Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2006, Article ID 37952, Pages 1–14
DOI 10.1155/WCN/2006/37952

Blind Synchronization in Asynchronous UWB Networks Based on the Transmit-Reference Scheme

Relja Djapic,[1] Geert Leus,[2] Alle-Jan van der Veen,[2] and António Trindade[2]

[1] *TNO-ICT, Brassersplein 2, 2612 CT Delft, The Netherlands*
[2] *Department of Electrical Engineering, Delft Institute of Microelectronics and Submicron-technology (DIMES), Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands*

Received 15 September 2005; Revised 13 December 2006; Accepted 13 December 2006

Ultra-wideband (UWB) wireless communication systems are based on the transmission of extremely narrow pulses, with a duration inferior to a nanosecond. The application of transmit reference (TR) to UWB systems allows to side-step channel estimation at the receiver, with a tradeoff of the effective transmission bandwidth, which is reduced by the usage of a reference pulse. Similar to CDMA systems, different users can share the same available bandwidth by means of different spreading codes. This allows the receiver to separate users, and to recover the timing information of the transmitted data packets. The nature of UWB transmissions—short, burst-like packets—requires a fast synchronization algorithm, that can accommodate several asynchronous users. Exploiting the fact that a shift in time corresponds to a phase rotation in the frequency domain, a blind and computationally effcient synchronization algorithm that takes advantage of the shift invariance structure in the frequency domain is proposed in this paper. Integer and fractional delay estimations are considered, along with a subsequent symbol estimation step. This results in a collision-avoiding multiuser algorithm, readily applicable to a fast acquisition procedure in a UWB ad hoc network.

Copyright © 2006 Relja Djapic et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Impulse radio (IR) ultra-wideband (UWB), further on simply called UWB, has recently been proposed as a system that can provide high data rate communications (up to 100 Mbit/s) on short distances (order of 10 m). Exploitation of the bandwidth of at least 500 MHz induces a great number of issues in the transceiver design and signal processing (see [1] for a recent overview of UWB signal processing and communications challenges).

The classical transceiver schemes use the data signal in order to modulate a carrier, that is, the spectrum of the data sequence is shifted from the baseband to a higher carrier frequency. Reversely, in UWB systems a carrier-less approach is employed. The information is conveyed by modulation of temporal pulses of extremely short duration—less than a nanosecond. As a consequence, the spectrum of the UWB signal is covering an extremely large frequency band. To allow for coexistence with already deployed narrowband communication systems such as GSM, GPS, and WLAN, the energy of the emitted UWB pulses needs to be lowered to the noise level. In addition, generation of the pulses is an extremely low-complexity and low-power operation [2] and therefore facilitates the accomplishment of low-cost transmitter devices. All these features make impulse radio attractive for high data rate, short distance, and multiuser personal area networks (PAN).

Propagation of a temporally narrow pulse (in the order of a nanosecond), also known as a *monocycle*, results in a channel impulse response that is much longer than the duration of the pulse itself and that has a large number of delay taps [3]. As the channel resolution is inversely proportional to the bandwidth of the signal, differences in path delays or path lengths of 1 ns and 30 cm, respectively, can be resolved [4, 5]. The resulting low probability of multipath fading permits a larger amount of transmitted energy to be collected at the receiver.

The large bandwidth of UWB signals allows to accommodate multiple simultaneous users. The most common modulation scheme that facilitates coexistence of multiple users in UWB systems is time-hopping pulse position modulation (TH-PPM) [6]. A monocycle is considered to be a part of a longer time interval defined as a *frame*. To avoid collisions due to multiple access, each user is assigned a random time-hopping code and shifts his monocycles within frames according to it.

The correlation receiver is considered to be the optimal receiver if the TH-PPM modulation scheme is used. Initially,

the channel impulse response has to be estimated and convolved with the known user code to obtain the template applied in the correlation process with the received signal. Performing an exhaustive search over different delays, averaging over several data symbols and finally searching for the maximum of the recollected energy function provide the estimate of the packet offset. Note that for this kind of receiver the knowledge of the channel is required. Some authors propose the implementation of a RAKE receiver to obtain the estimate of the channel but taking into account the current state of technology, we consider this approach unsuitable for implementation in a low-cost UWB transceiver because of the high computational complexity and high sampling rates.

A way to avoid channel estimation is in the implementation of a transmit-reference transmission scheme introduced already in [7] and revived by Hoctor and Tomlinson [8, 9]. The idea consists of the transmission of two pulses (*doublet*) one after another, where the first pulse is used as the reference for the second pulse which is modulated with data (polarity of the pulse corresponds to data symbols $\{-1, +1\}$). Both pulses undergo the same multipath channel. At the receiver, the reference pulse is delayed and correlated with the data pulse allowing to recollect the energy which was spread by the channel. The effective data rate is thus reduced by 50% but the receiver sampling rate and complexity are highly reduced because the correlation and integration steps are done in the analog part of the receiver.

Taking into account the fact that the UWB signal is transmitted at low-power level (comparable to the noise level), further performance improvements by suppressing the noise are possible as proposed in [10]. In [11], an advanced noiseless data model for a specific TR-UWB receiver was derived, taking into account that the channel has a long impulse response. The extension of this data model to the noisy case is presented in [12].

The work in [11, 12] represents the starting point for the present paper. In contrast to [11, 12], we consider finite data packets with an unknown time offset. The particular structure of the TR-UWB scheme only requires a synchronization at the chip-level, which is an easier problem than synchronization for more traditional pulse-based UWB schemes where the starting point of a very narrow pulse has to be found. Hence, the blind synchronization problem considered in this paper is to find the known user code at an unknown offset, which is an extension of a similar problem considered in CDMA, now for a more complicated data model. In particular, we propose an extension of the blind channel estimation algorithm for CDMA proposed by Torlak and Xu [13]. The received data samples are stacked in a matrix such that a shifted version of the user specific block code is in its column span. Subsequently, we exploit the fact that a shift in time corresponds to a phase rotation in the frequency domain. Finally, a MUSIC-like search for a shift invariant vector in the signal subspace provides a high resolution delay estimate.

Notation

^{T} denotes the matrix transpose, ^{H} the matrix complex conjugate transpose, $\dagger$ the matrix pseudoinverse (Moore-Penrose inverse). $\mathbf{I}$ (or $\mathbf{I}_p$) is the $(p \times p)$ identity matrix. $\mathbf{0}$ (or $\mathbf{0}_{p\times q}$) and $\mathbf{1}$ (or $\mathbf{1}_{p\times q}$) are $(p \times q)$ matrices for which all entries are equal to 0 and 1, respectively. For a vector, diag($\mathbf{v}$) is a diagonal matrix with the entries of $\mathbf{v}$ on the diagonal. $\otimes$ is the Kronecker product. vec($\mathbf{A}$) is a stacking of the columns of matrix $\mathbf{A}$ into a vector.

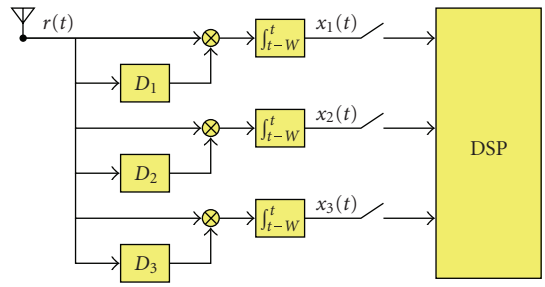


Figure 1: The structure of the autocorrelation receiver.

2. SINGLE USER DATA MODEL

2.1. Single doublet

In the TR-UWB scheme presented in [11, 14], pulses $g(t)$ are transmitted in pairs (doublets) which are mutually separated by a delay D_i, $i = 1, 2, \ldots, M$, where M represents the total number of delays used. Besides, we assume that $D_1 < D_2 < \cdots < D_M$. The first pulse is fixed and represents the reference, whereas the second one is modulated with the data. In the sequel, we first describe the data model for the synchronous single doublet transmission. In addition, we outline the parameters that arise as a result of the deployment of the specific correlation receiver derived in [11, 12, 15].

Consider the transmission of a single doublet $d(t)$,

$$d(t) = g(t) + c \cdot s \cdot g(t - D_i), \tag{1}$$

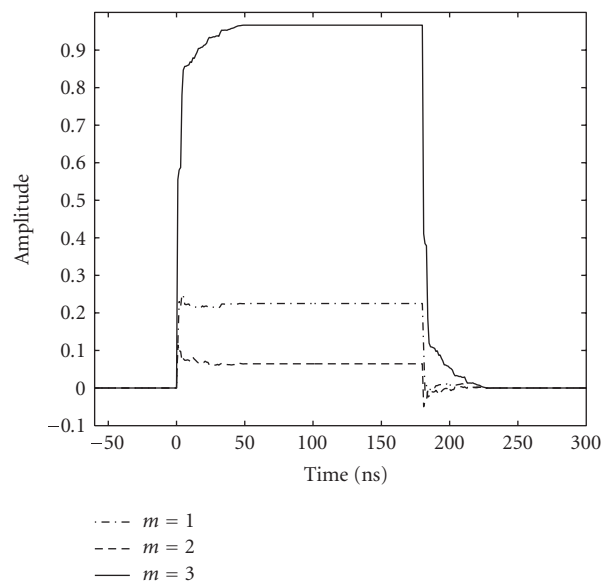
where $g(t)$ represents a reference pulse while $c \cdot s \cdot g(t - D_i)$ is a data modulated pulse, with scalars $c = \{\pm 1\}$ and $s = \{\pm 1\}$ representing a polarity code and a data symbol, respectively. Accordingly, the sign of the data modulated pulse is defined by the value of $c \cdot s = \{\pm 1\}$. We assume that a doublet is placed within a frame of length T_d and that a constraint

$$T_d \geq T_h + 2 \max\{D_i\} = T_h + 2D_M \tag{2}$$

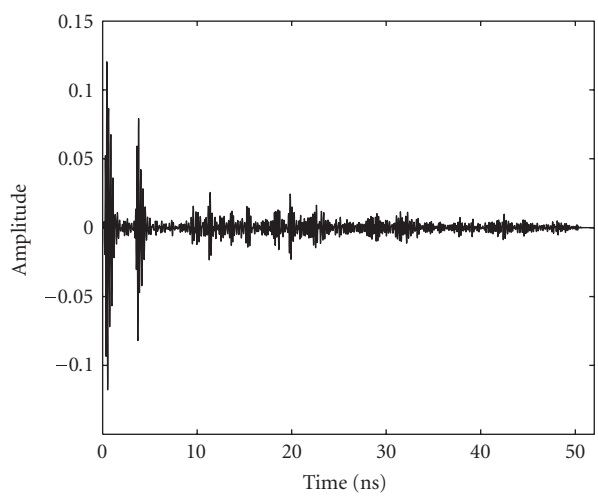
holds, where T_h stands for the duration of the channel impulse response. This condition implies that the pulses of a doublet affected by the channel fade out completely within a single frame after correlation (see Figure 1). In this manner, the existence of interframe interference in case of multiple doublet transmission is prevented (see Figure 4). After propagation through a long convolutive channel, the signal at the output of the receiver antenna can be written as

$$r(t) = h(t) + c \cdot s \cdot h(t - D_i), \tag{3}$$

where $h(t) = g(t) \star h_p(t)$ represents the overall channel impulse response obtained as the convolution of the transmitted pulse $g(t)$ and the channel impulse response $h_p(t)$. Note



(a)



(b)

Figure 2: (a) The signal at each of the $M = 3$ integrator outputs. (b) A measured UWB channel impulse response in a typical university building used to generate (a).

that the latter comprises the effects of the transmit and receive antennas together with the wireless propagation channel.

