Evaluation of a Noncoherent UWB Demodulation Algorithm

Adam Skelton   *Wireless Communications*   *Dr. Shengli Fu*   *4/28/2008*

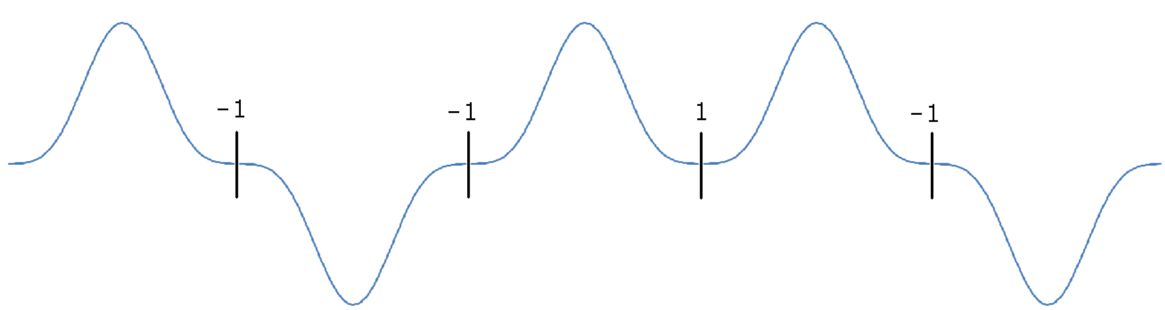
Introduction

Ultra-wideband is a promising new communications technology for high data rate, short range communications. There are two main variants of UWB. In OFDM-UWB, the bandwidth is broken into sections and the information is transmitted over several independent narrowband carriers. In impulse-based UWB, very short duration pulses are used to send the information in baseband. Unfortunately, because of the extremely short duration of pulses used, timing synchronization remains a significant problem for achieving maximum throughput and bit error rates in impulse-based UWB systems[1]. Several schemes have been put forth to deal with this problem, including a novel noncoherent demodulation algorithm that requires neither timing synchronization nor channel estimation for its operation[2]. This report is a presentation and analysis of that technique, along with some improvements.

Description of algorithm

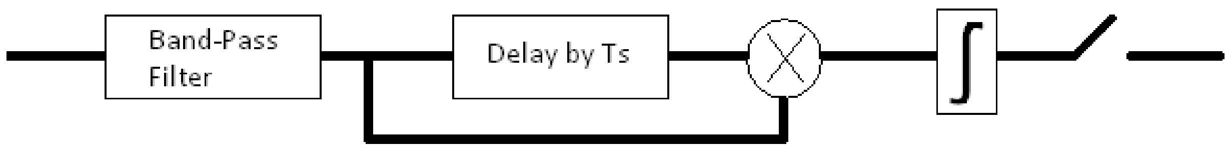
Assumptions

First of all, the algorithm requires that the data differentially encode a stream of pulses using binary signaling, so that a transition from a positive to a negative pulse or vice versa represents a 1, and a repetition of the previous pulse represents a 0, as shown in the figure below. Time-hopping spread spectrum and direct-sequence spread spectrum can be accommodated by making each symbol interval consist of several pulses, as long as all the symbols in a burst use the same code. Originally it was proposed to examine several modulation schemes under this algorithm, such as pulse amplitude modulation and pulse shape modulation, but they are inherently incompatible with this algorithm and will not be considered further. Second, the channel is assumed to be a tapped delay line, which is static over the length of one transmission burst, but is allowed to change between bursts. Last, the maximum channel delay is assumed to be small enough that there is no significant inter-symbol interference.

