# On Digital Receiver Design for Transmitted Reference UWB

Yiyin Wang, Geert Leus and Alle-Jan van der Veen
Delft University of Technology, Fac. EEMCS, Mekelweg 4, 2628CD Delft, The Netherlands

***Abstract*—A complete channel estimation, synchronization and equalization scheme for a transmitted-reference (TR) ultra-wideband (UWB) system is proposed in this paper. The scheme is based on a data model, which admits moderate data rate and takes all kinds of interference into consideration: between pulses, frames and symbols. Three channel estimators are derived to achieve joint channel and timing estimation, namely the linear minimum mean square error (LMMSE) estimator, the least squares (LS) estimator and the matched filter (MF). We check the performance of different combinations of channel estimation and equalization schemes, and find out the best combination, i.e. the one providing a good trade-off between complexity and performance. This paper proves that a TR-UWB can provide a good performance at a moderate data rate, meanwhile having a low cost digital implementation.**

## I. INTRODUCTION

Ultra-wideband (UWB) techniques can provide high speed, low cost and low complexity wireless communications with the capability to overlay existing frequency allocations [1]. Since employing ultra-short low-duty-cycle pulses as information carriers, UWB systems suffer from stringent timing requirements [1] and complex multi-path channel estimation. Recent research works have focused on channel estimation and synchronization methods based on a low sampling rate [2][3] or for non-coherent systems, such as transmitted-reference (TR) systems [4][5]. The TR-UWB systems [4] have attracted increasing interest, as they greatly simplify the channel estimation and synchronization problems. A two-stage acquisition scheme for TR-UWB systems is proposed in [5], which employs two sets of direct-sequence (DS) code sequences to facilitate coarse timing and fine aligning. The scheme assumes no inter-frame interference (IFI) and no inter-symbol interference (ISI). In [6], a blind synchronization method for TR-UWB systems executes a MUSIC-kind search in the signal subspace to achieve high resolution timing estimation. However, the complexity of the algorithm is very high because of the matrix decomposition.

Recently, a multi-user TR-UWB system that admits not only inter-pulse interference (IPI) but also IFI and ISI is proposed in [7]. The synchronization for such a system is at low-rate sample level. The analog parts can run independently without any feedback control from the digital parts. In this paper, we have developed a complete channel estimation, synchronization and equalization scheme based on the data model modified from [7]. Moreover, the property of the circulant matrix in the data model is exploited to reduce the computational complexity. Different combinations of channel estimators and equalizers are evaluated to find the one with the best trade-off between performance and complexity. The results confirm that the TR-UWB system is a practical scheme that can provide moderate data rate communications at a low cost.

Notation: We use upper (lower) bold face letters to denote matrices (column vectors). $x(\cdot)$ ($x[\cdot]$) represents a continuous (discrete) time signal. $\mathbf{0}_{m\times n}$ ($\mathbf{1}_{m\times n}$) is an all-zero (all-one) matrix of size $m \times n$, while $\mathbf{0}_m$ ($\mathbf{1}_m$) is an all-zero (all-one) column vector of length $m$. $\star$, $\otimes$ and $|\cdot|$ indicate time domain convolution, kronecker product and absolute value. $(\cdot)^T$, $(\cdot)^H$ and $\|\cdot\|_F$ designate transposition, conjugate transposition, and Frobenius norm. All other notation should be self-explanatory.

This research was supported in part by STW under the Green and Smart Process Technologies Program (Project 7976).

## II. DATA MODEL

In a TR-UWB system, pairs of pulses (doublets) are transmitted in sequence. The first pulse in the doublet is the reference pulse, whereas the second one is the data pulse. One frame period $T_f$ holds one doublet. $N_f$ frames constitute one symbol period $T_s = N_f T_f$, which is carrying a symbol $s_i \in \{-1, +1\}$, spread by a chip code $c_j \in \{-1, +1\}, j = 1, 2, \ldots, N_f$, which is repeatedly used for all symbols. The two pulses in the doublet are separated by some delay interval, which can be different for each frame. The receiver employs multiple correlation branches corresponding to different delay intervals. To simplify the system, we use a single delay and one correlation branch. The integrate-and-dump (I&D) in the correlation branch integrates over an interval of length $T_{sam}$. As a result, one frame results in $P = T_f/T_{sam}$ samples, which is assumed to be an integer.