Since both pulses of a doublet undergo the same channel, one is used as a "matched filter" for the other one at the receiver. This is the principle behind the autocorrelation receiver depicted in Figure 1 [11, 14]. The received data $r(t)$ is delayed over all possible delays D_m, $m = 1, 2, \ldots, M$, and correlated with the original nondelayed signal. Finally, integration with a sliding window of width W at the mth receiver branch yields

$$x_m(t) = \int_{t-W}^{t} r(\tau) r(\tau - D_m) d\tau. \tag{4}$$

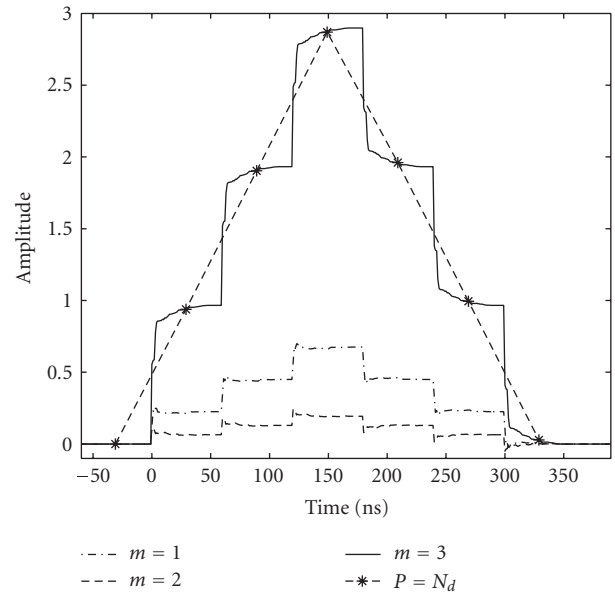


Figure 3: The signal at the output of the 1st, 2nd, and 3rd receiver branches. A single chip transmission is considered that comprises $N_d = 3$ doublets. The width of the sliding window integrator is $W = T_c = 3T_d$.

Let us now introduce the channel correlation function

$$\psi(t, \Delta) = \int_{t-W}^{t} h(\tau) h(\tau - \Delta) d\tau. \tag{5}$$

Assuming $W \geq T_d > T_h$, we can write $\psi(t, \Delta) = b(t)\rho(\Delta)$. Here, $\rho(\Delta) = \int_0^\infty h(\tau)h(\tau - \Delta)d\tau$ depicts the energy recollected in the correlation process. It is maximized for $\Delta = 0$ and is, in general, nonzero for $\Delta \neq 0$ (see [15]). Further, $b(t)$ has a brick shape and can be written as

$$b(t) = \begin{cases} 1, & T_h < t \leq W, \\ 0, & t < 0 \quad \text{or} \quad t > T_h + W, \\ \psi(t, \Delta)/\rho(\Delta), & 0 \leq t < T_h \quad \text{or} \quad W < t \leq W + T_h. \end{cases} \tag{6}$$

Note that in the regions $0 \leq t < T_h$ and $W < t \leq W + T_h$, $b(t)$ depends on the particular channel realization but it is approximated by a linear rising and decaying slope, respectively.

If we now assume that $W \gg D_M$, the output of the mth integrator (4), in case delay D_i is used at the transmitter and D_m at the receiver, becomes [15]

$$\begin{aligned} x_m(t) &= b(t)\{2\rho(D_m) + c \cdot s \cdot [\rho(D_m - D_i) + \rho(D_m + D_i)]\} \\ &= b(t)(\alpha_{m,i} \cdot c \cdot s + \beta_m), \end{aligned} \tag{7}$$

where $\alpha_{m,i} = \rho(D_m - D_i) + \rho(D_m + D_i)$ and $\beta_m = 2\rho(D_m)$ represent the unknown channel correlation coefficients that are real numbers corresponding to a gain and a DC offset, respectively [15]. While the gain depends on both the

transmitter delay D_i and the receiver delay D_m, the DC offset only depends on the receiver delay D_m. Moreover, both $\alpha_{m,i}$ and β_m depend on the particular correlation properties of the channel, as indicated in [11]. Note that although $\alpha_{m,i}$ is generally maximal if $m = i$, some residual information remains when $m \neq i$, as an effect of the channel correlation.

Figure 2(a) depicts the response of the system to a single transmitted doublet for the channel impulse response of Figure 2(b). The latter is obtained from a measurement campaign performed in a typical university building [16]. The spacing between the pulses at the transmit side is chosen to be $D_3 = 2.1$ ns, the data symbol is $s = +1$, and the polarity code is $c = +1$. At the receiver side three delay branches $m = 1, 2, 3$ are deployed where $D_m = \{0.7\,\text{ns}, 1.4\,\text{ns}, 2.1\,\text{ns}\}$. Deploying a sliding window integration that is three times wider than the frame width, that is, $W = 3T_d = 180$ ns produces the signal $x_m(t)$ with a nonzero support in the range $[0, W + T_h]$. In our case the length of the channel is $T_h = 50$ ns. The solid line depicts the signal at the output of the third receiver branch for the matched transmit and receive delays $\Delta = D_m - D_i = 0$. Signals for the nonmatching delays $D_m \neq D_i$ are depicted by dashed and dash-dotted lines.

2.2. Single chip transmission

According to the spectral regulations, the UWB signal needs to be transmitted at very low-power level. In order to be able to extract the useful information at the receiver side some kind of spreading gain needs to be introduced. The most simple approach is to repeat several, say, N_d frames of total duration $T_c = N_d T_d$. Define such a sequence of frames as *a chip* in which the spacing between pulses (D_i) and the polarity of the information pulses ($c \cdot s$) remains unchanged. In such a case the transmitted signal $t_x(t)$ is given by

$$t_x(t) = \sum_{d=0}^{N_d-1} g(t - dT_d) + c \cdot s \cdot g(t - dT_d - D_i). \quad (8)$$

The signal at the output of the mth receiver branch is computed as the superposition of the contributions of N_d doublets

$$\begin{aligned} x_m(t) &= \sum_{d=0}^{N_d-1} b(t - dT_d)(\alpha_{m,i} \cdot c \cdot s + \beta_m) \\ &= p(t)(\alpha_{m,i} \cdot c \cdot s + \beta_m). \end{aligned} \quad (9)$$

The function $p(t)$ represents a typical response of the sliding window integration process for a case in which a single chip is considered. In general, $p(t)$ has a staircase tent shape and is modeled as

$$p(t) = \sum_{d=0}^{N_d-1} b(t - dT_d), \quad (10)$$

where $b(t)$ is the brick shape function defined in (6). Note that since $b(t)$ depends on the particular channel realization, so does $p(t)$.

In Figure 3 the signal $x_m(t)$ at the integrator outputs is represented. A transmission of a single chip containing three doublets $T_c = 3T_d$ is taken into account. The strongest signal is obtained for matching transmit and receive delays (solid line). Dashed and dash-dotted lines depict the cases in which a delay mismatch occurs ($D_m \neq D_i$). In these cases, the signal is nonzero due to the effect of channel correlation coefficients $\alpha_{m,i}$ and β_m. Note that even though the transmitted chip is T_c wide, the deployment of a sliding window integration of width $W = T_c$ expands the nonzero support of the signal at the receiver side to the $[0, 2T_c]$ region.

2.3. Transceiver design for asynchronous multiple symbol transmission

In this section, we build a data model for the asynchronous transmission of multiple data symbols. As UWB systems cover a large frequency band and in order to avoid catastrophic collisions in multiuser scenarios, the broadcasted signal is spread by means of the polarity and time-hopping codes. As described in Section 2.1 the basic information unit is a *frame* of duration T_d. Further, N_d frames represent a chip of duration $T_c = N_d T_d$, and N_c chips represent a data symbol of duration $T_s = N_c T_c$. The jth chip of the kth data symbol is modulated by $s_k c_j$, where $s_k \in \{\pm 1\}$ represents the data symbol sequence and $c_j \in \{\pm 1\}$, $j = 0, 1, \ldots, N_c - 1$, represents the polarity code. The value of the delay D_i is constant within the jth chip but changes from chip to chip according to the so-called time-hopping code $J_{i,j}$, $i = 1, 2, \ldots, M$, $j = 0, 1, \ldots, N_c - 1$, which is 1 if the delay D_i is used for the jth chip and 0 otherwise. To summarize, the transmitted sequence can be written as

$$\begin{aligned} t_x(t) = \sum_{k=-\infty}^{\infty} \sum_{j=0}^{N_c-1} \sum_{d=0}^{N_d-1} \sum_{i=1}^{M} [&g(t - kT_s - jT_c - dT_d) \\ &+ s_k c_j g(t - kT_s - jT_c - dT_d - D_i)] J_{i,j}. \end{aligned} \quad (11)$$

An example of a transmitted pulse sequence for a single symbol is presented in Figure 4.

Hence, we can write the received sequence as

$$\begin{aligned} r(t) = \sum_{k=-\infty}^{\infty} \sum_{j=0}^{N_c-1} \sum_{d=0}^{N_d-1} \sum_{i=1}^{M} [&h(t - kT_s - jT_c - dT_d) \\ &+ s_k c_j h(t - kT_s - jT_c - dT_d - D_i)] J_{i,j}. \end{aligned} \quad (12)$$

Note that we consider no additive noise throughout this work, in order to simplify the presentation. However, all the simulations will be carried out in the presence of noise.