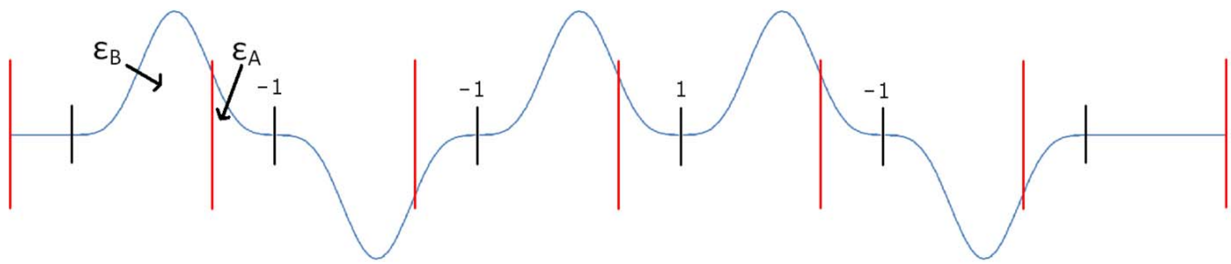


Symbol Capture

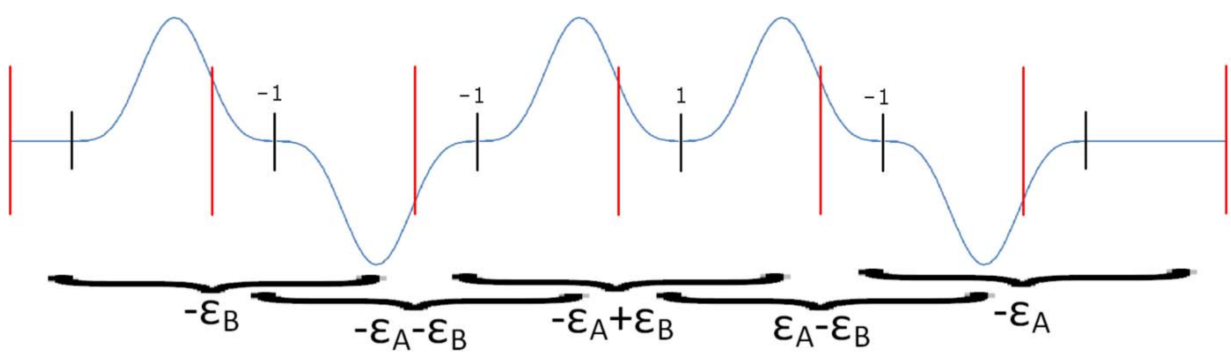
To begin with, the receiver, without regard for timing synchronization, captures symbol-length segments of the received waveform. Then, adjacent segments are correlated with each other, and the resulting scalars are passed on for decoding. Thus, the analog portion of the receiver is relatively simple and can be implemented as follows:



Once again, the capture is done without any attempt at timing synchronization. Therefore, the pulses will typically be split, as shown below:



Since every pulse is divided at the same place, the energy in each portion will be consistent across the transmission burst. We denote the energy of the right segment as ε_A, and the energy of the left segment as ε_B, as shown. Also, note that each pair of adjacent segments will encompass *two* pulse transitions, and thus two data bits. The first transition will determine the sign of ε_A in the result, and the second transition will determine the sign of ε_B. The following figure shows the correlation of adjacent segments, and how the results can be expressed in terms of the partial energies ε_A and ε_B.



Since there are four possibilities for the signs of ε_A and ε_B, there are four constellation points, as shown below:

Energy Estimation

In order to decode the received points, it is necessary to know ε_A and ε_B. Unfortunately, they are not known a-priori, and therefore must be estimated from the received points. This is the first step in the decoding process. The authors offer the following estimator, based on the fact that if you take the absolute value of the received points (without noise), they will take on only the two values $|\varepsilon_A-\varepsilon_B|$ and $\varepsilon_A+\varepsilon_B$ with equal probability:

$\max\{\varepsilon_A,\varepsilon_B\}$ = mean(abs(x))

$\min\{\varepsilon_A,\varepsilon_B\}$ = std(abs(x))

Then it is necessary to determine which of ε_A or ε_B is greater. The authors propose that the first two bits of all symbols be set to 1,-1, so that there will be a guaranteed $\varepsilon_A-\varepsilon_B$ in the result, which can be used to determine which is larger. This is not necessary, however, if the leading and trailing partial segments are used, as the absolute value of the first will be ε_B, and the absolute value of the last will be ε_A. Using this information can save those two bits per burst, reducing the average energy per bit.

Decoding

Decoding the received points can be done using several algorithms, and the choice of algorithm presents a tradeoff between bit error rate and computational complexity. The authors present two choices. The first is a maximum likelihood sequence decoder, and the second is a symbol-by-symbol conditional-maximum-likelihood decoder, which I have improved and offer as a third option. These algorithms will now be presented.

