Chapter 2

Timing Synchronisation for IR-UWB Communication Systems

Rshdee Alhakim, Kosai Raoof and Emmanuel Simeu

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/48663

1. Introduction

The interest for Ultra Wide Band (UWB) technology is growing fast especially in the short-range indoor wireless communication, for example, in wireless personal area networks (WPAN). The basic concept is to transmit and receive baseband impulse waveform streams of very low power density and ultra-short duration pulses (typically at nanosecond scale). These properties of UWB give rise to fine time resolution, rich multipath diversity, low probability of detection, enhanced penetration capability, high user-capacity, and potential spectrum compatibility with existing narrowband systems [1]. However, one of the most critical challenges in enabling the unique benefits of UWB transmissions is timing synchronization, because the transmitted pulses are narrow and have low power density under the noise floor [2].

Timing synchronization in wireless communication systems typically depends on the sliding correlator between the received signal and a transmit-waveform template (Clean Template). In Impulse-Radio Ultra-Wideband (IR-UWB) devices however, this approach is not only sub-optimum in the presence of rich resolvable multipath channel, but also incurs high computational complexity and long synchronization time [2, 3]. Some research for improving the synchronization performance for IR-UWB systems has been reported in [4-9]. Each of these approaches requires one or more of the following assumptions: 1) the absence of multipath; 2) the absence of time-hopping (TH) codes; 3) the multipath channel is known; 4) high computational complexity and long synchronization time; and 5) degradation of bandwidth and power efficiency. Timing with Dirty Templates (TDT) is an efficient synchronization approach proposed for IR-UWB, introduced in [10-13]. This technique is based on correlating the received signal with "dirty template" extracted from the received waveforms. This template is called dirty; because it is distorted by the unknown channel and by the ambient noise. TDT allows the receiver to enhance energy capture even when the

multipath channel and the Time-Hopping (TH) spreading codes are both unknown. Consequently, TDT approach contributes to enhance synchronization performance for IR-UWB and to reduce receiver structure complexity [14-15].

In general, the synchronization system consists of three units: signal detection, timing acquisition and tracking. Figure 1 depicts metaphorically the synchronization system block. Signal detection is the first unit, for deciding if the signal received is desired UWB signal or noise only [16]. Timing Acquisition unit is, a coarse synchronization, employed to find approximately a starting point of each received symbol and to reduce the timing error to within a fraction of UWB pulse duration [10-12]. The third step is a tracking to maintain and lock the satisfactory synchronization in the presence of timing offset variations in the received waveform, as a result of oscillator drifts or transmitter-receiver motion (Doppler effects) [17-18].

This chapter is organized as follows. Section 2 introduces the UWB signal model and presents the dirty template (DT) technique. Section 3 describes the DT detection of UWB signals in the presence of ambient noise and dense multipath channel. The Neyman-Pearson (NP) theorem is applied to set an optimal threshold and decide if the UWB signal is present or not. Then, the performance of the dirty template detector is evaluated in terms of the detection property and the false alarm property versus the threshold for different values of signal-to-noise ratio (SNR) and data-aided bit number. Section 4 extracts a timing acquisition estimator based on the dirty template algorithms. The performance of this estimator will be improved by modifying the structure of the cross-correlation operation. The focus of Section 5 is on the third unit of the synchronization system: Tracking. The delay locked loop DLL is a fundamental tracking technique used for maintaining the satisfactory synchronization and reducing timing error in UWB systems. DLL structure is derived and its loop parameters are selected to optimize noise and transient performance while taking Doppler effects into account. Section 6 ends the chapter with some conclusions.

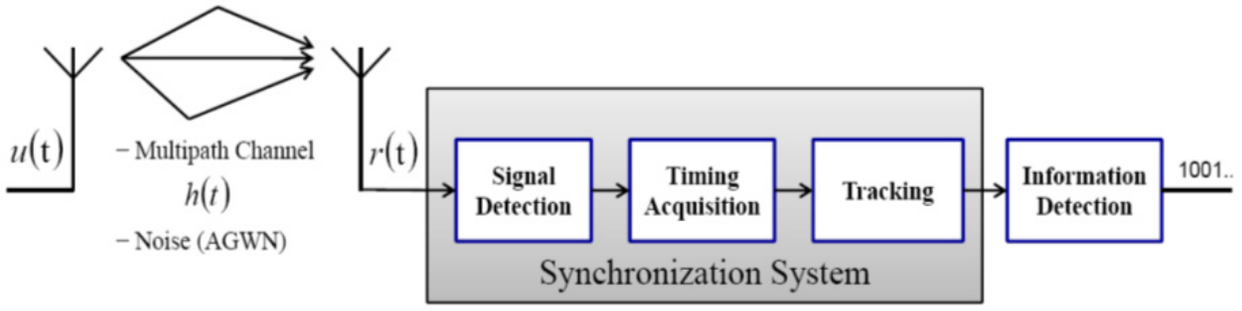

Figure 1. Synchronization System Block

2. UWB signal modelling

Consider an impulse radio UWB-IR system, where every symbol is transmitted over T_s period that consists of N_f pulses over N_f frames (one pulse per frame). Every frame of duration T_f contains N_c chips. The symbol waveform of duration $T_s := T_f N_f$ is $p_T(t) =$

$\sum_{j=0}^{N_f-1} p(t - jT_f - c_jT_c)$, where $p(t)$ is un ultra-short pulse, that has duration $T_p(\ll T_f)$, and $T_c := T_f/N_c$ is the chip duration with pseudo-random time-hopping (TH) codes $\{c_j\} \in [0, N_c - 1], j \in [0, N_f - 1]$ (see Figure 2).

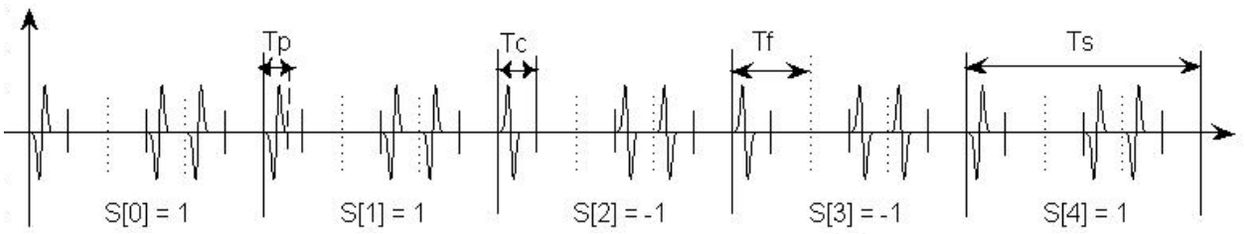

Figure 2. TH-UWB signal with PAM modulation ($N_f = 3, N_c = 2$, TH codes=[0, 1,0])