The output of the mth receiver branch in an asynchronous single user scenario is then modeled as

$$\begin{aligned} &x_m(t) \\ &\quad = \sum_{k=-\infty}^{\infty} \sum_{j=0}^{N_c-1} \sum_{i=1}^{M} p(t - kT_s - jT_c - \tau)[\alpha_{m,i} J_{i,j} c_j s_k + \beta_m J_{i,j}], \end{aligned} \quad (13)$$

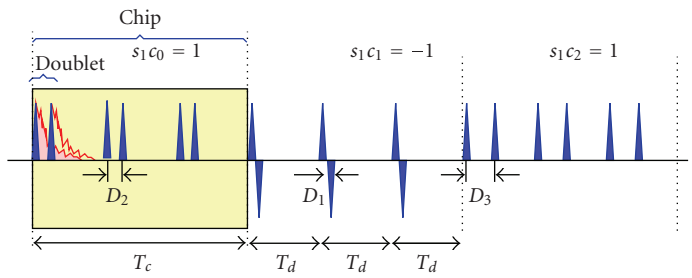


Figure 4: The structure of a transmitted UWB signal. The data symbol is set to $s_1 = +1$. The polarity (CDMA) code vector comprises three chips $\mathbf{c} = [c_0, c_1, c_2]^{\mathrm{T}} = [+1, -1, +1]^{\mathrm{T}}$, the delay code is $\mathbf{J} = [J_{2,0}, J_{1,1}, J_{3,2}]^{\mathrm{T}}$. The latter means that the transmit delays D_2, D_1, and D_3 are used for the 1st, 2nd, and 3rd chips, respectively.

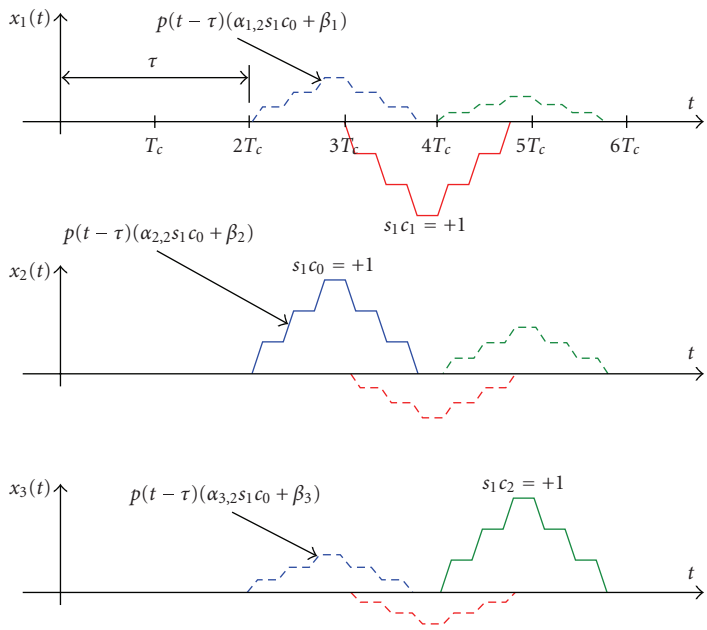


Figure 5: The appearance of the signals at the integrator outputs for the single transmitted data symbol presented in Figure 4.

where τ represents an unknown delay of the received data signal with respect to the beginning of the analysis window, which we try to estimate in this work. Since short polarity and time-hopping codes (c_j and $J_{i,j}$) are considered and symbols are assumed unknown in a first stage, we may restrict τ to the interval $\tau \in [0, T_s)$.

An example of the expected behavior of the signals at the output of the integrators is presented in Figure 5. Solid lines represent the integrator output for matched transmit and receive delays ($D_m = D_i$) while dashed lines depict the residual information for nonmatching delays $D_m \neq D_i$. The overall signal at each receiver branch is obtained as the sum of the matched and nonmatched delay contributions (sum of solid and dashed lines).

The bandwidth of $x_m(t)$ is of the same order of magnitude as the chip rate, which is significantly smaller than the transmission bandwidth. Hence, at this point, it is realistic to introduce sampling and switch to the digital domain. Let us sample $x_m(t)$ at rate P/T_c, where P is the oversampling factor. The sampled signal can then be written as

$$\begin{aligned} x_{m,n} &= x_m(nT_c/P) \\ &= \sum_{k=-\infty}^{\infty} \sum_{j=0}^{N_c-1} \sum_{i=1}^{M} p_{n,j+kN_c}\left[\alpha_{m,i}J_{i,j}c_j s_k + \beta_m J_{i,j}\right], \end{aligned} \tag{14}$$

where $p_{n,j} = p(nT_c/P - jT_c - \tau)$. The crucial observation now is that if we sample once per frame, that is, if we sample at rate $P = N_d$, $p_{n,j}$ may be observed as a sequence of samples of a perfectly known triangular pulse shape (see dash-asterisked line in Figure 3). As a result, $p_{n,j}$ is completely known if τ is an integer multiple of T_c/P. This fact is exploited in the process of estimating an arbitrary offset τ as presented in Section 3.

We generally stack the N_cP samples $x_{m,n}$, $n = kN_cP$, $kN_cP+1, \ldots, (k+1)N_cP - 1$ together in the $N_cP \times 1$ vector

$$\mathbf{x}_{m,k} = \left[x_{m,kN_cP}, \ldots, x_{m,(k+1)N_cP-1}\right]^{\mathrm{T}}, \tag{15}$$

and stack the M vectors $\mathbf{x}_{m,k}$, $m = 1, 2, \ldots, M$, together in the $N_c P \times M$ matrix

$$\mathbf{X}_k = [\mathbf{x}_{1,k}, \ldots, \mathbf{x}_{M,k}]. \tag{16}$$

We now first introduce a matrix model for a single transmitted data symbol, and then generalize this to multiple transmitted data symbols.

2.4. Single transmitted data symbol-matrix model

Suppose only the kth symbol is transmitted. If we then stack vertically $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$, we obtain as in [17] the following matrix model for a single transmitted data symbol:

$$\begin{bmatrix} \mathbf{X}_k \\ \mathbf{X}_{k+1} \end{bmatrix} = \widetilde{\mathbf{P}}\,\mathrm{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} s_k + \widetilde{\mathbf{P}}\mathbf{1}_{N_c}\mathbf{b}^{\mathrm{T}}, \tag{17}$$

where $\mathbf{A}$ and $\mathbf{b}$ are the $M \times M$ matrix and $M \times 1$ vector defined as $[\mathbf{A}]_{m,i} = \alpha_{m,i}$ and $[\mathbf{b}]_m = \beta_m$, respectively. As mentioned before, they depend on the correlation properties of the channel. It can be shown that $\mathbf{A}$ is symmetric, approximately Toeplitz, and diagonally dominant with positive entries on its main diagonal. The $N_c \times 1$ vector $\mathbf{c} = [c_0, \ldots, c_{N_c-1}]^{\mathrm{T}}$ is the known polarity code vector. The matrix $\mathbf{J}$ of size $M \times N_c$ is a known selector matrix which has a single unit element per column (chip), which determines the transmitter delay for that column (chip), or more specifically, $[\mathbf{J}]_{i,j+1} = J_{i,j}$. Finally, $\widetilde{\mathbf{P}}$ is the $2N_c P \times N_c$ block-Toeplitz matrix whose columns are shifts of $p_{n,j}$, or more specifically, $[\widetilde{\mathbf{P}}]_{n+1,j+1} = p_{n,j}$.

Let us now split τ in an integer delay κ and a fractional delay ϵ as $\tau = \kappa T_c/P + \epsilon + T_c/(2P)$, where $\kappa \in \{0, \ldots, N_c P - 1\}$ and $\epsilon \in [-T_c/(2P), T_c/(2P))$ (the additional offset $T_c/(2P)$ is included to force the interval for ϵ symmetric around 0). This allows us to write $\widetilde{\mathbf{P}}$ as

$$\widetilde{\mathbf{P}} = \begin{bmatrix} \mathbf{0}_{\kappa \times N_c} \\ \mathbf{P} \\ \mathbf{0}_{(K-\kappa)\times N_c} \end{bmatrix}, \tag{18}$$

where $K = (N_c - 1)P$ and $\mathbf{P}$ is the $(N_c + 1)P \times N_c$ block-Toeplitz matrix with entries given by $[\mathbf{P}]_{n+1,j+1} = p(nT_c/P - jT_c + T_c/(2P) - \epsilon)$, that is, it only depends on ϵ (see Figure 6). In other words, if we only focus on coarse synchronization, we may assume that $\epsilon = 0$ and thus that $\mathbf{P}$ is known.

As a result, we can rewrite (17) as

$$\begin{aligned} \begin{bmatrix} \mathbf{X}_k \\ \mathbf{X}_{k+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{0}_{\kappa \times M} \\ \mathbf{P}\,\mathrm{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}} \\ \mathbf{0}_{(K-\kappa)\times M} \end{bmatrix} \mathbf{A}^{\mathrm{T}} s_k + \begin{bmatrix} \mathbf{0}_{\kappa \times 1} \\ \mathbf{P}\mathbf{1}_{N_c} \\ \mathbf{0}_{(K-\kappa)\times 1} \end{bmatrix} \mathbf{b}^{\mathrm{T}} \\ &= \begin{bmatrix} \mathbf{0}_{\kappa \times M} & \mathbf{0}_{\kappa \times 1} \\ \mathbf{Z} & \mathbf{q} \\ \mathbf{0}_{(K-\kappa)\times M} & \mathbf{0}_{(K-\kappa)\times 1} \end{bmatrix} \begin{bmatrix} \mathbf{A}^{\mathrm{T}} s_k \\ \mathbf{b}^{\mathrm{T}} \end{bmatrix}, \end{aligned} \tag{19}$$

where $\mathbf{Z} := \mathbf{P}\,\mathrm{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}}$ is a $(N_c + 1)P \times M$ code matrix, which is known if $\epsilon = 0$, and $\mathbf{q} := \mathbf{P}0\mathbf{1}_{N_c} \approx \mathbf{1}_{(N_c+1)P}$ is a known $(N_c + 1)P \times 1$ offset vector. The approximation $\mathbf{q} \approx \mathbf{1}_{(N_c+1)P}$ follows from the structure of the $\mathbf{P}$ matrix. The channel parameters $\mathbf{A}$ and $\mathbf{b}$ as well as the data symbol s_k are unknown.

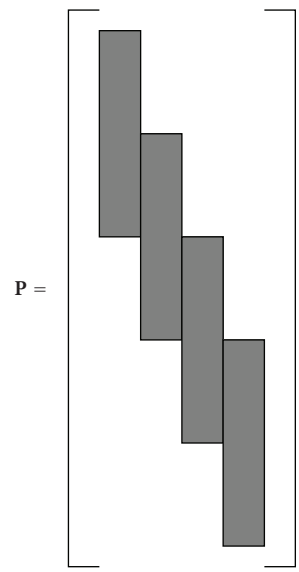


Figure 6: The structure of the $\mathbf{P}$ matrix. Each darkened block (a vector) collects the samples of $p(t)$ and the shifts thereof; $p(t) \neq 0$ for $t \in (0, 2T_c)$.

