Acquisition for a Transmitted Reference UWB Receiver

Andreas Schranzhofer
Swiss Federal Institute of Technology
Computer Engineering and Networks Laboratory
Gloriastr. 35, CH-8092 Zürich
Switzerland
Email: schranzhofer@tik.ee.ethz.ch

Yiyin Wang
Delft University of Technology
Mekelweg 4, NL-2628 CD Delft
The Netherlands
Email: yiyin.wang@tudelft.nl

Alle-Jan van der Veen
Delft University of Technology
Mekelweg 4, NL-2628 CD Delft
The Netherlands
Email: a.j.vanderveen@tudelft.nl

***Abstract*— An acquisition algorithm for a transmitted reference (TR) ultra wideband (UWB) receiver is developed that uses the statistical properties of TR-UWB signals to distinguish between noise and data, achieve acquisition and ignores or decodes the incoming samples accordingly. Unknown parameters, like noise power and arriving time, are estimated and continuously updated once new data is acquired, enabling the receiver to adapt to changing environments. The acquisition is a detection problem with unknown arriving time and noise power. It is split into two steps, chip level and symbol level acquisition. Only segments that pass the first step are considered for the second step. This approach reduces the computational complexity and the implementations' resource demands.**

I. Introduction

In ultra wideband (UWB) communication systems the receiver has the responsibility to sense the presence of a data signal and to acquire the start point of the signal, see [1]. Previously implemented receiver algorithms for transmitted reference (TR) ultra wideband (UWB) focus on the optimization of the bit-error-rate (BER) performance, assuming perfect synchronization and/or known noise power, see [2][3]. These algorithms start processing at the exact beginning of a data burst and thus need not to distinguish between data and noise or interfering signals. Other implementations designed to achieve synchronization for UWB systems assume a noise free environment, see [4].

This paper focuses on methods and algorithms to detect the presence of data signals and to acquire the arriving time of a signal. Using these algorithms a receiver is proposed, and subsequently implemented on an FPGA, which is capable of estimating the noise power. The receiver was designed with emphasis on computational simplicity and low-power behavior, allowing the receiver to shut down parts of its hardware in order to save energy in case only noise is present. Furthermore the receiver updates its noise power estimate continuously in order to adapt to changing noise environments.

II. Data Model

A TR-UWB transmission system as initially proposed in [5] is based on a reference pulse and a data bearing pulse, where the data bearing pulse is being modulated with the chip code and the actual data. The transmission is organized in frames, each frame consisting of one reference/data pulse doublet. The transmitted signal at the antenna, for a single chip, is:

$$d_i(t) = g(t) + c_i \cdot s \cdot g(t - D) \tag{1}$$

where $g(t)$ represents the pulse shape, D the delay, c_i the chip code value, and s the actual data to be transmitted, see also [3] and [2]. The received signal is the convolution of the transmitted signal with the physical channel

$$r_i(t) = h(t) + c_i \cdot s \cdot h(t - D) + \omega(t) \tag{2}$$

where $h(t) = g(t) * h_p(t)$ is the convolution of the pulse $g(t)$ with the channel $h_p(t)$ and $\omega(t)$ is additive white Gaussian noise (AWGN). The signal is correlated with a delayed version of itself and integrated over the frame time $T_f = W$ resulting in

$$x(t) \quad = \int_{t-W}^{t} r(\tau) r(\tau - D) d\tau \tag{3}$$

Based on [2] the data model for the single-delay, single user, TR-UWB receiver after integration and sampling results as:

$$x[n] = A \cdot b[n] + \omega[n] \tag{4}$$

where $A > 0$ represents the amplitude which depends on the channel energy, $b[n]$ represents the coded data, and $\omega[n]$ represents the noise. $x[n]$ is the output of the analog integrator, sampled at a rate corresponding to the frame time T_f. To further simplify the data model, no inter-frame interference (IFI) is considered in the data model, see [3] for an extension.