We consider that the symbol waveform has unit energy ($\int {p_T}^2(t)dt = 1$). By focusing on pulse amplitude modulation (PAM), where the information-bearing symbols $s[n] \in \{\pm 1\}$ are modelled as binary independent and identically distributed (i.i.d.) with energy $\mathcal{E}_s$ spread over N_f frames. The transmitted UWB waveform is then given by [15]:

$$u(t) = \sqrt{\mathcal{E}_s} \sum_{n=0}^{\infty} s[n] p_T(t - nT_s) \tag{1}$$

The signal $u(t)$ propagates through a multipath channel, whose impulse response $h(t) := \sum_{l=0}^{L-1} \alpha_l \delta(t - \tau_l)$ has coefficients α_l and delays τ_l, obeying $\tau_l < \tau_{l+1}$.The timing offset τ_0 refers to the first arrival time. To isolate τ_0, we define $\tau_{l,0} \coloneqq \tau_l - \tau_0$ as the relative time delay of each channel tap, where $\tau_{L-1,0}$ is channel delay spread. To avoid inter-symbol interference (ISI), T_f is selected to satisfy the following condition: $T_f \geq \left(c_{N_f-1} - c_0\right) T_c + \tau_{L-1,0} + T_p$ [2]. The received pulse within each frame is $p_r(t) := \sum_{l=0}^{L-1} \alpha_l p(t - \tau_{l,0})$; The waveform in the output of the receiver antenna is:

$$r(t) = \sqrt{\mathcal{E}_s} \sum_{n=0}^{\infty} s[n] p_R(t - nT_s - \tau_0) + w(t) \circledcirc \tag{2}$$

where $p_R(t)$ is the received waveform of each symbol:

$$p_R(t) := \sum_{j=0}^{N_f-1} p_r\left(t - jT_f - c_jT_c\right) = \sum_{l=0}^{L-1} \alpha_l p_T\left(t - \tau_{l,0}\right) \tag{3}$$

and $w(t)$ represents the bandpass filtered zero-mean additive white Gaussian noise (AGWN) with power spectral density (PSD) $N_0/2$ and with double sided bandwidth B ($\gg 1/T_s$) [11]. The timing offset τ_0 could be represented by: $\tau_0 = n_sT_s + \tau$, where $n_s = \lfloor \tau_0/T_s \rfloor \geq 0$ denotes the symbol-level timing offset, $\lfloor \cdot \rfloor$ represents the floor operation, and $\tau \in [0, T_s)$ the fine-timing offset [2]. By substituting τ_0 in (2), the received signal can be expressed by:

$$r(t) = \sqrt{\mathcal{E}_s} \sum_{n=0}^{\infty} s[n] p_R(t - nT_s - n_sT_s - \tau) + w(t) \circledcirc \tag{4}$$

Under mistiming ($\tau_0 \neq 0$), any T_s-long received segment of $r(t)$ can be represented by parts of two consecutive symbols (see Figure 3, the received UWB pulses are represented by triangles to illustrate the rich multipath effects of the channel), as bellow:

$$x(t+nT_s) = w(t+nT_s) + \begin{cases} \sqrt{\mathcal{E}_s}s[n-n_s-1]p_R(t+T_s-\tau): t \in [0,\tau) \\ \sqrt{\mathcal{E}_s}s[n-n_s]p_R(t-\tau) \quad\quad : t \in [\tau, T_s) \end{cases} \tag{5}$$

where $x(t)$ is the received segment of duration T_s.

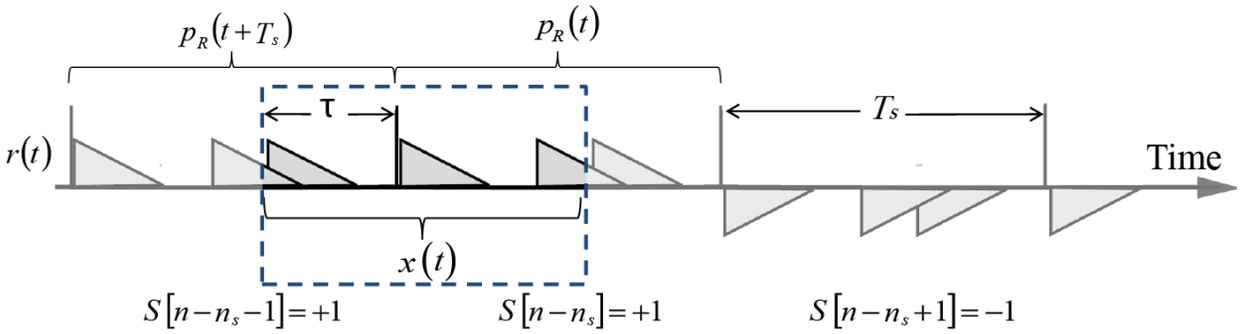

Figure 3. T_s-long observed received Symbol $x(t)$

The original TDT method proposed in [2], is defined as the cross-correlation between the pairs of successive received segments of duration T_s. That means: one segment in each pair of successive received symbols serves as the reference template signal for the other segment, as shown below:

$$R_{x,x}[n] = \int_0^{T_s} x(t+nT_s)x(t+(n+1)T_s)\,dt \tag{6}$$

$$= \int_0^{T_s}\left\{w(t+nT_s) + \sqrt{\mathcal{E}_s}\sum_{m=0}^{1} s[n-n_s-m]p_R(t+mT_s-\tau)\right\} \times \left\{w(t+(n+1)T_s) + \sqrt{\mathcal{E}_s}\sum_{m=0}^{1} s[n-n_s-m+1]p_R(t+mT_s-\tau)\right\}dt$$

$$= \breve{\omega}[n] + A\int_0^{T_s} p_R^2(t+T_s-\tau)\,dt + B\int_0^{T_s} p_R^2(t-\tau)\,dt$$

where the sampled noise $\breve{\omega}[n]$ explained in [11], is composed of three terms, two of them are the result of correlation between the symbol and the noise, and the third term is between shifted noises (noise-cross-noise). These terms could be treated approximately as mutually uncorrelated zero mean Gaussian noises. So $\breve{\omega}[n]$ could represent as zero-mean Gaussian noise, and its statistic properties are given as [11]:

$$\breve{\omega}[n] \sim \mathcal{N}\left(0, {\sigma_w}^2 = \frac{{N_0}^2}{2}BT_s + N_0\underbrace{\frac{\mathcal{E}_s\mathcal{E}_{max}}{\mathcal{E}_r}}\right) \tag{7}$$

Where: $\mathcal{E}_{max} = \int_0^{T_s} p_R(t)^2\,dt$ represents T_s-long received segment energy. The dirty template sample $R_{x,x}[n]$ is thus given as:

$$R_{x,x}[n] = \breve{\omega}[n] + A\int_{T_s-\tau}^{T_s} p_R^2(t)\,dt + B\int_0^{T_s-\tau} p_R^2(t)\,dt \tag{8}$$

$$where: \begin{cases} A = \mathcal{E}_s s[n-n_s-1].s[n-n_s] \\ B = \mathcal{E}_s s[n-n_s].s[n-n_s+1] \end{cases}$$

The possible values of A and B are exhibited in Table 1.

$s[n-1]$	+1	+1	-1	-1	+1	-1	+1	-1
$s[n]$	+1	-1	-1	+1	+1	-1	-1	+1
$s[n+1]$	-1	-1	+1	+1	+1	-1	+1	-1
A	+1	-1	+1	-1	+1	+1	-1	-1
B	-1	+1	-1	+1	+1	+1	-1	-1

Table 1. Possible values of A and B in (8)

The TDT technique can be applied to UWB receiver even in the presence of time hopping TH codes or Inter-Frame Interference (IFI), because the received segment and its dirty template contain the same TH codes and IFI properties regardless of the unknown channel characteristics (e.g. unknown time offset), but with the condition of the absence of ISI. Moreover, this method exploits the rich multipath diversity provided by UWB channels, and does not need to neither estimate the propagation channel nor generate clean correlation template at the receiver. Consequently, it reduces receiver complexity with high received energy capture.