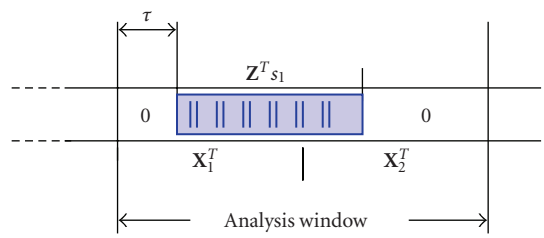


Figure 7: A single symbol spread by the block code, shifted over τ with respect to the beginning of a data packet.

The representation of the block matrix structure for a single symbol is depicted in Figure 7.

Observe that the presented data model resembles a conventional data model for DS-CDMA, up to the channel gain ($\mathbf{A}$) and DC offset ($\mathbf{b}$) term of the channel correlation. This will allow us to use synchronization methods similar in spirit to the DS-CDMA synchronization methods. But before we introduce these synchronization methods, we first generalize the above model to a data model for multiple transmitted data symbols.

2.5. Multiple transmitted data symbol-matrix model

When transmission of multiple data symbols is considered, intersymbol interference (ISI) arises due to the implementation of the sliding window integration. Generally two data symbols affect a single block of received data $\mathbf{X}_k$. Therefore, stacking $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$ vertically, we can modify (19) to the

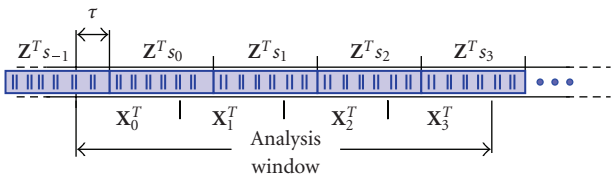


Figure 8: The structure of the analysis window for an asynchronous TR-UWB scheme.

following matrix model:

$$\begin{bmatrix} \mathbf{X}_k \\ \mathbf{X}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}'_\tau & \mathbf{Z}_\tau & \mathbf{Z}''_\tau & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{A}^{\mathrm{T}} s_{k-1} \\ \mathbf{A}^{\mathrm{T}} s_k \\ \mathbf{A}^{\mathrm{T}} s_{k+1} \\ \mathbf{b}^{\mathrm{T}} \end{bmatrix}. \tag{20}$$

The block columns $\mathbf{Z}'_\tau$, $\mathbf{Z}_\tau$, and $\mathbf{Z}''_\tau$, all of size $2N_cP \times M$, comprise the effects of the polarity and time-hopping codes as well as the effect of the pulse shape $p(t)$. We begin with defining the second block column $\mathbf{Z}_\tau$, which is similar to the first block column of (19): $\mathbf{Z}_\tau = [\mathbf{0}^{\mathrm{T}}_{\kappa\times M}, \mathbf{Z}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}_{(K-\kappa)\times M}]^{\mathrm{T}}$. This block column comprises the complete version of a user specific code matrix $\mathbf{Z} = \mathbf{P}\,\mathrm{diag}(\mathbf{c})\mathbf{J}^{\mathrm{T}}$, which is known if $\epsilon = 0$, shifted by an integer delay κ. The other two block columns $\mathbf{Z}'_\tau$ and $\mathbf{Z}''_\tau$ can be defined as $\mathbf{Z}'_\tau = [\mathbf{Z}'^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}_{(N_cP+K-\kappa)\times M}]^{\mathrm{T}}$ and $\mathbf{Z}''_\tau = [\mathbf{0}^{\mathrm{T}}_{(N_cP+\kappa)\times M}, \mathbf{Z}''^{\mathrm{T}}]^{\mathrm{T}}$. They contain only part of the user block code $\mathbf{Z}$. $\mathbf{Z}'$, with size $(N_cP - K + \kappa) \times M$, and $\mathbf{Z}''$, with size $(N_cP - \kappa) \times M$, depict the effect of a "previous" and a "subsequent" data symbol, respectively. It is thereby crucial to observe that $\mathbf{Z} = [\mathbf{Z}''^{\mathrm{T}} \;\; \mathbf{Z}'^{\mathrm{T}}]^{\mathrm{T}}$. Writing (20) in a more compact form, we obtain

$$\begin{bmatrix} \mathbf{X}_k \\ \mathbf{X}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{S}_k \\ \mathbf{b}^{\mathrm{T}} \end{bmatrix}, \tag{21}$$

where $\mathbf{G} = [\mathbf{Z}'_\tau, \mathbf{Z}_\tau, \mathbf{Z}''_\tau]$ and $\mathbf{S}_k = [\mathbf{A}s_{k-1}, \mathbf{A}s_k, \mathbf{A}s_{k+1}]^{\mathrm{T}}$.

Let us now define a received data matrix $\mathbf{X}$ as

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_0 & \mathbf{X}_1 & \cdots & \mathbf{X}_{n-1} \\ \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}, \tag{22}$$

where n is the length of the analysis window over which data is collected. Using (21), we can write this matrix as

$$\mathbf{X} = \begin{bmatrix} \mathbf{G} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{S} \\ \mathbf{1}^{\mathrm{T}}_n \otimes \mathbf{b}^{\mathrm{T}} \end{bmatrix}, \tag{23}$$

where $\mathbf{S} = [\mathbf{S}_0, \ldots, \mathbf{S}_{n-1}]$ (see also Figure 9). The structure of the received data blocks for multiple transmitted symbols is depicted in Figure 8.

In the case where the analysis window is not within the transmitted packet, we can use the same model but allow some of the symbols s_k to be zero (note that $\mathbf{1}^{\mathrm{T}}_n \otimes \mathbf{b}^{\mathrm{T}}$ will also change in that case).

3. BLIND SYNCHRONIZATION ALGORITHM

3.1. Optimization problem

We now describe the synchronization algorithm. In Figure 8 the relation between the received data at the integrator outputs $\mathbf{X}_k$ and the transmitted symbols is presented. We describe a block algorithm that provides an estimate of the delay τ, and also allows us to estimate the data symbols s_k.

The algorithm is an extension of the algorithm of Torlak and Xu [13], who considered blind channel estimation for DS-CDMA using subspace techniques. The idea is to use the property that the matrix $\mathbf{G}$ is orthogonal to the left nullspace $(\mathbf{U}_0)$ of the matrix $\mathbf{X}$, that is, $\mathbf{U}^{\mathrm{H}}_0\mathbf{G} = \mathbf{0}$. We can use this relationship in order to find an estimate of τ. More specifically, we solve

$$\begin{aligned} &\arg\min_\tau \|\mathbf{U}^{\mathrm{H}}_0\mathbf{G}\|^2 \\ &\quad = \arg\min_\tau \sum_i \left\| \begin{bmatrix} \mathbf{u}^{(i)}_1 \\ \mathbf{u}^{(i)}_2 \end{bmatrix}^{\mathrm{H}} \begin{bmatrix} \mathbf{Z}_2 & \mathbf{Z}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \mathbf{Z}_1 \end{bmatrix} \right\|^2, \end{aligned} \tag{24}$$

where $\mathbf{u}^{(i)}_1$ and $\mathbf{u}^{(i)}_2$ are both of size $N_cP \times 1$ and depict the first and the second halves of the ith column of $\mathbf{U}_0$, respectively. $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are of size $N_cP \times M$ and represent the upper and lower halves of the middle block column of $\mathbf{G}$, that is, $\mathbf{Z}_\tau = [\mathbf{Z}^{\mathrm{T}}_1 \;\; \mathbf{Z}^{\mathrm{T}}_2]^{\mathrm{T}}$.

We now aim to transform (24) without changing the criterion, in order to bring out the block column $\mathbf{Z}_\tau$, containing the user specific code matrix $\mathbf{Z}$, which is known if $\epsilon = 0$, shifted by an integer delay κ. Restacking (24) as in [13] yields

$$\arg\min_\tau \sum_i \left\| \mathbf{Z}^{\mathrm{H}}_\tau \begin{bmatrix} \mathbf{0} & \mathbf{u}^{(i)}_1 & \mathbf{u}^{(i)}_2 \\ \mathbf{u}^{(i)}_1 & \mathbf{u}^{(i)}_2 & \mathbf{0} \end{bmatrix} \right\|^2 = \arg\min_\tau \sum_i \|\mathbf{Z}^{\mathrm{H}}_\tau \mathcal{U}^{(i)}\|^2. \tag{25}$$

Here, i sweeps all the vectors from the left null space of $\mathbf{X}$. By stacking horizontally $\mathcal{U}^{(i)}$ for all possible i's we get the matrix $\mathcal{U}_0$. Now (25) can be written as

$$\arg\min_\tau \|\mathbf{Z}^{\mathrm{H}}_\tau \mathcal{U}_0\|^2. \tag{26}$$

At this point, let us make a distinction between integer delay estimation and noninteger delay estimation.

3.2. Integer delay estimation

We first assume that $\epsilon = 0$, and focus on the estimation of the integer delay κ. As already mentioned before, if $\epsilon = 0$, the matrix $\mathbf{P}$ and thus the matrix $\mathbf{Z}$ are completely known. As a result, $\mathbf{Z}_\tau$, which can then be written as $\mathbf{Z}_{\kappa T_c/P}$, only depends on κ and we can rewrite (26) as

$$\arg\min_\kappa \|\mathbf{Z}^{\mathrm{H}}_{\kappa T_c/P} \mathcal{U}_0\|^2. \tag{27}$$

This can be solved by performing an exhaustive search over $\kappa \in \{0, \ldots, N_cP - 1\}$, since we know $\mathbf{Z}_{\kappa T_c/P}$ up to the integer delay κ.