The coded data $b[n]$ can be represented by the actual data symbol $\mathbf{s}$ that has to be transmitted and the code sequence $\mathbf{c}$, consisting of N_f samples, representing N_f frames in the single user case. Thus the data symbol is transmitted using N_f frames, where the ith frame is coded by a chip c_i. One single data symbol can be represented as:

$$\mathbf{b} = s_1 \mathbf{c} = \begin{bmatrix} s_1 c_1 & s_1 c_2 & \dots & s_1 c_{N_f} \end{bmatrix} \tag{5}$$

and a series of N_s symbols can be written as:

$$\mathbf{b} = \mathbf{s} \otimes \mathbf{c} = \begin{bmatrix} s_1 c_1 & \dots s_1 c_{N_f} & \dots & \dots & s_{N_s} c_1 & \dots & s_{N_s} c_{N_f} \end{bmatrix} \tag{6}$$

III. System Model

Based on the previously stated data model, a hypothesis test, see [6], is derived. The detection problem facilitates a hypothesis test to distinguish between noise and data, resulting in two hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$ respectively. The hypotheses allow to compute a test statistics and following that a threshold. Samples exceeding this threshold are considered data bearing samples, the others as noise. The Neyman-Pearson approach, maximizing the probability of detection P_D for a given probability of false alarm P_{FA}, is employed to compute a Generalized Likelihood Ratio Test (GLRT). The hypotheses are

$$\begin{array}{lll} \mathcal{H}_0 & : & x[n] = \omega[n], \quad \sigma^2 > 0 \\ \mathcal{H}_1 & : & x[n] = A \cdot b[n] + \omega[n], \quad \sigma^2 > 0 \end{array}$$

with $x[n]$ being the output of the analog integrator, A an unknown amplitude, $\omega[n]$ i.i.d. gaussian noise with unknown variance σ^2, and with known message $b[n]$ (e.g. a training sequence $\mathbf{p}$ spread by the known code $\mathbf{c}$ such that $\mathbf{b} = \mathbf{p} \otimes \mathbf{c}$).

Stacking the samples $x[n]$ into a vector $\mathbf{x}$ and applying this data model to the GLRT results in a likelihood ratio test as follows:

$$L(\mathbf{x}) = \frac{p(\mathbf{x}; \hat{A}, \hat{\sigma}^2, \mathcal{H}_1)}{p(\mathbf{x}; \hat{\sigma}^2, \mathcal{H}_0)} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\lessgtr}} \gamma$$

where γ is a threshold and $\hat{A}, \hat{\sigma^2}$ are parameter estimates under each of the hypotheses. The MLE for $\hat{A}$, for a deterministic signal, where the deterministic term is represented by the known chip code $\mathbf{c}$ can be computed following [6].

A. GLRT for TR-UWB

Assuming the noise variance to be known, estimated by a predefined initialization sequence introduced later in the paper, simplifies the GLRT and the likelihood function can be rewritten as:

$$L(\mathbf{x}) = \frac{p(\mathbf{x}; \hat{A}, \mathcal{H}_1)}{p(\mathbf{x}; \mathcal{H}_0)} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\lessgtr}} \gamma$$

which yields a rather simple detector of the form:

$$T(\mathbf{x}) = \sum_{n=0}^{N-1} x[n] b[n] > \gamma \tag{7}$$

for the statistic and

$$\gamma = \sqrt{\sigma^2 \sum_{n=0}^{N-1} b^2[n]} \cdot Q^{-1}\left(\frac{P_{FA}}{2}\right) \tag{8}$$

to compute the threshold, see [6], where N is the length of the known deterministic sequence specified by $\mathbf{b}$ and σ^2 is the noise variance.

This allows to detect a given sample sequence in an input sample stream. We define this given sample stream as a known header $\mathbf{b}$, which precedes each data burst (packet). The header consists of N_s Symbols, each made up of N_f samples, resulting in a known sequence of $N = N_f \cdot N_s$. Once this header is detected correctly, acquisition is achieved.

In order to reduce the complexity and resource requirements of the algorithm the acquisition is split into two subsequently performed detection problems. The first step of the detector applies the test statistics, using $N = N_f$ and $\mathbf{c}$ as the known code sequence to the incoming data samples. Applying the threshold to the test statistics results in a partitioning of the incoming data vector into segments that supposedly contain data samples and segments that contain noise-only sample.

