are downsampled, and the original signal can be reconstructed perfectly from its wavelet elements.

This simplicity is the beauty of the discrete wavelet transform. The mathematical basis for wavelet transforms is too complex to be included here, but the actual implementation is straightforward.

The Application

Spread spectrum communication is exploding in popularity as new wireless devices cover scarce radio space. Wavelet transform methods are popping up in myriad signal processing applications. In this project, I hoped to learn about and combine direct sequence spread spectrum and wavelet transforms. I succeeded in the learning part; the direction I took in using wavelet transforms to decode incoming direct-sequence spread spectrum signals yielded decidedly mixed results.

The idea is simple: use the continuous wavelet transform to directly decode an incoming BPSK spread spectrum signal (see Figure 3), and compare the resulting bit information to the bit information recovered by the standard method (see Figure 4).

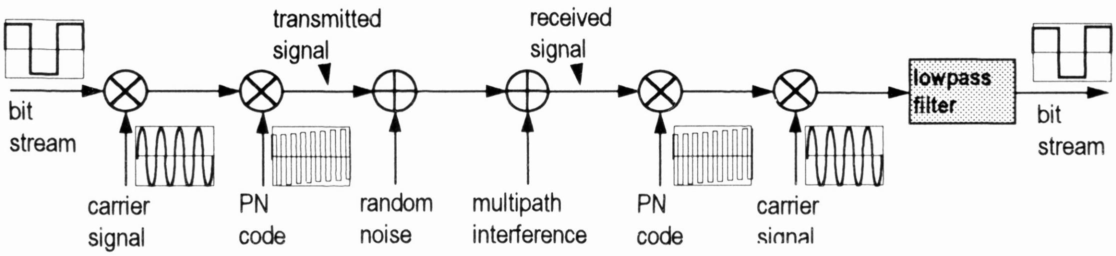

Figure 3: Diagram for standard spread spectrum model

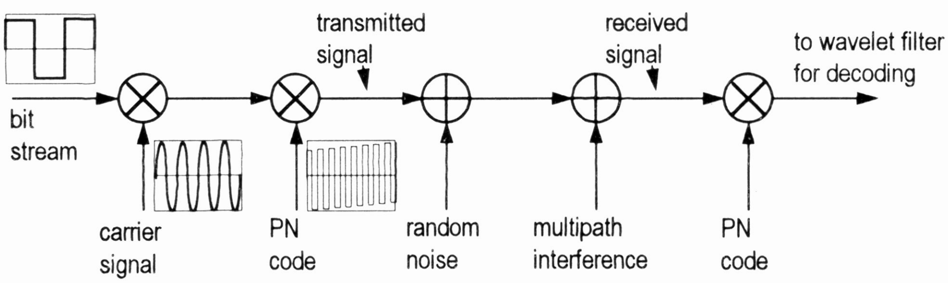

Figure 4: Diagram for the "wavelet" spread spectrum model

Since direct-sequence spread spectrum is a digital technology, any interesting results could directly apply to communication industries. These results would not apply to other forms of spread spectrum such as frequency hopping or time-hopping unless the form being used is a hybrid that includes DSSS techniques.

Process and Results: Test of Concept

Early simulations used binary phase-shift keying on a carrier located at 1000 Hz. The simulation used a bit rate of 100 bits per second (see Figure 5), with 10