3. Signal detection

At the receiver, detecting the received signal $r(t)$ and identifying the symbol-level offset n_s are achieved by using the dirty template data-aided (DA) algorithms. Supposing that we send M_1 training symbols, have the same value $\{s[n] = 1\}_{n=0}^{M_1-1}$ (or all training bits are equal -1), they are thus received during $[\tau_0, \tau_0 + M_1 T_s]$. With the presence of M_1 training symbols, the successive received symbols have the same values. In this case, by looking on the Table 1 and taking the corresponding values of A and B, then substituting them in (8), $R_{x,x}[n]$ becomes:

$$R_{x,x}[n] = \mathcal{E}_s \int_0^{T_s} P_R^2(t)\, dt \; + \widetilde{\omega_{d1}}[n] = \underbrace{\mathcal{E}_s \mathcal{E}_{max}}_{\mathcal{E}_r} + \widetilde{\omega_{d1}}[n] = \mathcal{E}_r + \widetilde{\omega_{d1}}[n] \tag{9}$$

where $\widetilde{\omega_{d1}}[n]$ is the dirty template noise in the case of signal detection. Assume that under the noise-only hypothesis $\mathcal{H}_0$, we observe M_1 (i.i.d.) samples of $R_{x,x}[n] = \widetilde{\omega_{d0}}[n]$ for $n = 0,1,\dots,M_1 - 1$, while under signal-present hypothesis $\mathcal{H}_1$, we observe $R_{x,x}[n] = \mathcal{E}_r + \widetilde{\omega_{d1}}[n]$ for $n = n_s, n_s + 1, \dots, n_s + M_1 - 1$. The detection problem is to distinguish between these two hypotheses:

$$\begin{aligned} \mathcal{H}_0 &: R_{x,x}[n] = \widetilde{\omega_{d0}}[n] && n = 0,1,\dots,M_1 - 1 \\ \mathcal{H}_1 &: R_{x,x}[n] = \mathcal{E}_r + \widetilde{\omega_{d1}}[n] && n = n_s, n_s + 1, \dots, n_s + M_1 - 1 \end{aligned} \tag{10}$$

From (7), we could find easily that: In the case of the hypothesis $\mathcal{H}_0$, the UWB signal is absent and the signal-noise terms become zeros. Thus:

$$\widetilde{\omega_{d0}}[n] \sim \mathcal{N}\left(0, {\sigma_0}^2 = \frac{{N_0}^2}{2} B T_s\right)$$

$$\widetilde{\omega_{d1}}[n] \sim \mathcal{N}\left(0, {\sigma_1}^2 = N_0 \mathcal{E}_r + \frac{{N_0}^2}{2} BT_s\right) \tag{11}$$

The Neyman-Pearson (NP) detector, mentioned in [16], decides $\mathcal{H}_1$ and declares the presence of $r(t)$ if the likelihood ratio exceeds a threshold : ($J(n) > \eta$), where η is a threshold set by the desired probability of false alarm (FA).

$$J(n) = \frac{p(R_{x,x}[n];\mathcal{H}_1)}{p(R_{x,x}[n];\mathcal{H}_0)} > \eta \tag{12}$$

$$J(n) = \frac{\frac{1}{(2\pi{\sigma_1}^2)^{M_1/2}} exp\left[-\frac{1}{2{\sigma_1}^2}\sum_{m=n}^{n+M_1-1}\left(R_{x,x}[m]-\mathcal{E}_r\right)^2\right]}{\frac{1}{(2\pi{\sigma_0}^2)^{M_1/2}} exp\left[-\frac{1}{2{\sigma_0}^2}\sum_{m=n}^{n+M_1-1}\left(R_{x,x}[m]\right)^2\right]} > \eta$$

$$J(n) = \left(\frac{\sigma_0}{\sigma_1}\right)^{M_1} exp\left[-\frac{1}{2{\sigma_1}^2}\sum_{m=n}^{n+M_1-1}\left(R_{x,x}[m] - \mathcal{E}_r\right)^2 + \frac{1}{2{\sigma_0}^2}\sum_{m=n}^{n+M_1-1}\left(R_{x,x}[m]\right)^2\right] > \eta$$

$$J(n) = \left(\frac{\sigma_0}{\sigma_1}\right)^{M_1} exp\left[\sum_{m=n}^{n+M_1-1}\left\{\left(-\frac{1}{2{\sigma_1}^2} + \frac{1}{2{\sigma_0}^2}\right)R_{x,x}^2[m] + \frac{\mathcal{E}_r}{{\sigma_1}^2}R_{x,x}[m] - \frac{{\mathcal{E}_r}^2}{2{\sigma_1}^2}\right\}\right] > \eta$$

Taking the logarithm of both sides does not change the inequality, so that:

$$ln\left(J(n)\right) = M_1 ln\left(\frac{\sigma_0}{\sigma_1}\right) + \sum_{m=n}^{n+M_1-1}\left\{\left(-\frac{1}{2{\sigma_1}^2} + \frac{1}{2{\sigma_0}^2}\right)R_{x,x}^2[m] + \frac{\mathcal{E}_r}{{\sigma_1}^2}R_{x,x}[m] - \frac{{\mathcal{E}_r}^2}{2{\sigma_1}^2}\right\} > ln(\eta)$$

$$\sum_{m=n}^{n+M_1-1}\left\{\left(-\frac{1}{2{\sigma_1}^2} + \frac{1}{2{\sigma_0}^2}\right)R_{x,x}^2[m] + \frac{\mathcal{E}_r}{{\sigma_1}^2}R_{x,x}[m]\right\} > ln(\eta) - M_1 ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{M_1{\mathcal{E}_r}^2}{2{\sigma_1}^2}$$

$$\sum_{m=n}^{n+M_1-1}\left\{\frac{{\sigma_1}^2-{\sigma_0}^2}{2\mathcal{E}_r{\sigma_0}^2}R_{x,x}^2[m] + R_{x,x}[m]\right\} > \left(ln(\eta) - M_1 ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{M_1{\mathcal{E}_r}^2}{2{\sigma_1}^2}\right)\frac{{\sigma_1}^2}{\mathcal{E}_r}$$

From (11) we have $({\sigma_1}^2 - {\sigma_0}^2 = N_0\mathcal{E}_r)$, then

$$\sum_{m=n}^{n+M_1-1}\left\{\frac{N_0}{2{\sigma_0}^2}R_{x,x}^2[m] + R_{x,x}[m]\right\} > \left(ln(\eta) - M_1 ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{M_1{\mathcal{E}_r}^2}{2{\sigma_1}^2}\right)\frac{{\sigma_1}^2}{\mathcal{E}_r}$$

By replacing the value of ${\sigma_0}^2$ from (11) into the last equation, the test statistic T will be:

$$T = \sum_{m=n}^{n+M_1-1}\left\{R_{x,x}^2[m] + N_0BT_sR_{x,x}[m]\right\} > \underbrace{\left(ln(\eta) - M_1 ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{M_1{\mathcal{E}_r}^2}{2{\sigma_1}^2}\right)\frac{{\sigma_1}^2N_0BT_s}{\mathcal{E}_r}}_{\xi} \tag{13}$$

The Neyman-Pearson (NP) detector in (13) decides $\mathcal{H}_1$ and declares the presence of signal if the test statistic T exceeds a threshold $T > \xi$ .The architecture of the dirty template detector (13) is shown in Figure 4. To verify the performance of the detection, first the false alarm property P_{FA} and the detection property P_D are identified as following:

$$P_{FA} = Pr\{T > \xi; \mathcal{H}_0\}_{\circledcirc} \tag{14}$$

$$P_D = Pr\{T > \xi; \mathcal{H}_1\}_{\circledcirc} \tag{15}$$

But it is not easy to find the probability that the test statistic value T proposed in (13), exceeds a threshold ξ ($Pr\{T > \xi\}$) by analytical means, so we must exploit a Monte Carlo

simulation. We plot the results of the Monte Carlo evaluation of $Pr\{T > \xi\}$ versus ξ in Figures 5-7 for different values of SNR and data-aided number. The solid curve represents false alarm property $Pr\{T > \xi; \mathcal{H}_0\}$ and the dashed curve represents detection property $Pr\{T > \xi; \mathcal{H}_1\}$. From these figures we could determine the optimal value of the threshold ξ where the difference between the detection property and the false property is maximized $(max(P_D - P_{FA})|_\xi)$. Furthermore, Figure 5 shows the interference area between the two curves, this area provides an indication on possible errors. When we reduce the false property by increasing ξ, the detection property will be simultaneously reduced. Figure 6-7 show that by increasing the SNR or data aided number, the two curves are clearly separated, which decreases the interference area. That is why the detection error property is reduced and the detection performance is enhanced, but these improvements are in the price of power efficiency or detector design simplification.