The received one-frame signal ($j$th frame of $i$th symbol) at the antenna output without noise is:

$$r(t) = h(t-\tau) + s_i c_j h(t - D - \tau) \tag{1}$$

where $\tau$ is the unknown timing offset and $h(t) = h_p(t) \star g(t)$, with $h_p(t)$ the UWB physical channel of length $T_h$, and $g(t)$ the pulse shape resulting from all the filter and antenna effects. Without loss of generality, the unknown timing offset $\tau$ in (1) is in the range of one symbol period, $\tau \in [0, T_s)$, since we propose to find the symbol boundary before acquiring the package header (see Section III). Then, $\tau$ can be decomposed as: $\tau = \delta \cdot T_{sam} + \epsilon$, where $\delta = \lfloor \frac{\tau}{T_{sam}} \rfloor \in \{0, 1, \ldots, L_s - 1\}$ denotes the sample level offset in a symbol range with $L_s = N_f P$, the symbol length in terms of number of samples, and $\epsilon \in [0, T_{sam})$ presents the fractional offset. Sample level synchronization consists of estimating $\delta$. The influence of $\epsilon$ will be absorbed in the data model and becomes invisible as we will show later.

With the received signal $r(t)$, the correlation branch of the receiver computes:

$$x[n] = \int_{(n-1)T_{sam}+D}^{nT_{sam}+D} r(t) r(t-D) dt \tag{2}$$

Taking $\epsilon$ into consideration, we can define the channel correlation function as:

$$R(\Delta, m) = \int_{(m-1)T_{sam}}^{mT_{sam}} h(t-\epsilon) h(t-\epsilon-\Delta)\, dt \quad m = 1, 2, \ldots \tag{3}$$

where $h(t) = 0$, when $t > T_h$ or $t < 0$. Then, we can define the $P_h \times 1$ channel energy vector $\mathbf{h}$ with entries $h_m$ as:

$$h_m = R(0, m) + R(2D, m + \frac{D}{T_{sam}}), \quad m = 1, \ldots, P_h \tag{4}$$

where $P_h = \lceil T_h/T_{sam} \rceil$ is the channel length in terms of number of samples and $R(0, m)$ is always non-negative. Although $R(2D, m + \frac{D}{T_{sam}})$ is always very small compared to $R(0, m)$, we still keep it to

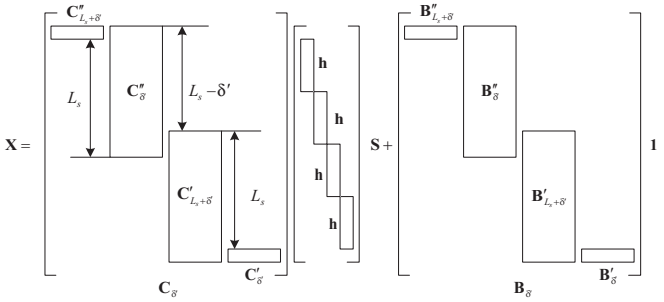


Fig. 1. The data model structure of $\mathbf{X}$

make the model more accurate. And the $P_h \times 1$ bias vector $\mathbf{b}$ with entries $b_m$ is defined as:

$$b_m = R(D, m) + R(D, m + \frac{D}{T_{sam}}), \quad m = 1, \ldots, P_h \qquad (5)$$

Note that these entries will change as a function of $\epsilon$, although $\epsilon$ is not visible in the data model. Using (4) and (5), $x[n]$ can be represented as:

$$x[n] = \begin{cases} s_i c_j h_{n-\delta} + b_{n-\delta} & n = \delta+1, \delta+2, \ldots, \delta+P_h; \\ 0 & \text{elsewhere} \end{cases}$$

where the bias term is caused by the IPI and is independent of the data symbols and the chip code.