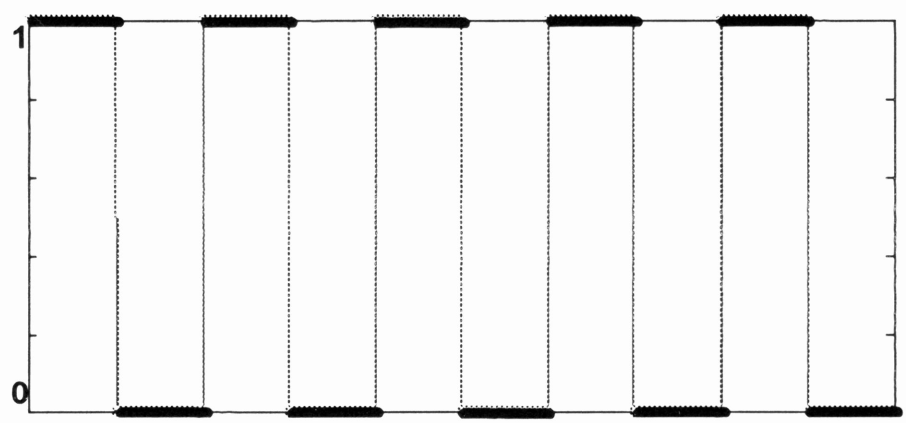

Figure 5: Bit stream for first simulations

pseudorandom chips per bit. Ten bits were used in the simulated signal, lasting 0.1 seconds for 5000 samples. All early simulations ran on the PC version of Matlab.

The pseudorandom bits used to spread the bandwidth of the BPSK signal were created with Matlab's RAND() command and rounded to one or zero (see Figure 6).

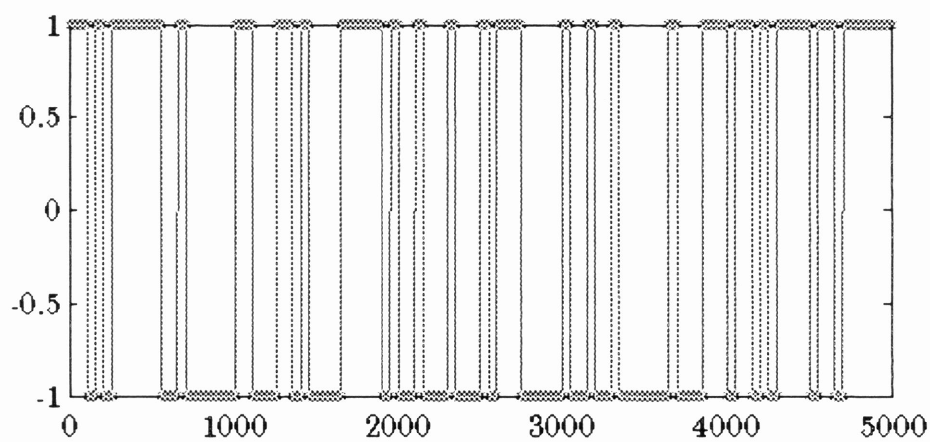

Figure 6: Plot of PN code bits used in early simulations

The bit and pseudorandom code streams were expanded, mixed with the sinusoidal carrier signal, and "transmitted" (see Figure 7).

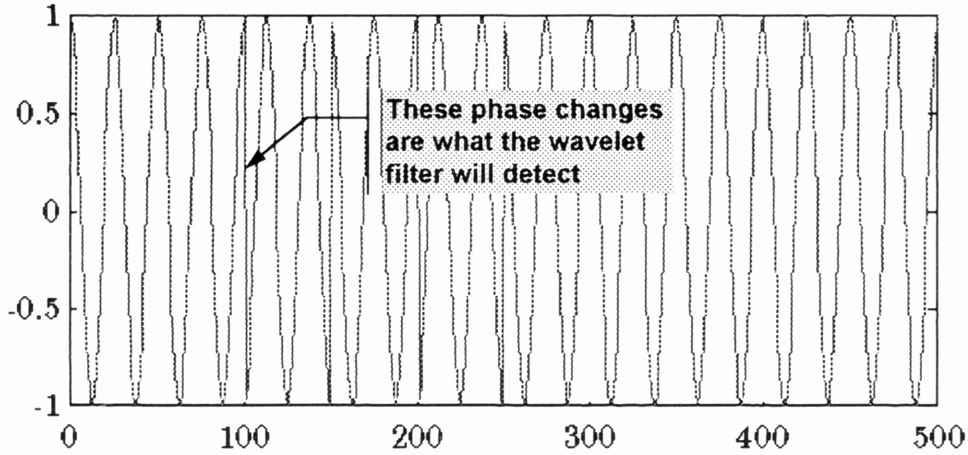

Figure 7: Time plot of the signal (0 to 0.01 sec)

The chip rate is consistent with guidelines set by the FCC for spread spectrum signals operating in the 900 MHz range at 1 Mbps. The early model is, of course, nowhere near 900 MHz; but since the signal is processed digitally, the signal used can be considered identical to one operating at higher frequencies over a shorter time frame. Later, direct comparisons with standard decoding were run at 900 MHz on a Sun SparcStation.