Maximum Likelihood Decoder

The authors note that the received points "can be viewed as the symbol-rate sampled output of an unknown first-order ISI channel whose impulse response taps are the partial channel energies ε_A and ε_B". The maximum likelihood sequence of data bits is that which minimizes the following equation for the entire sequence of bits.

$\{d(m-1),d(m)\} = \arg\min |d1*\varepsilon_A + d2*\varepsilon_B - x(m)|$

Therefore, maximum likelihood decoding can be done with an ML sequence detector, such as Viterbi's algorithm.

Conditional Maximum Likelihood Decoder

Notice once again that each symbol will appear in two consecutive outputs of the correlator. If either of ε_A or ε_B is much greater than the other, then each received value will consist almost entirely of only one of the data symbols, and the decoding can simply be done with a sign detector, as follows:

If $\varepsilon_B >> \varepsilon_A$, then use d(m) = sign{x(m)}

If $\varepsilon_A >> \varepsilon_B$, then use d(m) = sign{x(m+1)}

If the energies are roughly equal, then we must assume that the previous symbol was demodulated correctly and use a decision-feedback equalizer, as follows:

$d(m) = \arg\min |d(m-1)*\varepsilon_A + d*\varepsilon_B - x(m)|$

This can be simplified to the following form:

$d(m) = \text{sign}\{x(m) – d(m-1)*\varepsilon_A\}$

Deciding which one to use, either the simple sign detector or the sign detector with decision feedback, should be based on the one which minimizes the probability of error. The authors perform such an analysis and offer the following heuristic decision:

If $\varepsilon_B/\varepsilon_A < 0.5$, use the simple sign detector, otherwise use decision feedback.

Improved Conditional Maximum Likelihood Decoder

The need to estimate ε_A and ε_B requires that each transmission burst be collected in its entirety before demodulation can begin. For this reason, there is no penalty in performing the decoding *backwards*, from the end of the burst to the beginning. With this in mind, the decision feedback decoder can be used in all cases, and I offer the following algorithm:

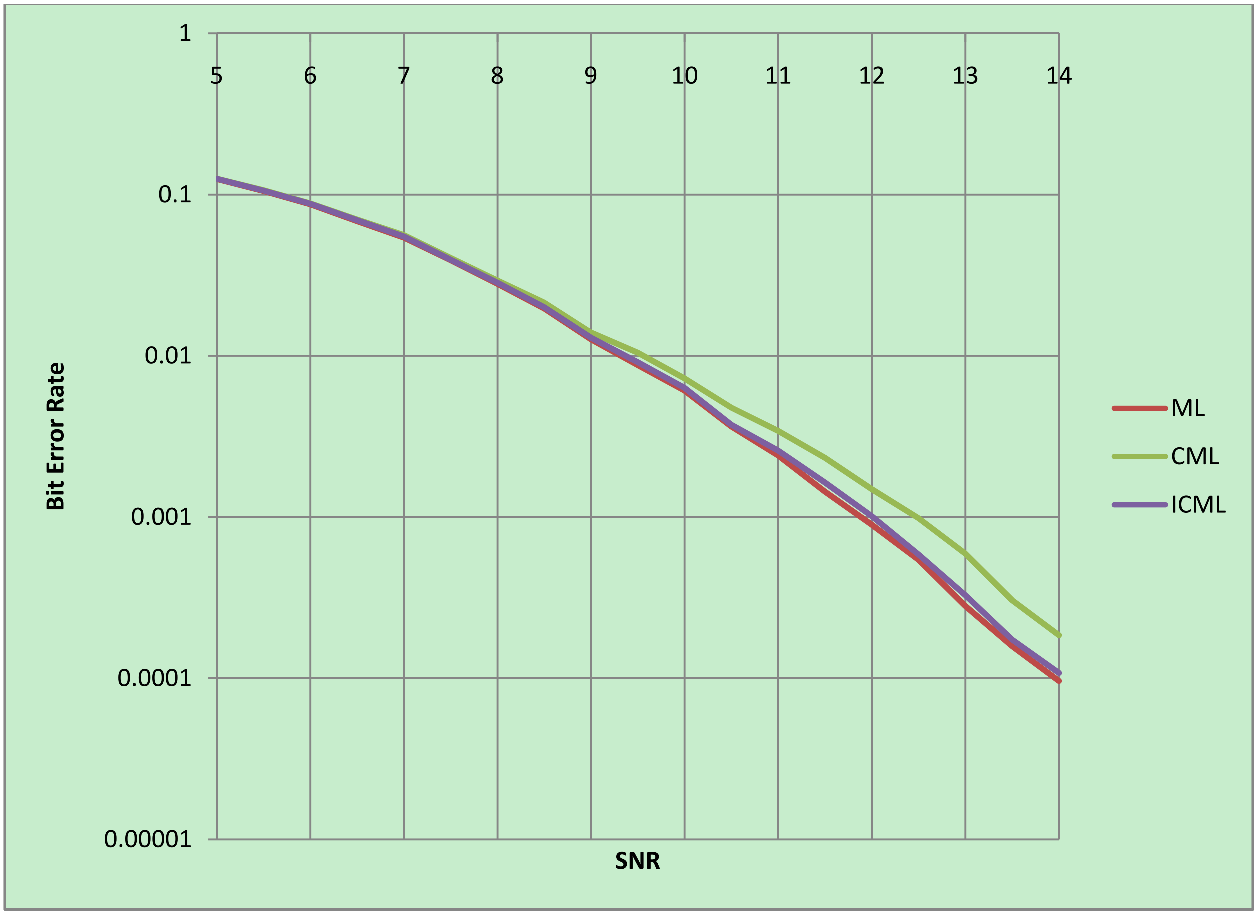
If $\varepsilon_B > \varepsilon_A$, then use the equation from above: $d(m) = \text{sign}\{x(m) – d(m-1)*\varepsilon_A\}$