Since the above time-domain approach is rather computationally intensive, we switch to a much simpler frequency-domain approach, recognizing that an integer shift in the time domain corresponds to a phase shift in the frequency domain. More specifically, we can write $\mathbf{Z}_{\kappa T_c/P}$ as

$$\mathbf{Z}_{\kappa T_c/P} = \mathbf{F}^{\mathrm{H}} \mathbf{D}_{\kappa T_c/P} \mathbf{F} \mathbf{Z}_0, \tag{28}$$

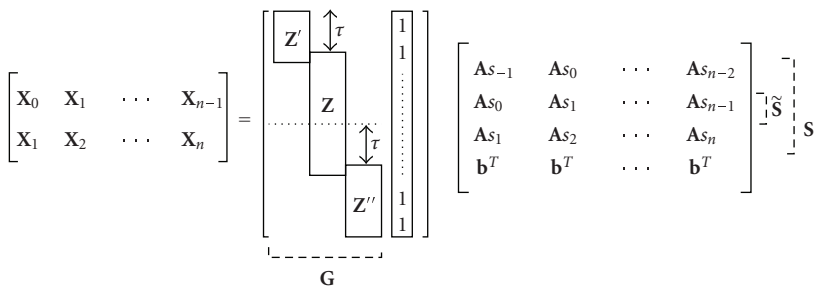


FIGURE 9: Block data model $\mathbf{X} = [\mathbf{G}\ \ \mathbf{1}][\mathbf{S}^{\mathrm{T}}\ \ (\mathbf{1}_n \otimes \mathbf{b})]^{\mathrm{T}}$ for the asynchronous single user case using a TR-UWB scheme.

where $\mathbf{F}$ stands for the $2N_cP \times 2N_cP$ normalized discrete Fourier transform matrix, $\mathbf{D}_\tau$ represents the $2N_cP \times 2N_cP$ diagonal matrix given by

$$\mathbf{D}_\tau = \operatorname{diag}\left([1, e^{-j2\pi\tau/(2N_cT_c)}, \ldots, e^{-j2\pi\tau(2N_cP-1)/(2N_cT_c)}]\right), \tag{29}$$

and $\mathbf{Z}_0$ is a completely known $2N_cP \times M$ matrix. If we now denote $\mathbf{z}_0^{(l)\mathrm{H}}$ as the lth row of $\mathbf{Z}_0^{\mathrm{H}}$, and define $\widetilde{\mathbf{z}}_0^{(l)} := \mathbf{F}\mathbf{z}_0^{(l)}$, $\widetilde{\mathcal{U}}_0 := \mathbf{F}\mathcal{U}_0$, and $\boldsymbol{\phi}_\tau = \operatorname{diag}(\mathbf{D}_\tau)$, we can rewrite (27) as

$$\begin{aligned}
&\arg\min_{\kappa} \left\|\mathbf{Z}^{\mathrm{H}}_{\kappa T_c/P}\mathcal{U}_0\right\|^2 \\
&\quad = \arg\min_{\kappa} \left\|\mathbf{Z}_0^{\mathrm{H}}\mathbf{F}^{\mathrm{H}}\mathbf{D}^*_{\kappa T_c/P}\mathbf{F}\mathcal{U}_0\right\|^2 \\
&\quad = \arg\min_{\kappa} \left\|\mathbf{z}_0^{(1)H}\mathbf{F}^{\mathrm{H}}\mathbf{D}^*_{\kappa T_c/P}\mathbf{F}\mathcal{U}_0 | \cdots | \mathbf{z}_0^{(M)H}\mathbf{F}^{\mathrm{H}}\mathbf{D}^*_{\kappa T_c/P}\mathbf{F}\mathcal{U}_0\right\|^2 \\
&\quad = \arg\min_{\kappa} \left\|\widetilde{\mathbf{z}}_0^{(1)H}\mathbf{D}^*_{\kappa T_c/P}\widetilde{\mathcal{U}}_0 | \cdots | \widetilde{\mathbf{z}}_0^{(M)H}\mathbf{D}^*_{\kappa T_c/P}\widetilde{\mathcal{U}}_0\right\|^2 \\
&\quad = \arg\min_{\kappa} \sum_{l=1}^{M} \left\|\boldsymbol{\phi}^{\mathrm{H}}_{\kappa T_c/P}\operatorname{diag}(\widetilde{\mathbf{z}}_0^{(l)H})\widetilde{\mathcal{U}}_0\right\|^2 \\
&\quad = \arg\min_{\kappa} \left\|\boldsymbol{\phi}^{\mathrm{H}}_{\kappa T_c/P}[\operatorname{diag}(\widetilde{\mathbf{z}}_0^{(1)H})\widetilde{\mathcal{U}}_0 | \cdots | \operatorname{diag}(\widetilde{\mathbf{z}}_0^{(M)H})\widetilde{\mathcal{U}}_0]\right\|^2 \\
&\quad = \arg\min_{\kappa} \left\|\boldsymbol{\phi}^{\mathrm{H}}_{\kappa T_c/P}\mathcal{K}\right\|^2.
\end{aligned} \tag{30}$$

Due to the structure of $\boldsymbol{\phi}_{\kappa T_c/P}$ that corresponds to the $(\kappa+1)$th column of the FFT matrix $\mathbf{F}$, searching for the $\boldsymbol{\phi}_{\kappa T_c/P}$ that minimizes the last expression is equivalent to performing an inverse FFT (IFFT) on the matrix $\mathcal{K}$ and searching for the row of the resulting matrix that has the lowest norm. The index of the row with the lowest norm determines the integer delay κ. Note that through the use of the (I)FFT this frequency-domain approach is much simpler than the earlier developed time-domain approach. However, since we have assumed $\epsilon = 0$, the resolution of this algorithm is limited by the sampling period T_c/P. This problem will be treated in the next section.

In order to compare the computational complexity of the integer delay estimation carried out in the time and frequency domain, we compute the number of multiplications needed in both cases. The time-domain search requires $\mathcal{O}(2M^2(N_cP)^4)$ multiplications, in contrast to $\mathcal{O}(M^2(2N_cP)^2\log_2(2N_cP))$ multiplications in case the proposed frequency-domain search is employed.

3.3. Noninteger delay estimation

Let us now consider the more general case, where $\epsilon \neq 0$. We can then actually proceed as in the previous section, by observing that if the sampling rate is close to the Nyquist rate, we can also express a noninteger shift in the time domain by a phase shift in the frequency domain. In other words, we can extend (28) for the noninteger delay case to

$$\mathbf{Z}_\tau = \mathbf{F}^{\mathrm{H}}\mathbf{D}_\tau\mathbf{F}\mathbf{Z}_0. \tag{31}$$

Following similar steps as in the previous section, we can then transform (26) to

$$\arg\min_{\tau} \left\|\boldsymbol{\phi}_\tau^{\mathrm{H}}\mathcal{K}\right\|^2. \tag{32}$$

As before, we can first look for an integer delay κ by computing the IFFT of $\mathcal{K}$ and searching for the row of the resulting matrix that has the lowest norm. The fractional delay ϵ is then obtained by performing an additional fine grid MUSIC-kind search around $\hat{\kappa}T_c/P$:

$$\arg\min_{\epsilon} \left\|\boldsymbol{\phi}^{\mathrm{H}}_{\hat{\kappa}T_c/P+\epsilon}\mathcal{K}\right\|^2. \tag{33}$$

The overall delay estimate is finally given by $\hat{\tau} = \hat{\kappa}T_c/P + \hat{\epsilon}$.

3.4. Symbol estimation

After estimating the delay τ, we can reconstruct the complete $\mathbf{G}$ matrix. Estimation of the transmitted data symbols is now possible by performing a deconvolution of the matrix $\mathbf{X}$ using the known user code, that is, we compute

$$\begin{bmatrix} \hat{\mathbf{S}} \\ \mathbf{1}_n^{\mathrm{T}} \otimes \hat{\mathbf{b}}^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{1} \end{bmatrix}^{\dagger}\mathbf{X}, \tag{34}$$

where $\dagger$ denotes the pseudoinverse. We subsequently limit our attention to the middle block row of $\hat{\mathbf{S}}$, name it $\widetilde{\mathbf{S}}$ as the part that carries most of the energy (see also Figure 9). The data symbols at this point can be estimated from $\widetilde{\mathbf{S}}$ in two different ways [17]: (i) by computation of the trace of the $M \times M$ data blocks $\mathbf{A}s_k$, or (ii) by vectorizing the $M \times M$ data blocks $\mathbf{A}s_k$ and stacking the results column-wise into a matrix, such that we get a rank one matrix whose row span corresponds to a scaled version of the data symbols. In both cases, the estimates can be further refined by iterations [12].

4. EXTENSION TO THE MULTIUSER CASE

In this section, we extend the previous ideas developed for a single user to multiple users. Let us start by extending the data model of Section 2.5 to multiple users. This is not trivial, since next to the autocorrelation terms of the different users, there are also crosscorrelation terms, due to the use of the autocorrelation receiver. However, since different users employ distinct time-hopping and polarity codes, propagate through different channels, and arrive at the receiver at random time instants, we can treat these cross terms as additive white noise, and add them to the other noise terms that might be present. As before, we do not take the additive noise terms into account in our derivations, but we do include them in our simulations.

As a result, indicating the user index by means of a superscript q ($q = 1, 2, \ldots, Q$), we can write the received data block $\mathbf{X}$ as

$$\begin{aligned}\mathbf{X} &= \sum_{q=1}^{Q} \left[\mathbf{G}^{(q)} \quad \mathbf{1}\right] \begin{bmatrix} \mathbf{S}^{(q)} \\ \mathbf{1}_n{}^{\mathrm{T}} \otimes \mathbf{b}^{(q)\mathrm{T}} \end{bmatrix} \\ &= \left[\mathbf{G}^{(1)} | \cdots | \mathbf{G}^{(Q)} \quad | \, \mathbf{1}\right] \begin{bmatrix} \mathbf{S}^{(1)} \\ \vdots \\ \mathbf{S}^{(Q)} \\ \mathbf{1}_n{}^{\mathrm{T}} \otimes \sum_{q=1}^{Q} \mathbf{b}^{(q)\mathrm{T}} \end{bmatrix},\end{aligned} \tag{35}$$

where $\mathbf{S}^{(q)} = [\mathbf{S}_0^{(q)}, \ldots, \mathbf{S}_{n-1}^{(q)}]$. Note that in the case some users are not active for the duration of the whole analysis window, several $\mathbf{S}_k^{(q)}$ matrices will be zero and some small changes in the structure of $\mathbf{1}_n^{\mathrm{T}} \otimes \sum_{q=1}^{Q} \mathbf{b}^{(q)\mathrm{T}}$ will occur. Consequently, a few additional vectors with low energy may emerge in the left signal space.