Early runs were attempted using Haar wavelet scaling function coefficients, lowpass and highpass filters of length 2. The Haar coefficients are simple (see Figure 8), but are useful in many signal processing applications, including video and

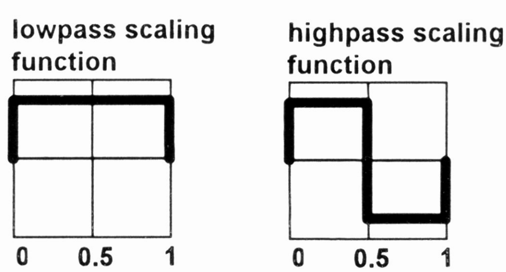

Figure 8: The Haar wavelet

audio processing. Unfortunately, when applied to the spread spectrum signal model, the Haar coefficients put some bit changes in different subbands, rendering easy analysis impossible.

A different, more complex wavelet was used to much greater success (see Figure 9) throughout the rest of the simulations.

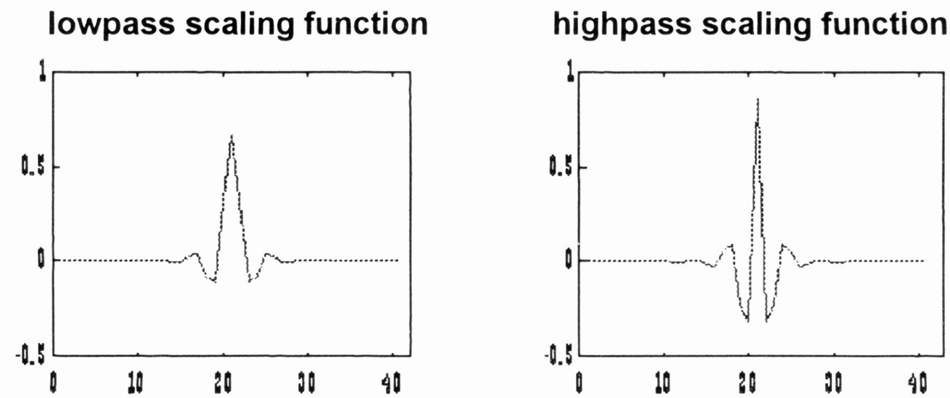

Figure 9: Wavelet used in simulations

The spread spectrum signal was first run through the wavelet filter without interference. The filter to detected the phase changes in the BPSK signal that indicate a +1 or -1 bit or series of bits. In other words, it worked in an ideal situation (see Figure 10).

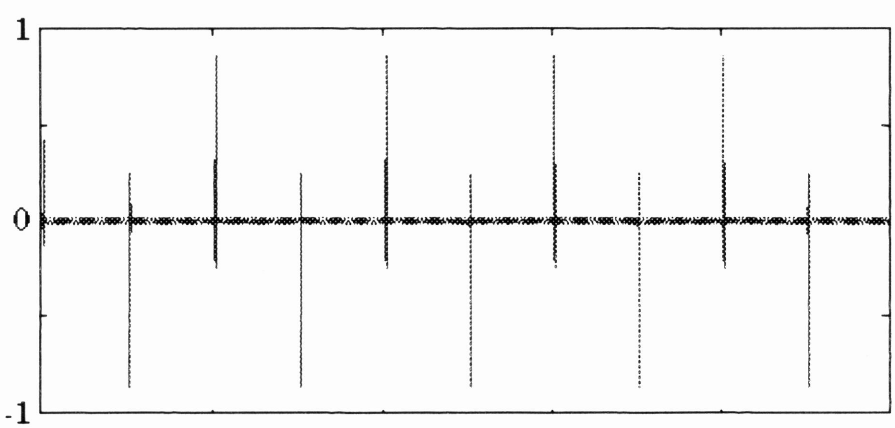

Figure 10: Plot of despread signal
(note that phase changes are in synch with bit changes in bit stream)

But how well? Next, additive white Gaussian noise was added to the signal. The signal was despread and decomposed. Since the result was unintelligible, the noise power was adjusted until the bit changes were visually detectable. At a signal to noise ratio (SNR) of 8 dB, the wavelet receiver detected the bit change blips acceptably (see Figure 11). The SNR required by the wavelet filter to recover the bit changes is higher than that required by standard recovery.