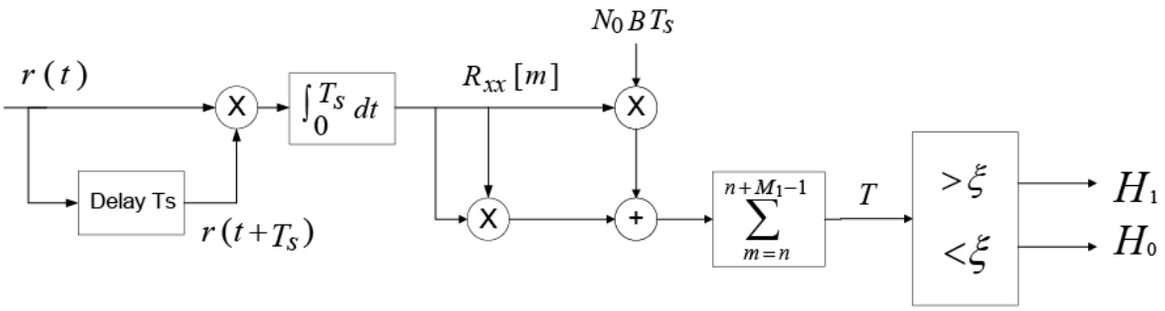

Figure 4. Block diagram of detection model for Dirty Template System

Another way of representing the detection performance of a Neyman-Pearson detector is to plot the detection property P_D versus the false alarm property P_{FA}. Each point on the curve corresponds to a value of (P_{FA}, P_D) for a given threshold ξ. As we have already found when ξ increases, P_{FA} and P_D decrease. This type of plot is called the receiver operating characteristic (ROC), shown in Figures 8-9. The ideal detector is when $P_D = 1$ for any value of P_{FA}. So it is clear that when the SNR or the data aided number increases the ROC curve approaches to the ideal case and the detection performance is improved.

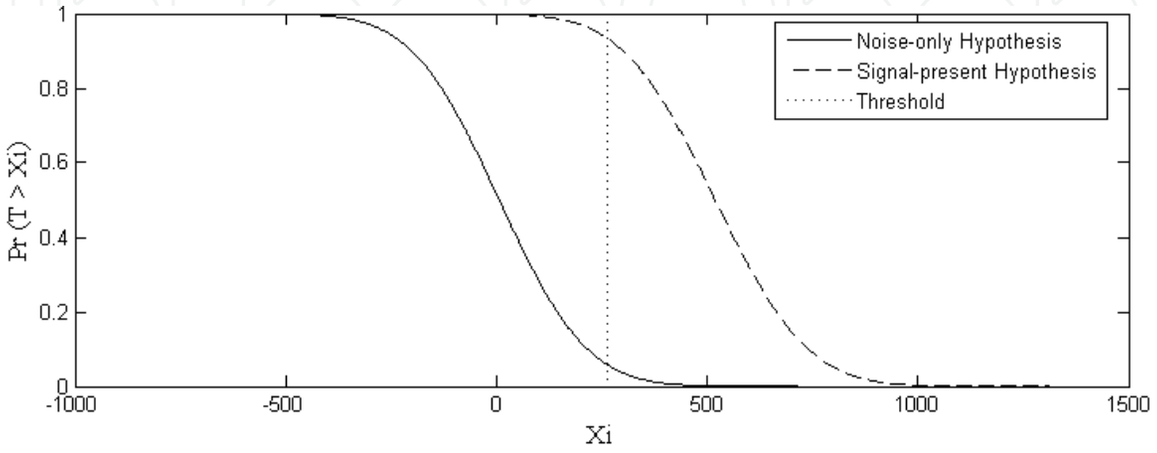

Figure 5. Monte Carlo simulation of $Pr\{T > \xi\}$ for SNR= -5 dB & Data Aided Number =8

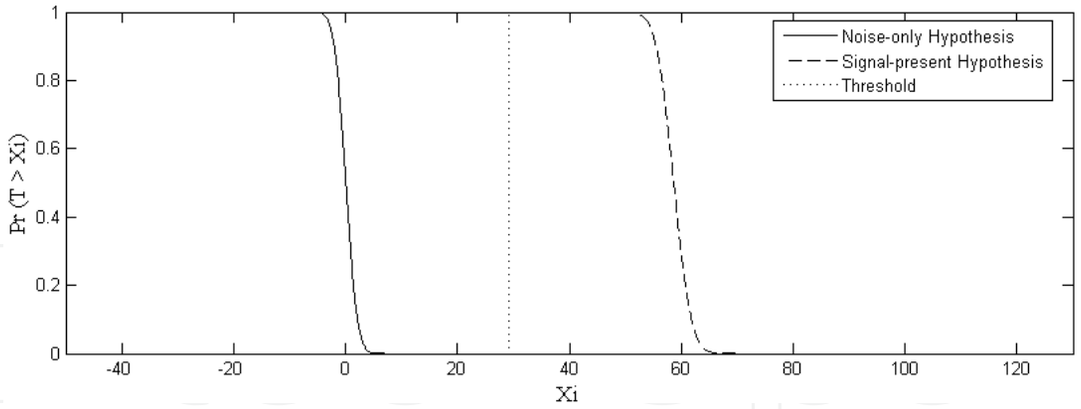

Figure 6. Monte Carlo simulation of $Pr\{T > \xi\}$ for SNR= -5 dB & Data Aided Number =32

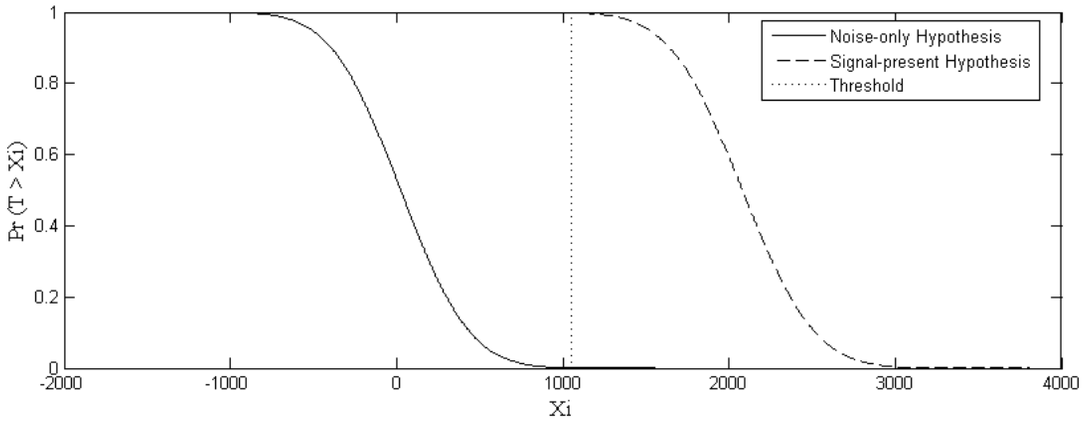

Figure 7. Monte Carlo simulation of $Pr\{T > \xi\}$ for SNR= 5 dB & Data Aided Number =8