If $\varepsilon_A > \varepsilon_B$, then decode the burst in *reverse* order, using: $d(m) = \text{sign}\{x(m+1) – d(m+1)*\varepsilon_B\}$

In this way, the feedback component always consists of the smaller of the two energies, reducing the likelihood that an incorrect decision will propagate, multiplying the errors.

Comparison of algorithms

The following simulation offers a performance comparison of the three decoding algorithms. Each transmission burst consists of 64 symbols with one pulse per symbol. The channel is multipath with three taps, although more could easily be accommodated. The timing offset is random and evenly distributed over the length of a symbol to represent all cases equally. To ensure a fair comparison, all three decoding algorithms are presented with the same correlator outputs.



As you can see, the improved algorithm has performance approaching that of the maximum likelihood case, at significantly reduced computational complexity!

Conclusions

The paper presents a novel and intuitive method of capturing and demodulating pulses of extremely short duration, in which timing synchronization is difficult or impossible. The receiver has the attractive property of having a simple analog front-end, consisting of only a band-pass filter, analog delay element, multiplier, and integrator. Each data symbol will contribute to two outputs of the correlator, opening the possibility for several different algorithms for demodulation. While the maximum likelihood demodulator using Viterbi's algorithm offers the lowest bit error rate, the improved CML demodulator has performance approaching that of the maximum likelihood case, at significantly reduced computational complexity, which is an important factor for low-cost implementations. Furthermore, in the case where the timing offset approaches zero, all algorithms reduce to a simple differential BPSK system. Therefore, this scheme could easily accommodate attempts at timing synchronization, but still function adequately in their absence. The issue of how to detect the beginning and end of a transmission burst under this system is not addressed, and could be a topic for further research.

References

1. *BER Sensitivity to Mistiming in Ultra-Wideband Impulse Radios - Part I: Nonrandom Channels.* **Tian, Zhi and Giannakis, Georgios B.** 4, s.l. : IEEE Transactions on Signal Processing, April 2005, IEEE, Vol. 53.

2. *Noncoherent Ultra-Wideband (De)Modulation.* **Yang, Liuqing, Giannakis, Georgios B. and Swami, Ananthram.** 4, s.l. : IEEE Transactions on Communications, April 2007, Vol. 55.