Now let's extend the data model to $2N_s$ symbols. We assume the channel length $P_h$ is no longer than the symbol length $L_s$, $P_h < L_s$. A single symbol with timing offset $\tau$ will then spread over at most three adjacent symbol periods. Define $\mathbf{x}_k = \Big[x[kL_s - L_s + 1], x[kL_s - L_s + 2], \ldots, x[kL_s]\Big]^T$, which is an $L_s$-long sample vector. By stacking such received sample vectors into a $2L_s \times (2N_s - 1)$ matrix $\mathbf{X} = \begin{bmatrix} \mathbf{x}_k & \mathbf{x}_{k+1} & \ldots & \mathbf{x}_{k+2N_s-2} \\ \mathbf{x}_{k+1} & \mathbf{x}_{k+2} & \ldots & \mathbf{x}_{k+2N_s-1} \end{bmatrix}$, we obtain the following decomposition:

$$\mathbf{X} = \mathbf{C}_{\delta'}(\mathbf{I}_4 \otimes \mathbf{h})\mathbf{S} + \mathbf{B}_{\delta'}\mathbf{1}_{4N_f \times (2N_s-1)} \qquad (6)$$

where $\mathbf{S} = \begin{bmatrix} s_{k-1} & s_k & \ldots & s_{k+2N_s-3} \\ s_k & s_{k+1} & \ldots & s_{k+2N_s-2} \\ s_{k+1} & s_{k+2} & \ldots & s_{k+2N_s-1} \\ s_{k+2} & s_{k+3} & \ldots & s_{k+2N_s} \end{bmatrix}$ and the structures of the other matrices are illustrated in Fig. 1. We first define a code matrix $\mathbf{C}$. It is a block-Sylvester matrix of size $(L_s + P_h - P) \times P_h$, whose columns are shifted versions of the extended code vector: $[c_1, \mathbf{0}_{P-1}^T, c_2, \mathbf{0}_{P-1}^T, \ldots, c_{N_f}, \mathbf{0}_{P-1}^T]^T$. The shift step is one sample. Its structure is shown in Fig. 2. The matrix $\mathbf{C}_{\delta'}$ of size $2L_s \times 4P_h$ is composed by four block columns, where $\delta = (L_s - \delta') \bmod L_s$, $\delta' \in \{0, 1, \ldots, L_s - 1\}$. The non zero parts of these four block columns result from splitting the code matrix $\mathbf{C}$ according to $\delta'$: $\mathbf{C}'_i(2L_s - i + 1 : 2L_s, :) = \mathbf{C}(1 : i, :)$ and $\mathbf{C}''_i(1 : L_s + P_h - P - i, :) = \mathbf{C}(i + 1 : L_s + P_h - P, :)$. The overlays between frames and symbols observed in $\mathbf{C}_{\delta'}$ indicate the existence of IFI and ISI. Then we define a bias matrix $\mathbf{B}$, which is of size $(L_s + P_h - P) \times N_f$ made up by shifted versions of the bias vector $\mathbf{b}$ with a shift step of $P$ samples, as shown in Fig. 2. The matrix $\mathbf{B}_{\delta'}$ of size $2L_s \times 4N_f$ also has four block columns, the non zero parts of which are obtained from the bias matrix $\mathbf{B}$ in the same way as $\mathbf{C}_{\delta'}$. Since the bias is independent of the data symbols and the chip code, it is the same for each frame. Each column of the resulting matrix $\mathbf{B}_{\delta'}\mathbf{1}_{4N_f \times (2N_s-1)}$ is the same and has a period of

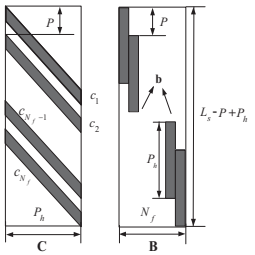


Fig. 2. The structure of the code matrix $\mathbf{C}$ and the bias matrix $\mathbf{B}$

$P$ samples. Defining $\mathbf{b}_f$ to be the $P \times 1$ bias vector for one such period, we have $\mathbf{B}_{\delta'}\mathbf{1}_{4N_f \times (2N_s-1)} = \mathbf{1}_{2N_f \times (2N_s-1)} \otimes \mathbf{b}_f$.