4.1. Identifiability for multiuser case

In the multiuser (MU) case as presented in (35), the matrix $\mathbf{G}_{MU} = [\,\mathbf{G}^{(1)}|\cdots|\mathbf{G}^{(Q)}\,|\mathbf{1}\,]$ is of size $2N_cP \times (3MQ + 1)$, whereas the matrix comprising all data blocks and offset effects, $\mathbf{S}_{MU} = [\mathbf{S}^{(1)T}, \ldots, \mathbf{S}^{(Q)T}, \mathbf{1}_n \otimes \sum_{q=1}^{Q} \mathbf{b}^{(q)}]^{\mathrm{T}}$, is of size $(3MQ + 1) \times Mn$. In order to determine the column space of $\mathbf{G}_{MU}$ from $\mathbf{X}$ (and hence its left nullspace), $\mathbf{G}_{MU}$ should be tall and of full column rank, that is, $2N_cP > 3MQ + 1$, and $\mathbf{S}_{MU}$ should be fat and of full row rank, that is, $3MQ + 1 < Mn$. Note that a full column rank $\mathbf{G}_{MU}$ is also required to subsequently detect the data symbols. From the first condition, a limit on the code size is obtained: $N_c > 3MQ/2P$, which for typical values of $P = 2$ and $M = 4$ yields $N_c > 3Q$. The condition on the size of $\mathbf{S}_{MU}$ gives the relation between the number of users Q and the lowest number of symbols transmitted n, that is, $Q < (Mn - 1)/3M$.

5. APPLICATION IN UWB NETWORKING

The ability to achieve high resolution packet offset estimation in a multiuser environment in a fast and computationally simple way is of crucial importance for the subsequent data symbol estimation step. Imagine the scenario of a UWB ad hoc network where users need to exchange their codes at the moment they join the network. The simplest way to solve this problem is to implement a *common code* known to all the users in the initialization phase. In existing wireless network protocols a data packet is considered to be lost if several users simultaneously use the same code which is known as the packet collision problem. Nevertheless, the structure and the design of the considered TR-UWB scheme will allow us to avoid the collision problem. In TR-UWB systems, different users propagate through different channels creating distinct correlation matrices $\mathbf{A}^{(i)}$. This can be viewed as an additional coding introduced by the channel itself and can be adopted to solve the collision problem, as illustrated next.

Consider a two-user system where both users adopt the same spreading code. The data model (35) then becomes

$$\mathbf{X} = \left[\mathbf{G}_{\tau_1} \quad \mathbf{G}_{\tau_2} \quad \mathbf{1}\right] \left[\mathbf{S}^{(1)\mathrm{T}} \quad \mathbf{S}^{(2)\mathrm{T}} \quad \left(\mathbf{1}_n \otimes \sum_{q=1}^{2} \mathbf{b}^{(q)}\right)\right]^{\mathrm{T}}. \tag{36}$$

The synchronization of both users to the common code and the subsequent data detection is in general only possible if $\tau_i \neq \tau_j$ for $i \neq j$ and by implementation of a common code that has a low autocorrelation property.

But even if the two users completely overlap in time it is still possible to separate both overlapping users and detect their data sequences. In that case the linear dependency between $\mathbf{G}_{\tau_1}$ and $\mathbf{G}_{\tau_2}$ reduces the rank of the code matrix, that is, $[\,\mathbf{G}_{\tau_1} \; \mathbf{G}_{\tau_2}\,] \rightarrow [\,\mathbf{G}_{\tau_1=\tau_2}\,]$. As a consequence, data blocks $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ merge into a single block $\mathbf{S}' = \mathbf{S}^{(1)} + \mathbf{S}^{(2)}$ changing (36) to

$$\mathbf{X} = \left[\mathbf{G}_{\tau_1=\tau_2} \quad \mathbf{1}\right] \left[\mathbf{S}'^{\mathrm{T}} \quad \left(\mathbf{1}_n \otimes \sum_{q=1}^{2} \mathbf{b}^{(q)}\right)\right]^{\mathrm{T}}. \tag{37}$$

Estimating the packet offset delay $\tau_1 = \tau_2$, we can reconstruct $\widehat{\mathbf{G}}_{\tau_1=\tau_2}$ and subsequently as in (34) obtain an estimate of the data matrix $\widehat{\mathbf{S}}' = \widehat{\mathbf{S}}^{(1)} + \widehat{\mathbf{S}}^{(2)}$. Considering only the mid-block row of $\widehat{\mathbf{S}}' = \widehat{\mathbf{S}}^{(1)} + \widehat{\mathbf{S}}^{(2)}$ (see Figure 9) we get $\widehat{\widetilde{\mathbf{S}}}' = \widehat{\widetilde{\mathbf{S}}}^{(1)} + \widehat{\widetilde{\mathbf{S}}}^{(2)}$, which can be modeled as

$$\widehat{\widetilde{\mathbf{S}}}' = [\mathbf{A}^{(1)}s_1^{(1)} + \mathbf{A}^{(2)}s_1^{(2)}, \ldots, \mathbf{A}^{(1)}s_n^{(1)} + \mathbf{A}^{(2)}s_n^{(2)}]. \tag{38}$$

Performing the vectorization of each $M \times M$ block of $\widehat{\widetilde{\mathbf{S}}}'$ yields

$$\begin{aligned}&[\,\mathrm{vec}\,(\mathbf{A}^{(1)}s_1^{(1)} + \mathbf{A}^{(2)}s_1^{(2)}), \ldots, \mathrm{vec}\,(\mathbf{A}^{(1)}s_n^{(1)} + \mathbf{A}^{(2)}s_n^{(2)})\,] \\ &\quad = \left[\mathbf{a}^{(1)} \quad \mathbf{a}^{(2)}\right] \begin{bmatrix} s_1^{(1)} & s_2^{(1)} & \cdots & s_n^{(1)} \\ s_1^{(2)} & s_2^{(2)} & \cdots & s_n^{(2)} \end{bmatrix},\end{aligned} \tag{39}$$

where $\mathbf{a}^{(i)} = \mathrm{vec}(\mathbf{A}^{(i)})$. A singular value decomposition of $\widehat{\widetilde{\mathbf{S}}}'$ produces a rank-two decomposition and is an indication of the existence of two overlapping users. Now, the column vectors $(\mathbf{a}^{(1)}, \mathbf{a}^{(2)})$ and the data symbols $(\{s_k^{(1)}\}, \{s_k^{(2)}\})$ can be estimated from the column and row span of $\widehat{\widetilde{\mathbf{S}}}'$. This approach fails only in the case when $\mathbf{A}^{(1)} = \gamma\mathbf{A}^{(2)}$ where γ is a scalar, but this has an extremely low probability of occurrence.

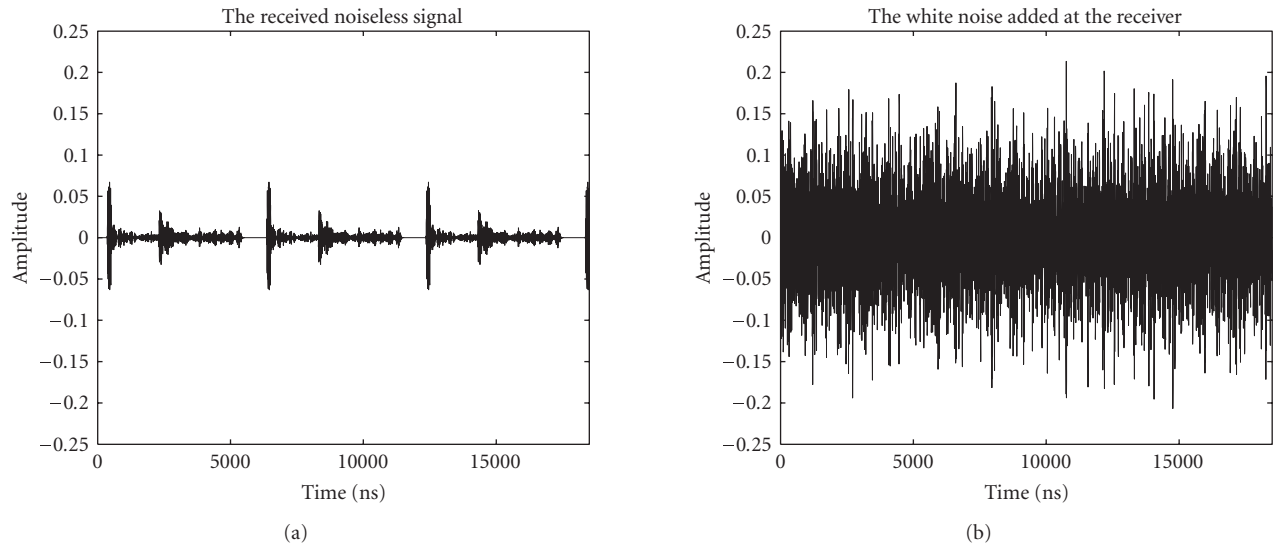


Figure 10: Received single user signal (a) and noise (b) ($E_b/N_0 = 34$ dB).

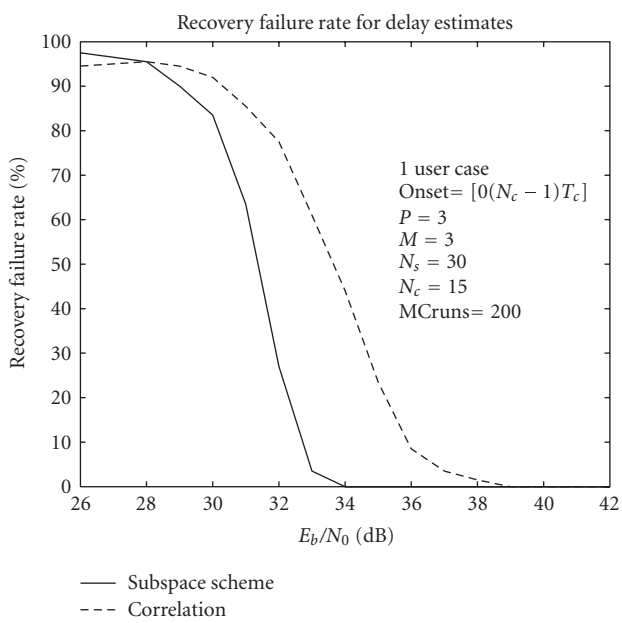


Figure 11: The percentage of incorrectly estimated packet offsets using the proposed subspace-based (solid line) and correlation-based (dashed line) schemes.