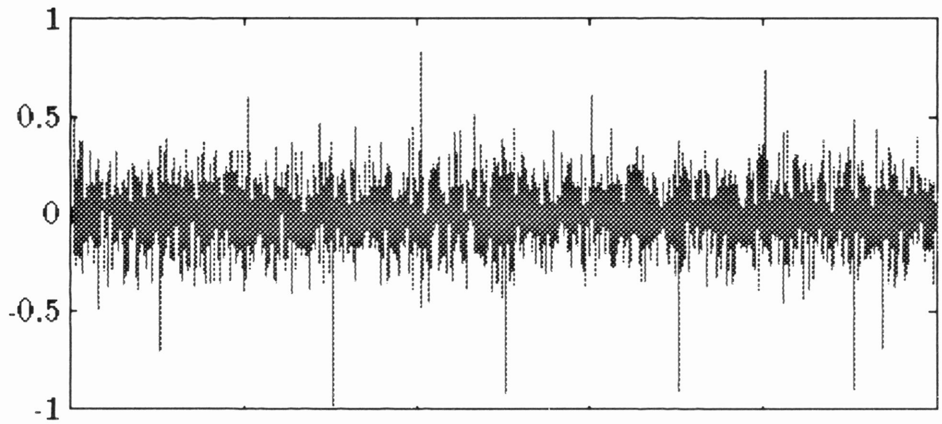

Figure 11: Plot of decomposition with 8 dB SNR

Next, multipath interference in the form of a reflected version of the original signal was added to and decoded with the original signal. The reflection was delayed by .002 seconds, enough time to cause the incoming signal to drop out occasionally; smaller delays had similar effects, but larger delays look like Gaussian noise thanks to the pseudorandom code. At a 3 dB SNR, the wavelet receiver decoded the bit changes well above the phase changes caused by the interfering signal (see Figure 12). Below a 1 dB SNR, the receiver's output was unintelligible.

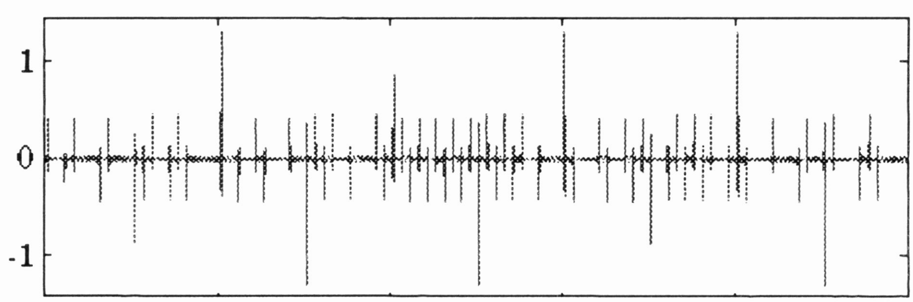

Figure 12: Plot of signal with multipath interference with 3 dB SNR

In the final test of the wavelet receiver, the intended signal is masked with both Gaussian noise and multipath interference. A reasonable (potentially detectable) pattern emerged when Gaussian noise with an 8 dB SNR and multipath interference with a 3 dB SNR were added to the transmitted signal. If the power of either interference element was raised much at all, though, the receiver produced unintelligible output.

The wavelet receiver is capable of successfully decoding a BPSK signal. To determine its usefulness beyond an academic curiosity, the wavelet receiver was compared with the standard ideal BPSK receiver.