The first detection step represents a rough acquisition on chip level. The detection is based solely on the known code sequence $\mathbf{c}$. Only samples that are classified as *data in noise* in this detection step are taken into consideration in the second detection step.

The second detector is basically the same detection problem as the first one, as a known sequence of unknown amplitude has to be found. The known sequence at this stage is the header sequence $\mathbf{p}$ of length N_s and the input to the detector is the output of the first detector, such that:

$$y[k] = \sum_{n=kN_f}^{kN_f+N_f-1} x[n] c[n] \tag{9}$$

Stacking scalars $y[k]$ for $k = 1 \ldots \text{length}(x) - N_f$ into a vector results in $\mathbf{y}$, consequently the second detector works on symbol level. Only segments that were classified as *data in noise* segments by the first detection step are taken into consideration for the second detector.

The test statistics $T(\mathbf{y})$ for the second detection step can be formulated as the correlation of the vector $\mathbf{y}$ with the known header sequence $\mathbf{p}$.

$$T(\mathbf{y}) = \sum_{k=0}^{N_s-1} y[k] p[k] > \gamma' \tag{10}$$

The signal $\mathbf{y}$ will be normal distributed in the noise-only case, with a variance of $\sigma_p^2 = N_f \cdot \sigma^2$, and a mean of $\mu_p = N_f \cdot \mu$, where μ is the mean of the input signal $\mathbf{x}$ to the first detector, following [7]. Assuming a probability of false alarm, P'_{FA}, a threshold can be computed as:

$$\gamma' = \sqrt{N_f \cdot \sigma^2 \sum_{n=0}^{N_s-1} p^2[n]} \cdot Q^{-1}\left(\frac{P'_{FA}}{2}\right) \tag{11}$$

Assuming $P'_{FA} = P_{FA}$ and $N_s = N_f$ so that the terms $\sigma^2 \sum_{n=0}^{N_s-1} p^2[n]$ and $\sigma^2 \sum_{n=0}^{N_f-1} c^2[n]$ are equal, the equation can be simplified. Comparing Equation (8) and (11), considering the previously stated assumptions, results in:

$$\gamma' = \sqrt{N_f} \cdot \gamma \tag{12}$$

The overall probabilities of false alarm and detection result as the multiplication of the first and second steps' individual probabilities, thus degrading the overall P_D performance. This represents the tradeoff between resource requirements and performance.

B. Noise power estimation

As the noise power is unknown, we introduce an iterative approach to estimate it. This relaxes the condition of a defined data free initial phase to measure the noise variance and additionally introduces adaptivity to the system.

In a first iteration an initial variance σ^2 out of the input vector $\mathbf{x}$ is computed. This variance is used to compute an initial threshold γ. Applying the previously described acquisition algorithm to the input vector, with a threshold based on this measured σ^2, results in an initial segmentation of the input vector into data and noise only segments.

The initially computed variance is calculated using all samples in the input vector $\mathbf{x}$, thus also considering the data bearing samples, resulting in an elevated threshold. This results in some data blocks being undetected, especially in low SNR environments.

The next iteration of the algorithm recomputes the noise variance by considering only those segments of the input vector $\mathbf{x}$ that were classified as noise-only in the previous step. This results in a new variance σ^2. Updating the threshold and reprocessing the input vector results in a new segmentation, that is supposedly closer to the correct one.

The iterative detection process stops once the difference between the newly and previously computed threshold does not exceed a specified value. Using the last configuration, data can be decoded by reading the sign of the first detection steps' output, vector $\mathbf{y}$, at every N_f-th position.

IV. Simulation

In our simulations a symbol was made up of $N_f = 8$ chips and the header was defined as a sequence of $N_s = 8$ symbols followed by a data burst, the payload, of 20 symbols. The samples created that way were stacked into a data vector. We call it a data block. The transmission vector is created in a way, that there are sequences of samples in between subsequent data blocks that represent noise only.

The duty cycle is defined as the number of data bearing samples in relation to the total number of samples in the transmission vector. It is varied by increasing or decreasing the number of subsequent data blocks, each consisting of 28 symbols, in a transmission vector, resulting in a increased or decreased duty cycle respectively. Multiple subsequent data bursts, intersected by noise only segments, tests the receivers' ability to acquire multiple subsequent transmissions.