Because of the correlation at the receiver, the noise, which contaminates the output of the I&D, contains two components. One is the cross correlation between the noise and the data signal, and the other is the auto correlation of the noise. We ignore the cross term and only consider the auto term, whose statistical parameters can be estimated when there is no signal transmitted. Let us now assume the receiver collects additive white gaussian noise (AWGN) with double sided power spectral density $\frac{N_0}{2}$ and the prefilter to get rid of the noise outside of the band of interest has a processing bandwidth $B$. Then, the noise auto correlation term is also assumed to be AWGN, with zero mean and a variance $\sigma^2 = \frac{{N_0}^2 B T_{sam}}{2}$ [8].

## III. CHANNEL ESTIMATION, SYNCHRONIZATION AND EQUALIZATION

### A. Detection–signal or noise?

The first task of the receiver is to detect the existence of a signal. In order to separate the detection and synchronization problems, we assign the first segment of the training sequence to detection only. In this segment, we transmit all "+1" symbols and employ an all "+1" chip code. Assume the segment is $2M_1$ symbols long and the observation window is $M_1$ symbols. We collect the samples in one observation window, calculate a test statistic and examine whether it exceeds a threshold. If not, we jump into the next successive observation window of $M_1$ symbols long. In this way, we speed up our search procedure by jumping $M_1$ symbols. Once the threshold is exceeded, we skip the next $M_1$ symbols in order to be out of the first segment of the training sequence and we are ready to start the channel estimation and synchronization at the sample level. The sample level synchronization finds out the symbol boundary, which can later be used for symbol level synchronization to acquire the header. This two stage synchronization strategy decomposes a two-dimensional search into two one-dimensional searches, reducing the complexity.

### B. Channel estimation and sample level synchronization

*1) Bias estimation:* As we have seen in the asynchronous data model, the bias term is annoying. It doesn't have any useful information, but disturbs the signal. We will show that this bias seriously degrades the channel estimation performance later. The second segment of the training sequence consists of "+1,-1" symbol pairs employing a random chip code. The total length of the second segment should be $M_1 + 2N_s$ symbols, which includes the budget for jumping $M_1$ symbols after the detection. The "+1,-1" symbol pairs can be used for bias estimation as well as channel estimation. Since the bias is independent of the data symbols and the useful signal part has zero mean, due to the "+1,-1" training symbols, we can estimate the $L_s \times 1$ bias vector of one symbol long, $\mathbf{b}_s = \mathbf{1}_{N_f} \otimes \mathbf{b}_f$, as:

$$\hat{\mathbf{b}}_s = \frac{1}{2N_s}[\mathbf{x}_k \quad \mathbf{x}_{k+1} \quad \ldots \quad \mathbf{x}_{k+2N_s-1}]\mathbf{1}_{2N_s} \qquad (7)$$

*2) Channel estimation:* To take advantage of the second segment of the training sequence, we only pick odd (or even) columns of $\mathbf{X}$ in (6). Because all the odd (or even) columns depend on the same symbols, it leads to a simplification of (6):

$$\begin{aligned}\widetilde{\mathbf{X}} &= \begin{bmatrix} \mathbf{x}_k & \mathbf{x}_{k+2} & \dots & \mathbf{x}_{k+2N_s-2} \\ \mathbf{x}_{k+1} & \mathbf{x}_{k+3} & \dots & \mathbf{x}_{k+2N_s-1} \end{bmatrix} \\ &= [(\mathbf{C}'_{L_s+\delta'} + \mathbf{C}''_{L_s+\delta'}) \quad (\mathbf{C}'_{\delta'} + \mathbf{C}''_{\delta'})](\mathbf{I}_2 \otimes \mathbf{h})[-s_k \quad s_k]^T \mathbf{1}_{N_s}^T \\ &\quad + \mathbf{1}_{2\times N_s} \otimes \mathbf{b}_s \end{aligned} \tag{8}$$