Process and Results: Comparison with Standard Methods

Comparison tests were run on Matlab 4.2 on a Sun SparcStation 20. The simulated signal operated at 900 MHz, the frequency space allocated to the FCC for commercial spread spectrum devices. In early (and what would prove to be final) simulations, 20 random bits (see Figure 13) were run at a rate of 1 million bits per second (Mbps). To keep within FCC regulations, 10 pseudorandom chips per bit would spread the signal. The resulting signal was sampled at 10 GHz, combined with Gaussian noise with SNR 0 dB, and processed with the wavelet filter and the standard recovery method.

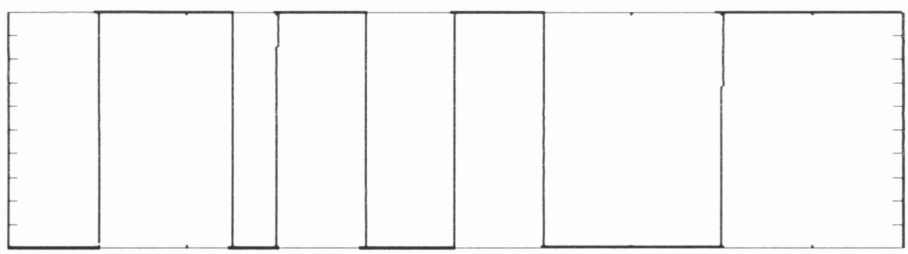

Figure 13: Plot of data bits used in sample (20 random bits)

The results were eye-opening. Even at an SNR of 0 dB, standard reconstruction accurately returned the bit information (see Figure 14).

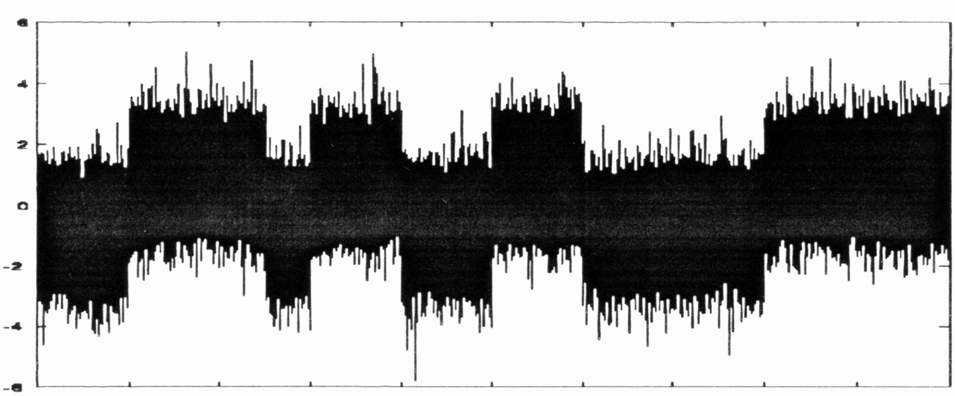

Figure 14: Plot of normal reconstruction of data bits at 0 dB SNR

The wavelet receiver did not fare as well with Gaussian noise at 0 dB. The noise level caused too many phase changes for the wavelet receiver to hope to sort from the bit phase changes (see Figure 15).