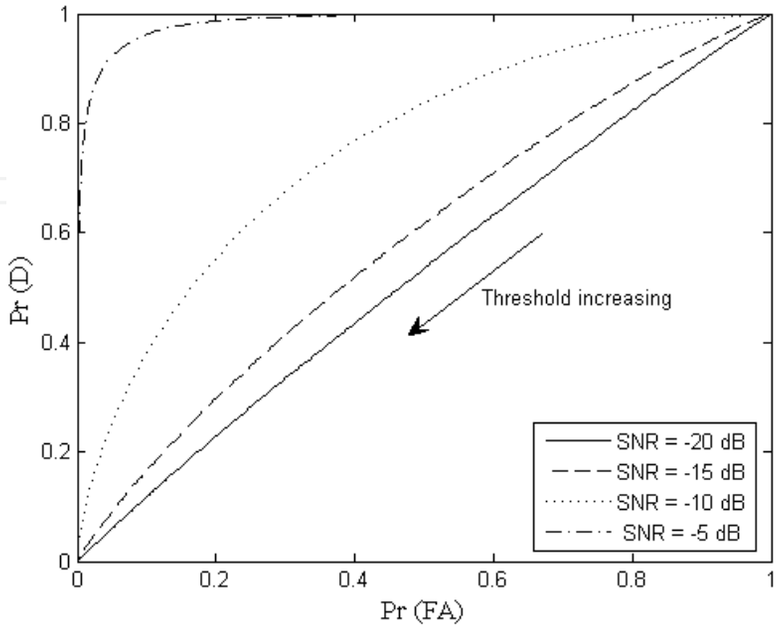

Figure 8. ROC for Data Aided Number =8

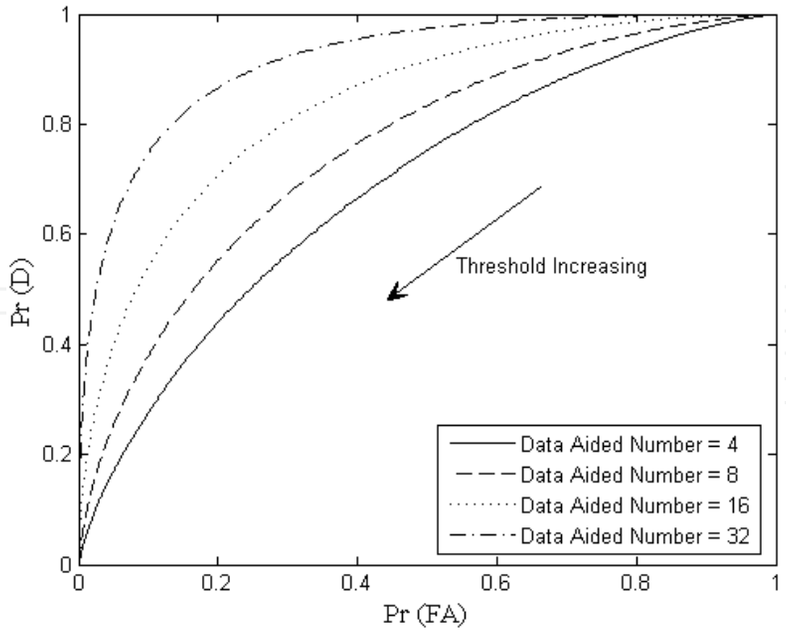

Figure 9. ROC for SNR= -10 dB

In this simulation, we select the pulse $p(t)$ as the third derivative of the Gaussian function with unit energy and duration $T_p \approx 1$ ns. Each symbol contains $N_f = 24$ frames with duration $T_f = 100$ ns. After the signal detection is done, the symbol-level timing offset $\hat{n}_s$ can be estimated via a line search to maximize the test statistic $T(n_s)$,as show below:

$$\hat{n}_s = arg\ max\ T(n_s); \qquad T(n_s) = \sum_{m=n_s}^{n_s+M_1-1}\left(R_{x,x}^2[m] + N_0 B T_s R_{x,x}[m]\right) \tag{16}$$

To avoid the influence of neighboring information-bearing symbols on $\hat{n}_s$ estimating, we separate between the data-aided symbols and information-bearing symbols by four-symbol pattern $(-1,-1,\ +1,+1)$. This pattern permit to exhibit a unique maximum peak at $n = \hat{n}_s$.

In this section, UWB signal detection has been carried out based on the Neyman-Pearson theory in (13) (Figure 4). Then, detection performance of the proposed method has been evaluated by Matlab simulation results (Figures 5-9). The analysis shows that the detection performance of the dirty template approach is improved by increasing the signal-to-noise ratio (SNR) or data-aided (DA) number. But these improvements reduce the power efficiency and complicate the receiver design. After detecting the UWB signal, the symbol-level timing offset estimation relies on searching the best statistics T in (16).

In practice, before the signal detection system runs for the first time, the value of ξ in (13) should be adjusted by sending train of data-aided symbols and by utilizing temporarily feedback loop, the optimal value of the threshold ξ is determined where the difference between the detection property and the false property is maximized ($max(P_D - P_{FA})|_\xi$). For the bandwidth B in (13), it is specified by The UWB standards approved by the FCC (e.g. $B \approx 5-7$ GHz for indoor applications) [19]. And for the background noise energy N_0 in (13), it could be estimated by listening to the environment during noise-only period.

4. Timing acquizition

Supposing that signal detection and symbol-level offset estimation are correctly achieved, the new time offset is confined within one symbol duration: $\tau_0 = \tau \in [0, T_s)$. Thus the objective of this section is to estimate the timing offset τ_0 based on dirty template algorithms in both data aided (DA) and non-data aided (NDA) modes. The original TDT acquisition method, proposed in [2], is based on searching a peak in the output of the correlation between the received signal and a dirty template, as shown below:

$$\hat{\tau}_0 = \underset{m}{arg\ max}\ \frac{1}{M_2} \sum_{n=0}^{M_2-1} \left(\int_0^{T_s} x(t + nT_s + \tau_m).x(t + (n+1)T_s + \tau_m)\, dt \right)^2 \tag{17}$$

where M_2 is the size of the successive received segments, used for accomplishing the TDT acquisition operation, $\tau_m = m\Delta_t \in [0, T_s)$; Δ_t represents the size of the increment, and m denotes the number of increments. Since the synchronization accuracy can be improved by reducing Δ_t, but in the price of fast estimation timing. Practically, in DA mode, the form of training symbols used for achieving fast acquisition is as follows [2]:

$$s[n] = \begin{cases} -1, & if(n \bmod 4) = 0, or, 1 \\ +1, & if(n \bmod 4) = 2, or, 3 \end{cases} : n \in [0, M_2 - 1] \tag{18}$$

where (X mod Y) denotes the modulo operation, where X and Y are both real. We can explain TDT in (17) by another way: when we do the cross-correlation between signals with its sliding replica, we obtain the unique maximum point at $\tau_0 = 0$. The TDT use the same principle to achieve the synchronization.

However, we can notice from Figure 10, in the UWB received signal, the presence of gaps between the received symbol and its dirty template. These gaps may cause multiple maxima points around the peak (optimal point) at the correlator output in (17), so the estimation error of the timing synchronization may increase. To avoid this problem, we modify the structure of the cross-correlation operation by adding a suitable window filter. In this case, the timing estimation in (17) can be achieved, as follows:

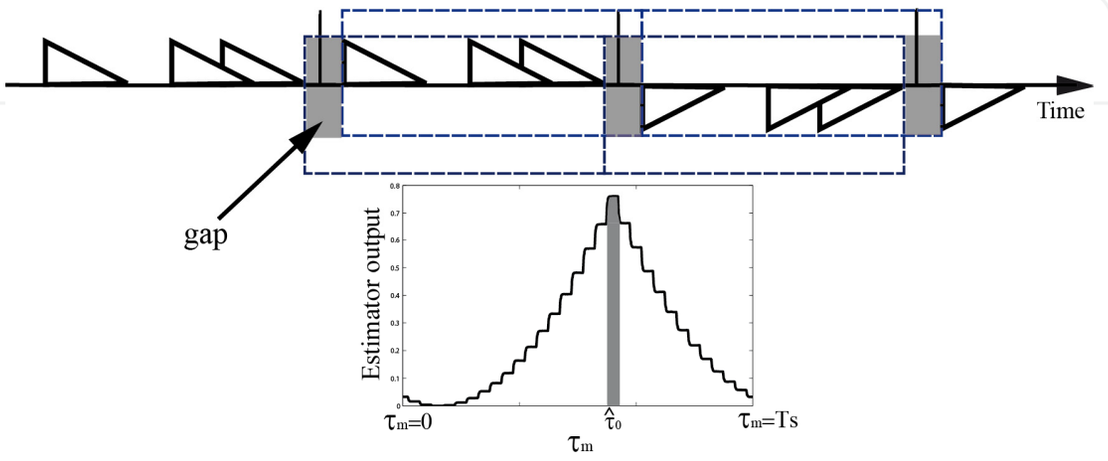

Figure 10. Timing reference illustration

$$\hat{\tau}_0 = \underset{m}{arg\,max}\ \frac{1}{M_2}\sum_{n=0}^{M_2-1}\left(\int_0^{T_s} x(t+nT_s+\tau_m).x(t+(n+1)T_s+\tau_m).W(t)dt\right)^2 \quad (19)$$

where the window $W(t)$ contains the information of spreading TH codes, as shown below:

$$W(t) = \sum_{j=0}^{N_f-1} c_j p\left(t - jT_f\right);\ p(t) = \begin{cases} 1 & si\ \ 0 \le t \le T_p + \tau_{L-1,0} \\ 0 & otherwise. \end{cases} \quad (20)$$