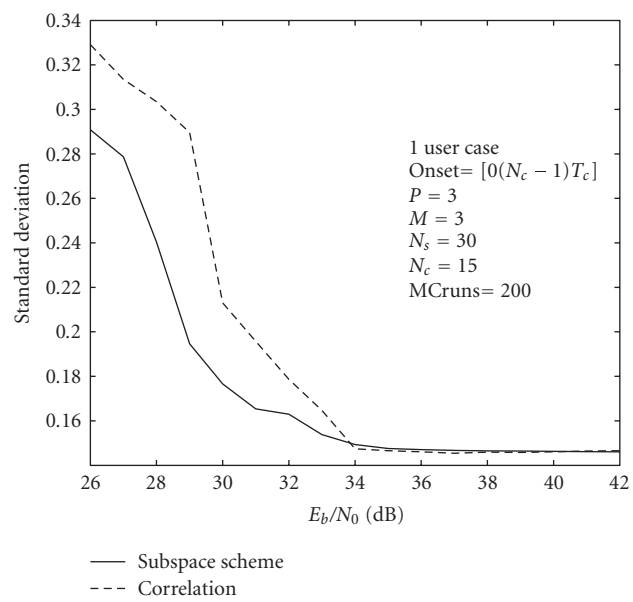


Figure 12: Standard deviation of the correctly estimated packet offset delays.

6. SIMULATIONS

6.1. Single user case

The performance of the proposed algorithm is first tested for a single user in noise. Signals are generated in accordance to the description provided in Section 2. Two hundred and fifty Monte Carlo runs are performed for fixed polarity and time-hopping codes. Data symbols and noise are varied in each run, as well as the packet offsets, which are randomly chosen from the interval $[0, N_c T_c)$. We consider the transmission of $N_s = 30$ data symbols, a polarity code length of $N_c = 15$ chips, and $M = 3$ possible delays $D_1 = 0.7$ ns, $D_2 = 1.4$ ns, and $D_3 = 2.1$ ns. Transmitted data pulses are convolved with the channel impulse responses measured for different scenarios in a typical university building. The following scenarios are taken into account: (1) office, (2) corridor, (3) corridor-to-office, (4) library, and (5) office-to-office. Both line of

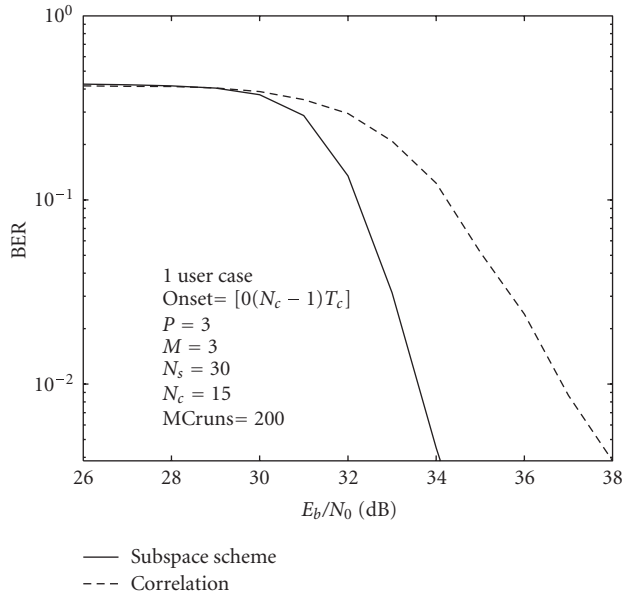


Figure 13: BER of the estimated data symbols computed for all estimates of τ.

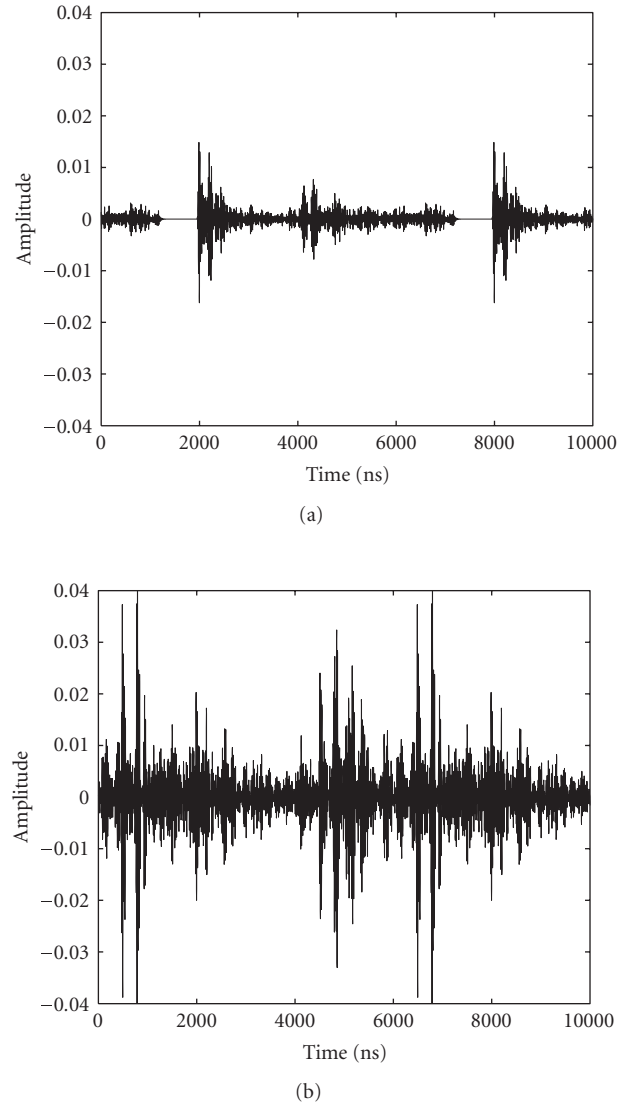


Figure 14: Received signal of a single user (a) and three users (b) (SIR = −10 dB).

site and nonline of site channel impulse responses are covered in this fashion (see [16]). A sampling rate of 10 ps is used in the channel measurements. We limit the measured channel impulse responses to the interval [0, 50] ns, as the contributions of the channel components that fall outside this interval are insignificant. The duration of the frame is chosen to be $T_d = 60$ ns. The energy of a single transmitted data symbol (bit) is defined as $E_b = 2N_dN_c \int [h(t)]^2 dt$, where $h(t)$ represents the total channel impulse response, including the monocycle and the transmit and receive antenna effects. N_0 is the power spectral density of the white Gaussian noise which is added after the receive antenna. The E_b/N_0 is changed in steps of 1 dB.

After the white Gaussian noise is added, a bandpass filtering is performed to limit the bandwidth of the observed signal to the interval 4–10 GHz. This filtering step reduces the impact of the noise and the low-frequency interference. Note though that the E_b/N_0 is computed before the filtering.

At the output of the autocorrelation receiver with three receiver branches, oversampling is performed at a rate equal to the number of frames per chip, that is, $P = N_d = 3$. Note that the oversampling factor $P = 3$ leads to a sampling rate that is still lower than the Nyquist rate for the expected pulse shapes at the integrator outputs, but it is sufficient as the estimation errors for τ are negligible compared to sampling at the Nyquist rate [18].

An example of a received single user signal and noise (after bandpass filtering) is presented in Figure 10. In this figure $E_b/N_0 = 34$ dB. At this E_b/N_0, the useful signal will clearly drown in the noise, yet the proposed method can synchronize, as will be illustrated next.

Figure 11 shows the percentage of cases where the packet offset is estimated incorrectly. An estimate is considered to be incorrect if it does not fall into the interval $\tau - T_c/2 \leq \hat{\tau} < \tau + T_c/2$. The solid line depicts the performance of the delay estimation based on the subspace-based frequency-domain search (as presented in Section 3.3), while the dashed line shows the performance of the correlation-based scheme:

$$\arg\max_{\tau} \left|\left|\mathbf{X}^{\mathrm{H}}\mathbf{G}\right|\right|^2, \tag{40}$$

which can be solved in a similar fashion as the subspace-based scheme, but which does not require a subspace decomposition. Figure 12 shows the standard deviation of the "good" estimates of τ expressed as a fraction of the chip duration T_c for the subspace-based (solid line) and correlation-based (dashed) schemes. In Figure 13 the BER of the symbol estimates is shown for all estimates of τ. We deployed a decorrelating receiver and the vectorization approach described in Section 3.4.

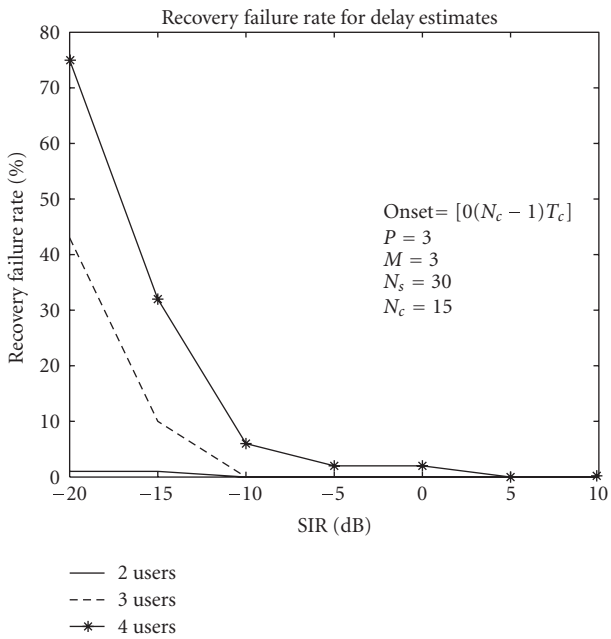


Figure 15: The percentage of incorrectly estimated packet offsets.

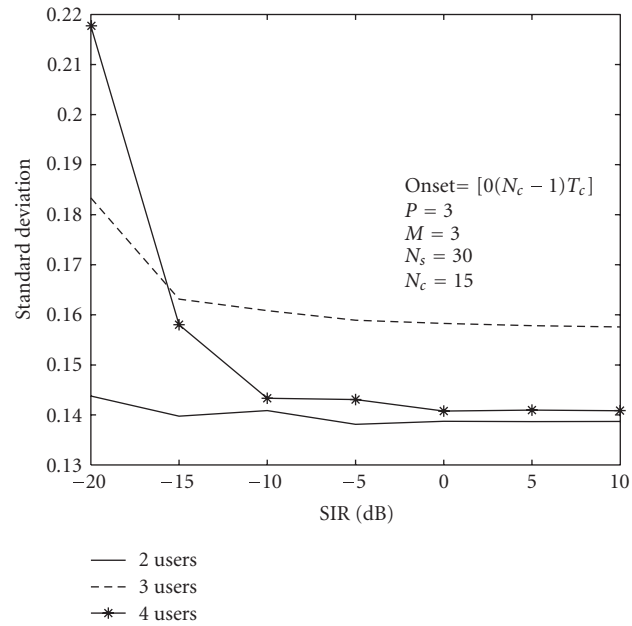


Figure 16: Standard deviation of the correctly estimated packet offset delays.