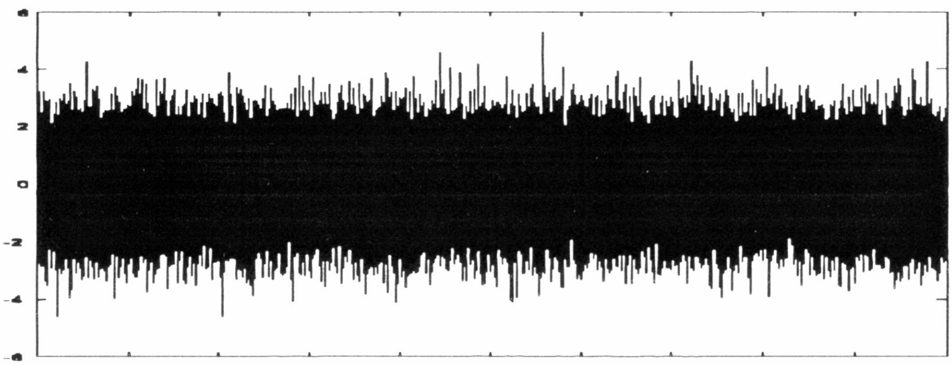

Figure 15: Plot of wavelet reconstruction of data bits at 0 dB SNR

Results for multipath interference were similar to the results for Gaussian noise interference. The standard reconstruction technique easily handled high levels of both kinds of noise, while the wavelet receiver required very low levels of noise and multipath interference in order to find the bit changes. In order to obtain a reasonable output for the wavelet receiver, noise had to be reduced to 10 dB below the transmitted signal (see Figure 16). On top of the noise power reduction, the multipath signal strength was reduced in order to obtain a reasonable output.

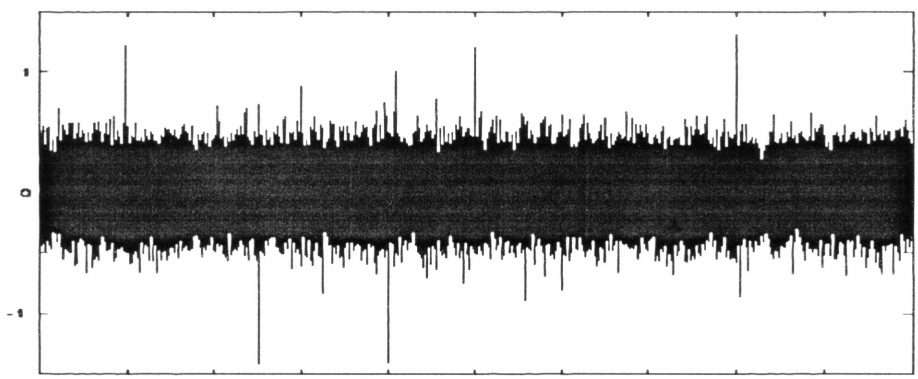

Figure 16: Plot of wavelet reconstruction of data bits
Gaussian noise at -10 dB, multipath at -3 dB

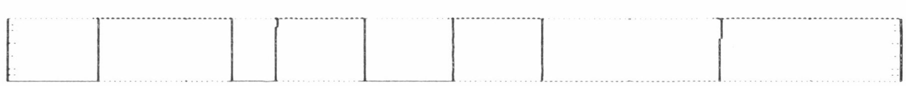

Figure 17: Plot of data bits (for reference)

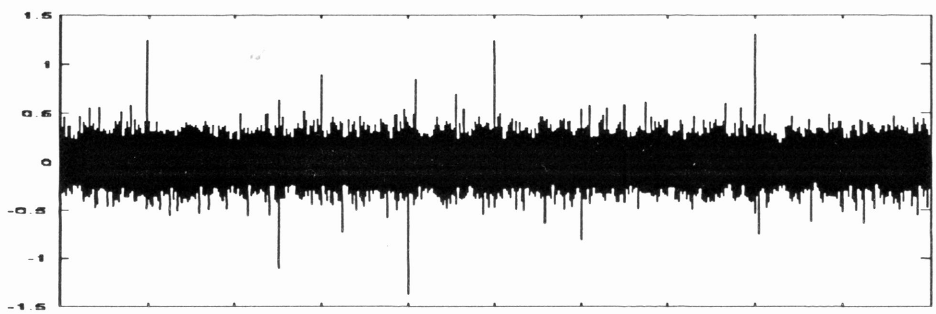

Figure 18: Plot of wavelet reconstruction of data bits
Gaussian noise at -10 dB, multipath at -8 dB

Conclusion

The wavelet receiver is technically capable of recovering bit information from a spread spectrum signal. Since it perceives the phase changes that correspond to bit changes as "blips" instead of DC output as in standard signal recovery, a more complex receiver must be used to detect and transmit the change properly and check carefully for errors due to an errant phase change.

No effort was made to this end, however, after roughly comparing the performance of the wavelet receiver with standard signal recovery. Standard recovery has a significant advantage in noise rejection. The wavelet processing cannot screen out the different scales involved in Gaussian noise or a multipath signal without eliminating the phase change that results from a bit change in the bit stream.