As the time of channel delay spread $\tau_{L-1,0}$ is unknown, we assume that the window width could be practically inside the range $\{T_c, 2T_c\}$. Figure 11 illustrates the structure of this window filter. We can notice that this window filter reduces the effect of these gaps at the UWB receiver. However, the window may slightly reduce the energy capture at the output of cross-correlation operation, but fortunately that doesn't impact on the estimation accuracy.

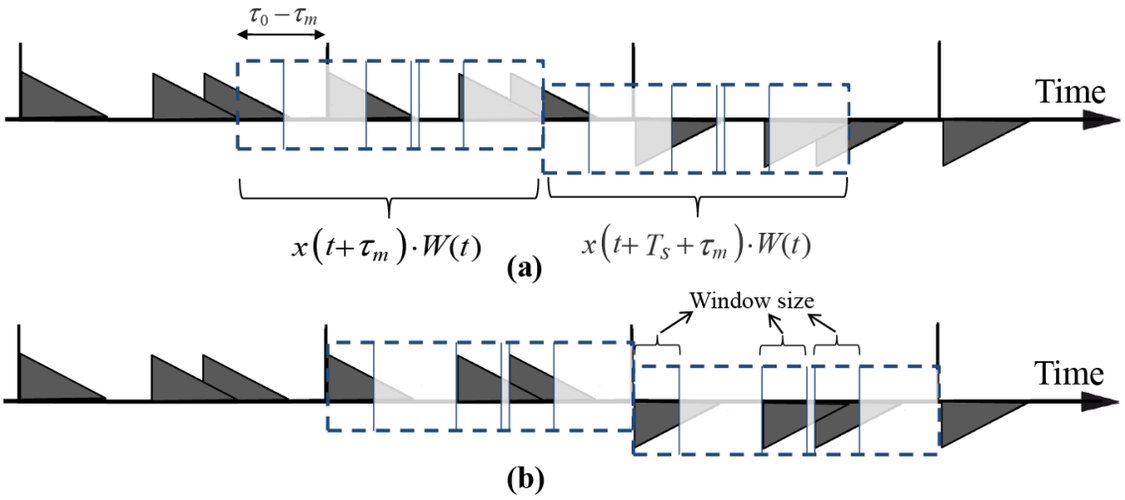

Figure 11. Window filter illustration, (a): $\boldsymbol{\tau_m \neq \tau_0}$, (b): $\boldsymbol{\tau_m = \tau_0}$

Figures 12-14 show the captured energy at the correlator output for various values of $\tau_m = m\Delta_t \in [0, T_s)$. Each of these figures has three graphs, one (*dashed line*) represents the DT estimator without any window, and the others (*dotted & solid lines*) represent the DT estimator with window size equal to $6T_c$ and $2T_c$ respectively. We have studied the UWB system without and with TH codes and with SNR= (20 or 3) dB. These figures show that the correlator output without window in (17) has multiple maxima points around the peak. Besides, they exhibit that decreasing the window size in (20) leads to decrease the maxima point number, until we reach a single maxima point (at $\tau_m \approx \tau_0$) in (19). Moreover, the window filter also contributes to reduce the unwanted noise effect on the timing estimation. For example, when window size equals $2T_c$, the received noise energy will be equal to (N_0) multiplied by the normalized window size ($2T_c/T_f$). In practice, this window $W(t)$ may simplify the implementation of timing operation, because we don't need to take all sampled points of the symbol to calculate the cross-correlation, the sample points inside the windows are sufficient.

To evaluate the performance of the proposed DT synchronizers for DA & NDA modes, we run a simulation in Matlab. In this simulation, we select the pulse $p(t)$ as the third

derivative of the Gaussian function with unit energy and duration $T_p \approx 1$ns. Each symbol contains $N_f = 24$ frames with duration $T_f = 100$ns. The simulations are performed in a Saleh-Valenzuela channel [20]. The parameters of this channel are chosen with (1/Λ, 1/λ, Γ, γ) = (2, 0.5, 30, 5) ns. The maximum channel delay spread of the channel is about 99 ns, and the inter-symbol interference (ISI) is negligible. We generate τ_0 randomly from a uniform distribution over $[0, T_s)$. We employ TH spreading codes of period N_f, which is generated from a uniform distribution over $[0, N_c - 1]$, with $N_c = 6$, and $T_c = 10$ns. We supposed the size of the increment Δ_t equal to $T_c/2$.

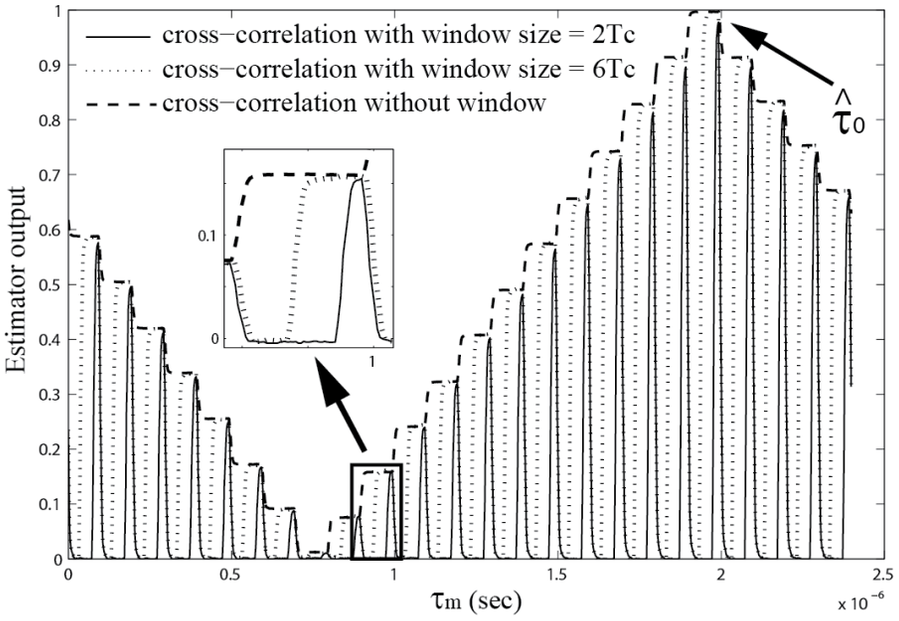

Figure 12. Correlator output energy (without TH codes, SNR=20 & M_2=16)