Because we jump into this second segment of the training sequence after detecting the signal, we don't know whether the symbol $s_k$ is "+1" or "-1". Rewriting (8) in another form and employing the noise analysis result from the previous section leads to:

$$\widetilde{\mathbf{X}} = \mathbf{C}_s \mathbf{h}_{ss\delta} \mathbf{1}_{N_s}^T + \mathbf{1}_{2\times N_s} \otimes \mathbf{b}_s + \mathbf{N} \tag{9}$$

where $\mathbf{N}$ represents the noise matrix similarly defined as $\widetilde{\mathbf{X}}$, $\mathbf{C}_s$ is a known $2L_s \times 2L_s$ circulant code matrix, whose first column is $[c_1, \mathbf{0}_{P-1}^T, c_2, \mathbf{0}_{P-1}^T, \dots, c_{N_f}, \mathbf{0}_{L_s+P-1}^T]^T$, and the vector $\mathbf{h}_{ss\delta}$ of length $2L_s$ joins the timing and the channel information, which contains two channel energy vectors with different signs, $s_k\mathbf{h}$ and $-s_k\mathbf{h}$, located according to $\delta$:

$$\mathbf{h}_{ss\delta} = \begin{cases} \text{circshift}(\left[s_k\mathbf{h}^T, \mathbf{0}_{L_s-P_h}^T, -s_k\mathbf{h}^T, \mathbf{0}_{L_s-P_h}^T\right]^T, \delta) & \delta \neq 0 \\ \left[-s_k\mathbf{h}^T, \mathbf{0}_{L_s-P_h}^T, s_k\mathbf{h}^T, \mathbf{0}_{L_s-P_h}^T\right]^T & \delta = 0 \end{cases}$$

where $\text{circshift}(\mathbf{a}, n)$ circularly shifts the values in the vector $\mathbf{a}$ down by $n$ elements. According to (9) and assuming the channel energy has been normalized, the linear minimum mean square error (LMMSE) estimate of $\mathbf{h}_{ss\delta}$ then is:

$$\hat{\mathbf{h}}_{ss\delta} = \mathbf{C}_s^H (\mathbf{C}_s \mathbf{C}_s^H + \frac{\sigma^2}{N_s}\mathbf{I})^{-1} \frac{1}{N_s} (\widetilde{\mathbf{X}} - \mathbf{1}_{2\times N_s} \otimes \mathbf{b}_s) \mathbf{1}_{N_s} \tag{10}$$

Define

$$\hat{\mathbf{h}}_{s\delta} = \left[\hat{\mathbf{h}}_{ss\delta}(1:L_s) - \hat{\mathbf{h}}_{ss\delta}(L_s+1:2L_s)\right]/2 \tag{11}$$

then we get the symbol long LMMSE channel estimate as: $\hat{\mathbf{h}}_\delta = |\hat{\mathbf{h}}_{s\delta}|$. According to a property of circulant matrices, $\mathbf{C}_s$ can be decomposed as: $\mathbf{C}_s = \mathcal{F}\mathbf{\Lambda}\mathcal{F}^H$, where $\mathcal{F}$ is the normalized DFT matrix of size $2L_s \times 2L_s$ and $\mathbf{\Lambda}$ is a diagonal matrix with the frequency components of the first row of $\mathbf{C}_s$ on the diagonal. Hence, the matrix inversion in (10) can be simplified dramatically. Observing that $\mathbf{C}_s^H(\mathbf{C}_s\mathbf{C}_s^H + \frac{\sigma^2}{N_s}\mathbf{I})^{-1}$ is a circulant matrix, the bias term actually does not have to be removed in (10), since it is implicitly removed when we calculate (11). Therefore, we don't have to estimate the bias term explicitly for channel estimation and synchronization. The bias estimation can still be used for equalization though.