These transmission vectors were used as input vectors to the proposed system to test the BER and noise estimation performance of the proposed receiver.

A. BER

Data transmissions are simulated for different E_b/N_0 and P_{FA}, where undetected transmissions are taken into account with $BER = 0.5$. The transmissions are simulated using Monte-Carlo runs, in order to reduce the influence of adverse conditions.

In Fig. 1 the BER performances degrades for high E_b/N_0 transmissions (9 - 16 dB) and high probabilities of false alarm P_{FA} (starting at 10^{-2}). With increasing duty cycle, the effect decreases in absolute terms, but compared to simulations with lower P_{FA} the BER performance is still inferior.

For low E_b/N_0 levels the BER degrades with decreasing P_{FA}, starting at 11 dB in Fig. 1 and at 13 dB in Fig. 2. A low duty cycle and a P_{FA} in the magnitude of 10^{-3} results in the best overall BER performance for different E_b/N_0.

The rising BER at high probabilities of false alarm can be explained by the additional samples that are considered as *data in noise*, as a high probability of false alarm results in a low threshold, and thus noise samples are more likely to exceed it and to be classified as data. This results in additional data in the incoming data vectors and hence higher BER, when comparing them to the expected data vectors in the simulations. The effect can be compensated by suitable protocols and data coding.

The inferior BER at high E_b/N_0 levels for higher duty cycles, in Fig. 2, is due to the fact that the estimation of the noise power is less accurate, as less data samples are available for its estimation that can clearly be classified as noise-only. This results in an inaccurate threshold and misdetection.

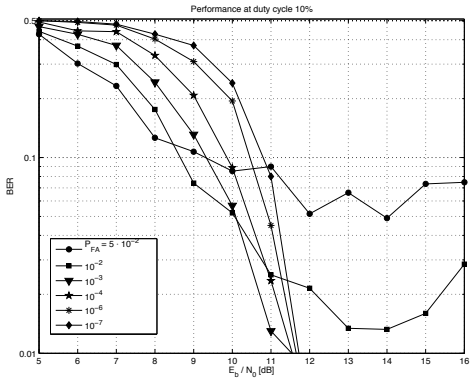

Fig. 1. BER at duty cycle = 10%

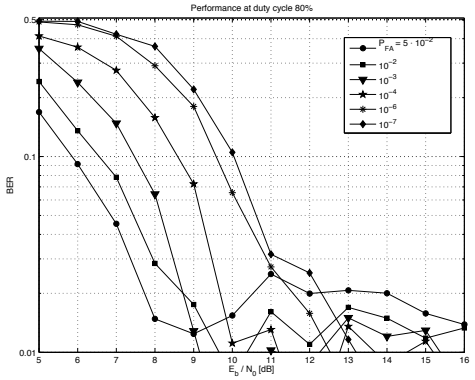

Fig. 2. BER at duty cycle = 80%

B. Noise power estimation

Fig. 3 shows the performance of the noise estimation algorithm proposed in Section III-B. For different levels of P_{FA} and different duty cycles simulations are done, and the number of iterations until the estimated variance converges to the true variance are counted. The performance of the noise estimation depends mostly on the signal to noise ratio E_b/N_0, as the noise level is the major factor when computing the threshold. In a low noise environment, even if the duty cycle is very high, the *data in noise* and *noise only* segments can easily be distinguished, and hence in the second or third iteration the estimated noise level converges to the true noise level. In high noise environments, signal and noise cannot be discriminated any more, and consequently a percentage of transmissions do not converge at all.

To visualize the performance of the iterative noise estimation approach, a specific transmission is considered as successfully decoded when the bit-error rate (BER) differs from 0.5. A BER of 0.5 means that the signal was not decoded at all, and thus that the noise estimation and threshold calculation did not lead to a successful signal segmentation. Figure 3 shows that the noise estimation works well for a high P_{FA}, which is coherent as a high P_{FA} directly results in a high P_D. It is also obvious that the higher the duty cycle, the lower the probability that the iterative approach results in a successful segmentation. The higher the duty cycle in the signal, the smaller the data set that is available for the actual noise estimation, and so the accuracy of the calculation suffers. A higher duty cycle triggers a higher initial estimate, which results in a higher initial threshold which increases the probability to miss a data burst.