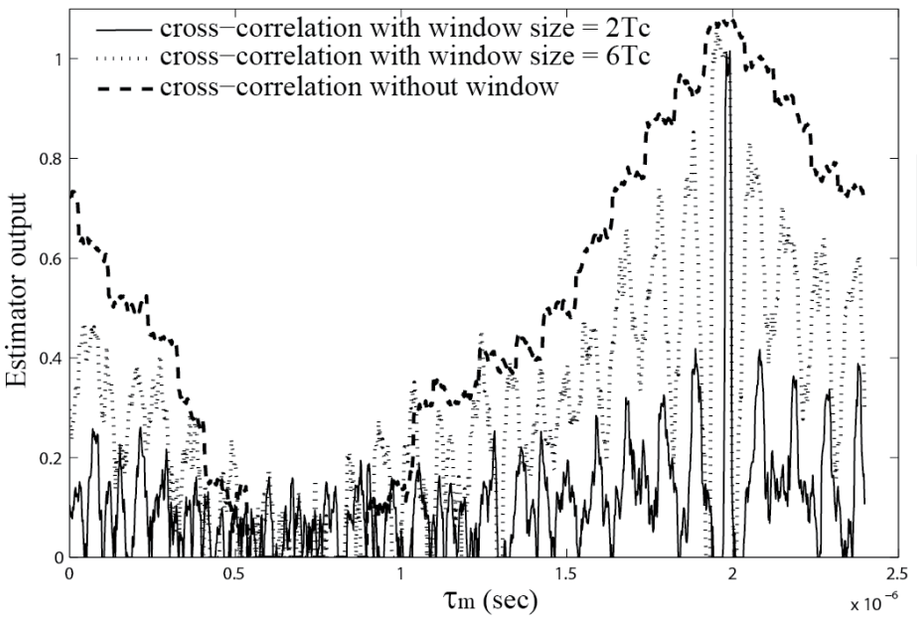

Figure 13. Correlator output energy (TH codes, SNR=3 & M_2=16)

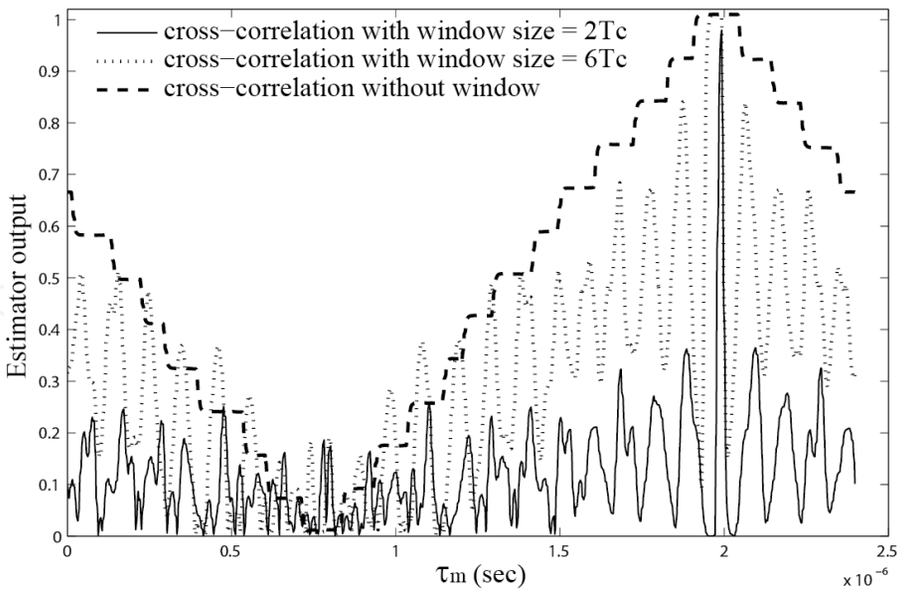

Figure 14. Correlator output energy (TH codes, SNR=20 & M_2=16)

We compare the accurate acquisition between the DT estimator without any window mentioned in (17) and the modified estimator with window size equal to $2T_c$ as mentioned in (19). Figures 15-16 show the comparison of mean square error (MSE) performance in DA & NDA modes with dirty templates for various values of M_2. In these figures, the MSE results are normalized by T_s^2, and plotted versus the signal-to-noise ratio (SNR) per pulse. The simulations confirm that, the DT estimator with window (*solid lines*) has higher timing estimation performance than the DT estimator without window (*dashed lines*). In general, as M_2 increases the normalized MSE decreases. Increasing SNR also helps to reduce the MSE. That means, when the observation number M_2 or the signal-to-noise ratio SNR is increased, the mean square error MSE is reduced and the estimation performance is improved. In addition, we can clearly notice that all curves with high SNR reach an error floor, which depends on the synchronization accuracy. This error floor can be reduced by decreasing the size of (Δ_t), but with increasing simulation time and computational complexity.

The results also show that the timing estimator for DA mode in Figure 15 has smaller MSE values and more accurate timing simulation than NDA mode in Figure 16 for the same SNR & M_2, but with less bandwidth efficiency. Moreover, Figure 15 shows that the DA estimator could be used with small training pattern size such as $M_2 = 4$, that helps to reduce the number of operations performed at the receiver as well as the synchronization time.

In this section, we present the original TDT algorithm used for achieving rapid, accurate and low-complexity timing acquisition. It relies on searching a peak in the output of the correlation between the received signal and a dirty template. But we have found the presence of multiple maxima points around the peak at the output of the correlator, and that may increase the complication to estimate the timing offset error (TOE). To avoid this problem, we modify the structure of the cross-correlation, by adding the suitable window

filter. This window contains information of the TH codes. This modified approach guarantees that we obtain a single maximal peak at the outside of the estimator and that improve the estimation error performance. The simulation results show the estimation performances of DA and NDA modes of the proposed DT method, and confirm that for the same size of correlation sequence pattern, the DA mode has high performance and fast timing, compared to NDA mode, but that is in the price of the bandwidth efficiency.

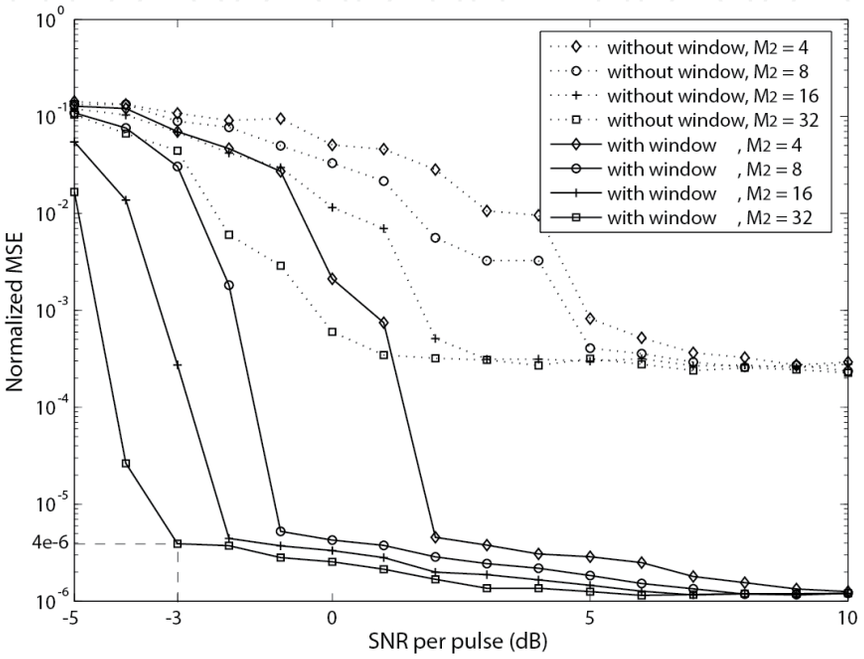

Figure 15. Normalized MSE vs. SNR per pulse for DA mode (window size = 2Tc)