When the signal to noise ratio (SNR) is high, $\| \mathbf{C}_s\mathbf{C}_s^H \|_F \gg \| \frac{\sigma^2}{N_s}\mathbf{I} \|_F$, (10) can be replaced by:

$$\hat{\mathbf{h}}_{ss\delta} = \frac{1}{N_s} \mathcal{F}\mathbf{\Lambda}^{-1}\mathcal{F}^H (\widetilde{\mathbf{X}} - \mathbf{1}_{2\times N_s} \otimes \mathbf{b}_s) \mathbf{1}_{N_s} \tag{12}$$

It is a least squares (LS) estimator and equivalent to a deconvolution of the chip sequence in the frequency domain. On the other hand, when the SNR is low, $\| \mathbf{C}_s\mathbf{C}_s^H \|_F \ll \| \frac{\sigma^2}{N_s}\mathbf{I} \|_F$, (10) becomes:

$$\hat{\mathbf{h}}_{ss\delta} = \frac{1}{\sigma^2} \mathcal{F}\mathbf{\Lambda}^H\mathcal{F}^H (\widetilde{\mathbf{X}} - \mathbf{1}_{2\times N_s} \otimes \mathbf{b}_s) \mathbf{1}_{N_s} \tag{13}$$

which boils down to a matched filter (MF). The MF can also be processed in the frequency domain. The LMMSE estimator in (10), the LS estimator in (12) and the MF in (13) all have a similar computational complexity. However for the LMMSE estimator, we have to estimate $\sigma^2$ and normalize the channel energy. Fig. 3 indicates the symbol long channel estimate $\hat{\mathbf{h}}_\delta$ with bias removal and $|\hat{\mathbf{h}}_{ss\delta}(1:L_s)|$ without bias removal, where $\hat{\mathbf{h}}_{ss\delta} = \mathbf{C}_s^H(\mathbf{C}_s\mathbf{C}_s^H + \frac{\sigma^2}{N_s}\mathbf{I})^{-1}\frac{1}{N_s}\widetilde{\mathbf{X}}\mathbf{1}_{N_s}$, when SNR is high. The MF for $\hat{\mathbf{h}}_\delta$ has a higher noise floor than the LMMSE estimator for $\hat{\mathbf{h}}_\delta$, since its output is the correlation of the channel energy vector with the code autocorrelation function. The bias term lifts the noise floor of the channel estimation resulting from the LMMSE estimator and distorts the estimation, while it doesn't have much influence on the MF. The stars in the figure present the real channel parameters as a reference. The positions of the peaks in Fig. 3 indicate the timing information and the areas around the peaks are the most interesting part. Although the LMMSE estimator without bias suppresses the estimation errors over the whole symbol period, it has a similar performance as all the other estimators in the interesting part.

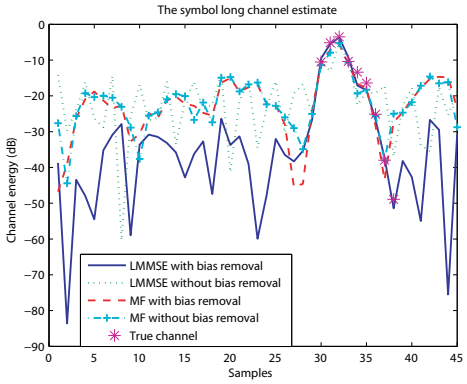


Fig. 3. Channel Estimation, when SNR is 18 dB

*3) Sample level synchronization:* The channel estimate $\hat{\mathbf{h}}_\delta$ is one symbol long. But we know that the true channel will generally be much shorter than the symbol length. We have to detect the part that contains most of the channel energy, and cut out the other part to be robust against noise. This basically means that we have to estimate the unknown timing $\delta$. Define the search window length as $L_w$ in terms of the number of samples. The optimal length of the search window depends on the channel energy profile and the SNR. We will show the influence of different window lengths on the $\delta$ estimation in the next section. Define $\hat{\mathbf{h}}_{w\delta} = [\hat{\mathbf{h}}_{s\delta}^T, -\hat{\mathbf{h}}_{s\delta}^T(1:L_w-1)]^T$, as long as $L_w > 1$. The $\delta$ estimate is then given by:

$$\hat{\delta} = \underset{\delta}{\text{argmax}} \left| \sum_{n=\delta+1}^{\delta+L_w} \hat{\mathbf{h}}_{w\delta}[n] \right| \tag{14}$$