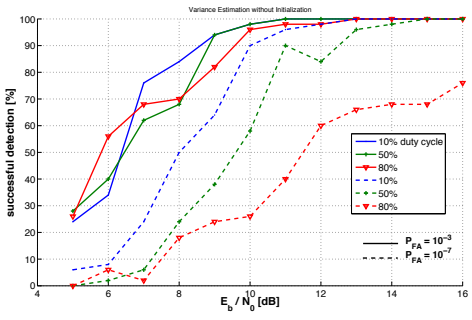

Fig. 3. Noise estimation performance

V. Implementation

The algorithm was implemented on a Xilinx Spartan 3 FPGA and the UWB Communication Platform developed at Delft, University of Technology. The current system implementation uses 1575 (3%) LUTs[1]. Additionally to LUTs, the Spartan 3 FPGA includes hardware implemented multipliers. Due to the very frequent use of multipliers in our system, this feature contributes strongly to the performance. The FPGA has a total of 40 hardware implemented 18x18 bit multipliers, of which we use 8 (20%).

[1]lookup table

VI. Conclusion

The structure of the proposed system reduces the complexity of the detector from $O(N_f \cdot N_s)$, where N_f represents the length of the code sequence $\mathbf{c}$ and N_s represents the number of header symbols used in the header synchronization sequence $\mathbf{p}$ to $O(N_f + N_s)$. The detection of a long synchronization sequence, with $N_f \cdot N_s$ known synchronization samples, is split into two detection problems of shorter sequences, each based on different signal hierarchies (chip level vs. symbol level). Additionally this structure reduces the number of multiplications from $N_f \cdot N_s$ to $N_f + N_s$ and the number of adders required to compute the correlation output from $(N_f \cdot N_s) - 1$ to $(N_f - 1) + (N_s - 1)$, which supports the energy efficiency of the system. Separating the detection problem into two, allows the hardware implementation to shut down the second detector in case the first one detects only noise, thereby reducing the power consumption.

The proposed system relaxes some assumptions that had to be made in previous implementations, such as the knowledge of the variance and the arrival time. Other assumptions are introduced, such as the known header sequence, that has to be transmitted preceding each data block, enabling the system to find the synchronization point. Adapting to changing environmental conditions, using a subsequently updated variance to calculate the threshold relaxes the condition of an absolutely data free initialization sequence and allows the system to work in environments with varying interferences.

Both, the power consumption reduction and the adaptivity to changing noise environments address crucial matters in UWB Systems in particular and in mobile, wireless Systems in general.

References

[1] S. Aedudodla, S. Vijayakumaran, and T. Wong, "Timing acquisition for transmitted reference DS-UWB signals," in *Military Communications Conference, 2005. MILCOM 2005. IEEE*, 17-20 Oct. 2005, pp. 3087–3093Vol.5.

[2] Q. Dang, A. Trindade, A.-J. van der Veen, and G. Leus, "Signal model and receiver algorithms for a transmit-reference ultra-wideband communication system," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 4, pp. 773–779, April 2006.

[3] A. Trindade, Q. H. Dang, and A.-J. van der Veen, "Signal processing model for a transmit-reference UWB wireless communication system," in *Ultra Wideband Systems and Technologies, 2003 IEEE Conference on*, 16-19 Nov. 2003, pp. 270–274.

[4] Y. Wang, R. van Leuken, and A.-J. van der Veen, "Design of a practical scheme for ultra wideband communication," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, 21-24 May 2006, p. 4pp.

[5] R. Hoctor and H. Tomlinson, "Delay-hopped transmitted-reference RF communications," in *Ultra Wideband Systems and Technologies, 2002. Digest of Papers. 2002 IEEE Conference on*, 21-23 May 2002, pp. 265–269.

[6] S. M. Kay, *Fundamentals of Statistical Signal Processing - Detection Theory*, ser. Prentice Hall signal processing series. Prentice Hall PTR, 1993, vol. II.

[7] R. M. Gray and L. D. Davisson, *An Introduction to Statistical Signal Processing*. Stanford University, University of Maryland: Cambridge University Press, 2004.