6.2. Multiple user case

In this section, we test the resistivity of the proposed synchronization scheme to multiuser interference. To clearly see the effect of the multiuser interference, we consider a noiseless scenario. We define the signal to interference ratio (SIR) as SIR $= 10\log(P_1/P_I)$, where P_i represents the energy of a single data symbol (bit) of user i at the receiver before the bandpass filtering step is carried out. Note that it corresponds to the E_b for user i. As each data symbol (bit) is spread over N_c chips and further over N_d frames containing two pulses (a doublet) the signal energy can be expressed as $P_i = 2N_dN_c\int[h^{(i)}(t)]^2dt$, where $h^{(i)}(t)$ is the channel corresponding to user i. $P_I = \sum_{i=2}^{N} P_i$ collects the energy P_i of all interfering sources $i = 2,\ldots,N$. In Figure 14, the received signal of a single user (a) and three users (b) can be observed (after bandpass filtering). For the three-user case, we take SIR $= -10$ dB, that is, the two interfering users together are 10 dB stronger than the user of interest. Note that the x-axis represents the number of samples, where sampling is performed at a rate of 10 ps.

In Figure 15, we present the recovery failure rate versus the signal to interference ratio. Any packet offset estimate $\hat{\tau}$ that does not fit into the range $\tau - T_c/2 < \hat{\tau} < \tau + T_c/2$ is considered to be a failure. The chosen interval is considered to provide sufficient recovery of the energy after the deployment of a decorrelating receiver. We start by choosing a time-hopping and polarity code, the latter being a gold code sequence of length N_c. Both codes remain unchanged in all Monte Carlo runs. In each run, a new set of channels as well as packet offsets is assigned to each of the users and is kept the same for all the values of the SIR changed in steps of 5 dB. $N_s = 30$ stands for the number of data symbols within a transmitted packet. The oversampling rate $P = 3$ equals the number of frames (doublets) per chip. Delays used in the time-hopping scheme are chosen to be $D_1 = 1$ ns, $D_2 = 2$ ns, and $D_3 = 3$ ns. In Figure 15, the solid, dashed, and dash-asterisked lines correspond to the two-, three-, and four-user cases, respectively. The performance of the algorithm drops by increasing the total number of users. This can be explained by an augmented influence of the cross-correlation terms as the number of users increases. However, in the four-user case, the algorithm exhibits a low failure rate even for SIR $= 0$ dB, that is, in the case the energy of the signal of interest equals the energy of all interfering sources. This issue could be improved by selecting the user codes to have lower cross-correlation properties for any code offset.

Figure 16 describes the standard deviation of the "good" estimates of τ expressed as a fraction of the chip duration T_c. Due to the low number of Monte Carlo iteration, and to the unresolvable ambiguity related to the initial sampling point, the three-user scenario has a slightly degraded performance compared to the other scenarios.

7. CONCLUDING REMARK

In this paper, we have presented an algorithm that provides fast, low-complexity, blind packet synchronization in multiuser TR-UWB systems. Its foremost application could be the fast initial code exchange in multiuser asynchronous UWB ad hoc networks.

ACKNOWLEDGMENTS

The authors would like to thank Zoubir Irahhauten from Delft University of Technology for providing us the measured

channel impulse responses. This research was supported in part by the Dutch Min. Econ. Affairs/Min. Education Freeband-impulse project *Air-Link* and by NWO-STW under the *VICI* program (DTC.5893). Parts of this paper were presented in [17, 19].

REFERENCES

[1] L. Yang and G. B. Giannakis, "Ultra-wideband communications," *IEEE Signal Processing Magazine*, vol. 21, no. 6, pp. 26–54, 2004.

[2] S. Bagga, W. A. Serdijn, and J. R. Long, "A PPM Gaussian monocycle transmitter for ultra-wideband communications," in *International Workshop on Ultra Wideband Systems; Joint with Conference on Ultra Wideband Systems and Technologies*, pp. 130–134, Kyoto, Japan, May 2004.

[3] D. Cassioli, M. Z. Win, and A. F. Molisch, "The ultra-wide bandwidth indoor channel: from statistical model to simulations," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 6, pp. 1247–1257, 2002.

[4] M. Z. Win and R. A. Scholtz, "On the robustness of ultra-wide bandwidth signals in dense multipath environments," *IEEE Communications Letters*, vol. 2, no. 2, pp. 51–53, 1998.

[5] M. Z. Win and R. A. Scholtz, "Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications," *IEEE Transactions on Communications*, vol. 48, no. 4, pp. 679–691, 2000.

[6] M. Z. Win and R. A. Scholtz, "Impulse radio: how it works," *IEEE Communications Letters*, vol. 2, no. 2, pp. 36–38, 1998.

[7] R. Gagliardi, "A geometrical study of transmitted reference communication systems," *IEEE Transactions on Communication Technology*, vol. 12, no. 4, pp. 118–123, 1964.

[8] R. Hoctor and H. Tomlinson, "Delay-hopped transmitted-reference RF communications," in *IEEE Conference on Ultra Wideband Systems and Technologies (UWBST '02)*, pp. 265–270, Baltimore, Md, USA, May 2002.

[9] N. van Stralen, A. Dentinger, K. Welles II, R. Gauss Jr., R. Hoctor, and H. Tomlinson, "Delay hopped transmitted reference experimental results," in *IEEE Conference on Ultra Wideband Systems and Technologies (UWBST '02)*, pp. 93–98, Baltimore, Md, USA, May 2002.

[10] G. Leus and A.-J. van der Veen, "Noise suppression in UWB transmitted reference systems," in *IEEE 5th Workshop on Signal Processing Advances in Wireless Communications (SPAWC '04)*, pp. 155–159, Lisbon, Portugal, July 2004.

[11] A. Trindade, Q. H. Dang, and A.-J. van der Veen, "Signal processing model for a transmit-reference UWB wireless communication system," in *IEEE Conference on Ultra Wideband Systems and Technologies (UWBST '03)*, pp. 270–274, Reston, Va, USA, November 2003.

[12] Q. H. Dang, A.-J. van der Veen, and A. Trindade, "Statistical analysis of a transmit-reference UWB wireless communication system," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 3, pp. 317–320, Philadelphia, Pa, USA, March 2005.

[13] M. Torlak and G. Xu, "Blind multiuser channel estimation in asynchronous CDMA systems," *IEEE Transactions on Signal Processing*, vol. 45, no. 1, pp. 137–147, 1997.

[14] R. Hoctor and H. Tomlinson, "Delay-hopped transmitted-reference RF communications," in *IEEE Conference on Ultra Wideband Systems and Technologies(UWBST '02)*, pp. 265–269, Washington, DC, USA, May 2002.

[15] 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," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 4, part 1, pp. 773–779, 2006.

[16] Z. Irahhauten, G. J. Janssen, H. Nikookar, A. Yarovoy, and L. P. Ligthart, "UWB channel measurements and results for office and industrial environments," in *IEEE International Conference on Ultra-Wideband (ICUWB '06)*, Waltham, Mass, USA, September 2006.

[17] R. Djapic, G. Leus, and A.-J. van der Veen, "Blind synchronization in asynchronous UWB networks based on the transmit-reference scheme," in *Proceedings of 38th Asilomar Conference on Signals, Systems, and Computers (ACSSC '04)*, vol. 2, pp. 1506–1510, Pacific Grove, Calif, USA, November 2004.

[18] A.-J. van der Veen, M. C. Vanderveen, and A. J. Paulraj, "Joint angle and delay estimation using shift-invariance properties," *IEEE Signal Processing Letters*, vol. 4, no. 5, pp. 142–145, 1997.

[19] R. Djapic, G. Leus, and A.-J. van der Veen, "Synchronization and detection for transmited reference UWB systems," in *Proceedings of the Asilomar Conference on Signals, Systems, and Computers (ACSSC '05)*, pp. 1084–1088, Pacific Grove, Calif, USA, October/November 2005.

Relja Djapic was born in Novi Sad, Serbia, in 1975. He received the Electrical Engineering degree from the University of Novi Sad, Serbia, in 2000, and the Ph.D. degree from TU Delft, The Netherlands, in 2006. His research interests include signal processing for communication systems, blind source separation, and synchronization schemes in wireless ad hoc networks and ultra-wideband systems. He is currently with TNO-ICT, Delft, where he is working on broadband communication techniques over cable and coax.

Geert Leus was born in Leuven, Belgium, in 1973. He received the Electrical Engineering degree and the Ph.D. degree in applied sciences from the Katholieke Universiteit, Leuven, Belgium, in June 1996 and May 2000, respectively. He was a Research Assistant and a Postdoctoral Fellow of the Fund for Scientific Research, Flanders, Belgium, from October 1996 to September 2003. During that period, he was affiliated with the Electrical Engineering Department, Katholieke Universiteit Leuven, Leuven, Belgium. Currently, he is an Assistant Professor at the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. During the summer of 1998, he visited Stanford University, and from March 2001 to May 2002, he was a Visiting Researcher and Lecturer at the University of Minnesota, Minneapolis. His research interests are in the area of signal processing for communications. He received the 2002 and 2005 IEEE Signal Processing Society Best Paper Awards. He is a Member of the IEEE Signal Processing for Communications Technical Committee, an Associate Editor for IEEE Transactions on Signal Processing, IEEE Transactions on Wireless Communications, and the EURASIP Journal on Applied Signal Processing.

Alle-Jan van der Veen was born in The Netherlands in 1966. He received the Ph.D. degree (cum laude) from TU Delft in 1993. Throughout 1994, he was a Postdoctoral Scholar at Stanford University. At present, he is a Full Professor in Signal Processing at TU Delft. He is the recipient of the 1994 and 1997 IEEE Signal Processing Society (SPS) Young Author Paper Awards, and was an Associate Editor for IEEE Transactions on Processing (1998–2001), Chairman of IEEE SPS Signal Processing for Communications Technical Committee (2002–2004), and Editor-in-Chief of IEEE Signal Processing Letters (2002–2005). He is currently an Editor-in-Chief of IEEE Transactions on Signal Processing, and Member-at-Large of the Board of Governors of IEEE SPS. His research interests are in the general area of system theory applied to signal processing, and in particular algebraic methods for array signal processing, with applications to wireless communications and radio astronomy.

António Trindade was born in Lisbon, Portugal, in 1973. He graduated from IST, Technical University of Lisbon, in 1997, and continued to work as a Research Engineer at TU Delft, in cooperation with Nokia Research. From 2002 onwards, he was working as a Ph.D. student in the area of signal processing for mobile communications, in particular on the application of multiple antennas for WCDMA, in cooperation with Ericsson, and on the development of ultra-wideband communication systems. He is currently with ChipIdea Microelectronica, Lisbon. His interests are in ultra-wideband communication systems, cross-layer signal processing design and optimization, and MAC and PHY integration for low-power transceivers.