*4) Equalization and symbol level synchronization:* Based on the channel estimate $\hat{\mathbf{h}}_\delta$ and the timing estimate $\hat{\delta}$, we can select the set of $P$ samples (the frame length in terms of number of samples) which has most energy to construct a simple MF, thereby taking advantage of low cost and easy implementation. In another way, we could also construct a ZF equalizer or a LMMSE equalizer according to the data model (6) to resolve the IFI and the ISI to achieve a better performance with a higher computational complexity. The estimated bias can be used here. We skip the details due to lack of space.

Till now, the sample level synchronization confirms the boundaries of the symbols. However it is not able to explore the boundary of the training header, since it just employs pairs of "+1,-1" symbols. After the sample level synchronization, the demodulation is triggered. The third segment of the training sequence is a known training symbol pattern. Once we find the matching symbol pattern, we can distinguish the training header. Symbol level synchronization is then accomplished.

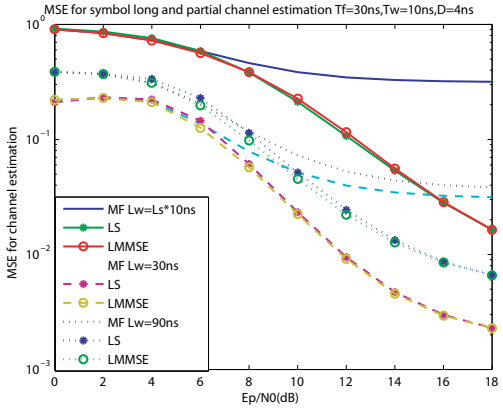


Fig. 4. MSE Performance for symbol long channel estimation

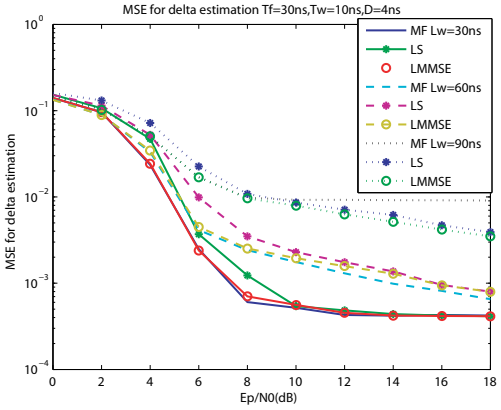


Fig. 5. MSE Performance for $\delta$ estimation, $L_w$ is 30 ns

## IV. SIMULATION RESULTS

The performance of different combinations of channel estimation and equalization schemes is evaluated for a single user and single delay TR-UWB system. We use a Gaussian second derivative pulse that is 0.2 ns wide. The delay interval $D$ is 4 ns. The symbol block length $2N_s$ is 30. The chip code is a pseudonoise (PN) sequence. The code length $N_f$ is 15. The frame period $T_f$ is 30 ns. The IEEE UWB channel model CM3 [9] is employed and truncated to 90 ns, which represents a NLOS channel. The oversample rate $P$ is 3, which results in $T_{sam} = 10$ ns.

500 Monte Carlo runs evaluate the mean squared error (MSE) of $\hat{\mathbf{h}}_\delta$ vs. SNR. In each run, the timing offset and the channel are randomly generated. The results for the symbol long estimates and the $L_w$ long estimates are shown in Fig. 4. The MF curves always have the highest noise floor, since its output is the convolution of the channel energy vector with the code autocorrelation function. The performance gap for symbol long estimates between the LS (LMMSE) estimator and the MF is large. When we concentrate on the channel estimates in a limited range, such as 30 ns and 90 ns, the gap between the MF and the LS (LMMSE) estimator is smaller. The normalized MSE $E[|(\hat{\delta} - \delta)/L_s|^2]$ for $\delta$ estimation is also assessed with different values of $L_w$ based on different channel estimators. Observing Fig. 5, the $\delta$ estimates based on MF, LS and LMMSE channel estimates with the same $L_w$ have similar performance and the optimal $L_w$ is 30ns. The MSE for $\delta$ with $L_w$=30ns is saturated after the SNR reaches 10dB, since we use NLOS channels, and the first path may not be the strongest. Meanwhile the differences of the MSE for channel estimation based on different methods are quite small around 10dB. As a result, we choose the MF as the channel estimator. Furthermore, combinations of the MF channel estimator with the different equalizers are investigated. We employ $L_w = 30$ns for synchronization. Fig. 6 shows the BER performance. The MF equalizer is 2dB worse than the ZF and the LMMSE equalizer, which employ 90ns long channel estimates. The optimal combination considering cost and performance would be a MF channel estimator with a ZF equalizer. According to the results above, we can remark that the IFI and the ISI after the I&D is not so serious in our simulation setup, since the channel energy attenuates exponentially and one frame contains most of the energy. The LMMSE estimator has the potential to handle more serious IFI and ISI. The effects of the bias on the BER performance can be ignored, but it has to be taken into account for the channel estimation (done implicitly). When we want to shorten the frame length to achieve a higher data rate, more interference will be generated. We then need a more accurate data model to handle this interference.

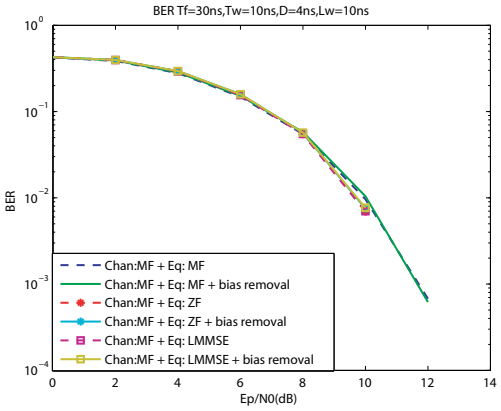


Fig. 6. BER Performance for CM3

## REFERENCES

[1] L. Yang and G.B. Giannakis, "Ultra-wideband communications: An idea whose time has come," *IEEE Signal Process. Mag.*, vol. 21, pp. 26–54, Nov. 2004.

[2] Z. Tian and G.B. Giannakis, "A GLRT approach to data-aided timing acquisition in UWB radios –Part I: Algorithms," *IEEE Trans. Wireless Commun.*, vol. 4, pp. 2956–2967, Nov. 2005.

[3] J. Kusuma, I. Maravic, and M. Vetterli, "Sampling with finite rate of innovation: channel and timing estimation for uwb and gps," in *Proc. IEEE Int. Conf. Communications*, Anchorage, AK, May 2003, vol. 5, pp. 3540–3544.

[4] R. Hoctor and H. Tomlinson, "Delay-hopped transmitted-reference RF communications," in *Proc. IEEE Conf. UWB Systems & Technologies*, Baltimore, MD, May 2002, pp. 265–269.

[5] S. Aedudodla, S. Vijayakunmaran, and T.F. Wong, "Timing acquisition in ultra-wideband communication systems," *IEEE Trans. Veh. Technol.*, vol. 54, pp. 1570–1583, Sept. 2005.

[6] R. Djapic, G. Leus, and A.J. van der Veen, "Blind synchronization in asynchronous multiuser uwb networks based on the transmit-reference scheme," in *Proc. IEEE Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2004, vol. 2, pp. 1506–1510.

[7] Q.H. Dang and A.J. van der Veen, "A decorrelating multiuser receiver for Transmit-Reference UWB systems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, pp. 431–442, Oct. 2007.

[8] Q.H. Dang, A.J. van der Veen, and A. Trindade, "Statistical analysis of a transmit-reference UWB wireless communication system," in *Proc. IEEE Int. Conf. Acoustic, Speech, and Signal Proc.*, Philadelphia, PA, March 2005, vol. 3, pp. 317–320.

[9] J.R. Foerster, "Channel modeling sub-committee report final," Tech. Rep. IEEE P802.15-02/368r5-SG3a, IEEE P802.15 Working Group for WPAN, 2002.