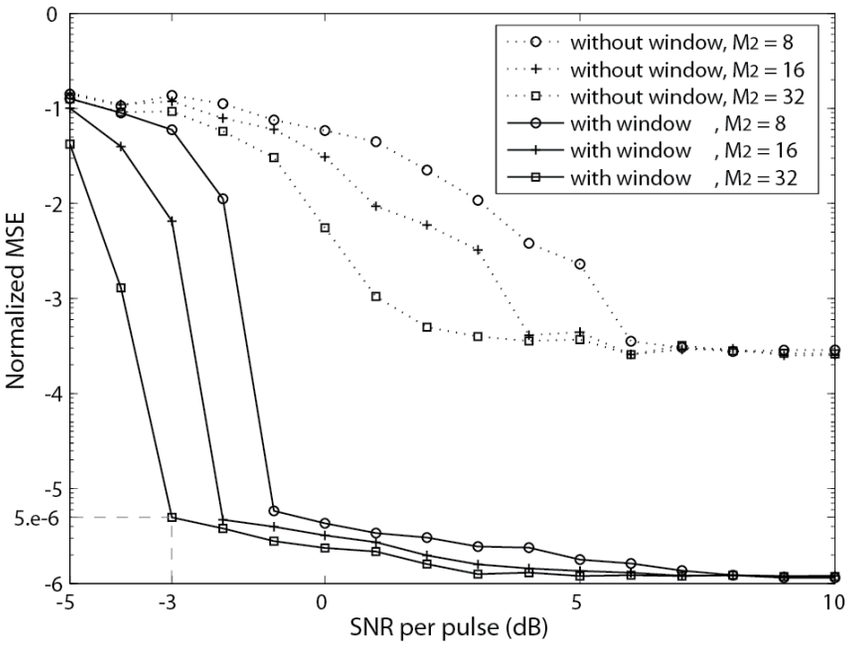

Figure 16. Normalized MSE vs. SNR per pulse for NDA mode (window size = 2Tc)

5. Tracking

The next step, after detecting UWB signals and estimating the timing offset, is to maintain and lock the satisfactory synchronization between the receiver and the received data signals. Tracking unit is used to alleviate the effects of timing offset variations due to transmitter-receiver motion (Doppler effects) and to maintain the transmission quality. For tracking purpose, we will use Delay-Locked Loop (DLL) approach which is considered as a fundamental tracking technique for UWB and spread-spectrum devices [21]. Several DLL schemes have been proposed to improve the tracking performance in UWB systems [22-24]. This section shows how to combine DLL with TDT, which enables adaptively following timing offset variations in the received signals and to enhance the BER performance.

Assuming that signal detecting and timing acquisition have been correctly achieved, so the timing error is confined within a fraction of UWB pulse duration and the tracking loop is "*in lock*" at the beginning of its operation. Figure 17 shows the proposed DLL method, known also as Early-Late Gate (ELG), consists of two correlator branches. The first branch is called *early correlator* because the received signal is cross-correlated with the advanced version of the reference signal $y(t+\Delta)$. And the second is called *late correlator*, where the received signal is correlated with the retarded version of the reference signal $y(t-\Delta)$. In other word, $y(t)$ is advanced and delayed in time by Δ. The correlation outputs are given by:

$$\begin{cases} R_e = \int_0^{T_s} r(t)y(t-\hat{\tau}+\Delta)\,dt \\ R_l = \int_0^{T_s} r(t)y(t-\hat{\tau}-\Delta)\,dt \end{cases} \tag{21}$$

where the reference signal $y(t)$ is extracted from the received waveform based on training (data-aided) sequence. We can employ the same training bits which have already been used to detect UWB signal (or even which have been used to acquire the timing). Assuming that M transmitted training bits equal one $\{s[n]=1\}_{n=1}^{M}$, in this case, the noise-free part of $r(t)$ in (2) consists of $p_R(t)$ replicas with spacing T_s. Hence, $y(t)$ can be approximately yield from $r(t)$ by averaging operation:

$$y(t) = \frac{1}{M}\sum_{n=1}^{M}\big(p_R(t-nT_s)+w_n(t)\big); \quad t\in[0,T_s) \tag{22}$$

Then:

$$y(t) \cong p_R(t)+\bar{w}(t): \quad \bar{w}(t)=\frac{1}{M}\sum_{n=1}^{M} w_n(t) \sim \mathcal{N}\left(0,\bar{\sigma}^2=\frac{N_0}{2M}\right) \tag{23}$$

We can notice that an increase in the number of data-aided bits M lead to reduce the noise variance $\bar{\sigma}^2$ and improve the tracking performance. Another way to decrease the amount of noise term $\bar{w}(t)$ inside the reference template $y(t)$ in (23) is to pass $y(t)$ through the window filter $W(t)$, mentioned in (20). Assume that the window size equals $1.5T_c$, the noise energy $\bar{w}(t)$ will be divided by $(1.5\,T_c/T_f)$.

In the presence of oscillator drifts or transmitter-receiver motion, this will produce the mistiming $(\tau \neq 0)$ between the reference and received signals. Figure 17 shows that DLL

tracking is performed by estimating the time offset τ between $r(t)$ and $y(t)$. Then, it compensates this offset by shifting the reference signal position to $y(t-\hat{\tau})$, where $\hat{\tau}$ denotes the DLL estimate of τ.

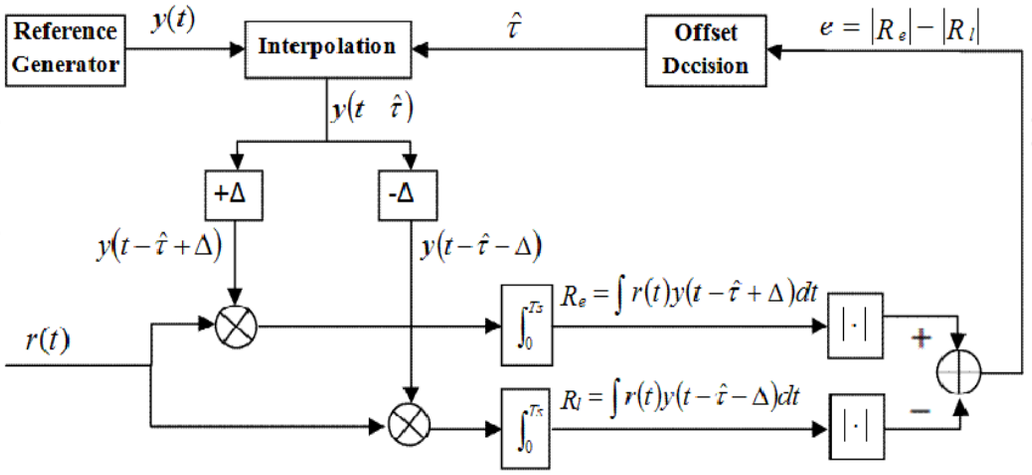

Figure 17. Delay-Locked Loop diagram

Supposing the time offset $\epsilon = \hat{\tau} - \tau$ is within UWB pulse duration (the tracking loop is "*in lock*"). Substituting (2) and (23) in (21), the cross-correlation outputs $R_{e,l}[\epsilon, n]$ become:

$$R_{e,l}[\epsilon, n] = \int_0^{T_s} \{s[n]p_R(t) + w_n(t)\} \times \{p_R(t - \epsilon \pm \Delta)\bar{w}(t)\}dt\,; \quad t \in [0, T_s) \tag{24}$$

Thus:

$$R_{e,l}[\epsilon, n] = s[n]\Gamma_{e,l}[\epsilon] + n_{e,l}[\epsilon, n] \tag{25}$$

where:

$$\Gamma_{e,l}[\epsilon] = \int_0^{T_s} p_R(t)p_R(t - \epsilon \pm \Delta)\, dt$$

Like as we have seen before in (6), the sample noise factors $n_{e,l}[\epsilon, n]$ is composed of three terms, two of them are the result of correlation between the symbol and the noise, and the third term is as noise-cross-noise [11]. $n_{e,l}[\epsilon, n]$ could represent as zero-mean Gaussian noises $n_{e,l} \sim \mathcal{N}(0, \sigma_{e,l}^2)$:

$$n_{e,l}[\epsilon, n] = \xi_1 + \xi_2 + \xi_3 \tag{26}$$

where:

$$n_{e,l}[\epsilon, n]: \begin{cases} \xi_1 = \int_0^{T_s} p_R(t - \epsilon \pm \Delta)w_n(t)\, dt \\ \xi_2 = \int_0^{T_s} s[n]p_R(t)\bar{w}(t)\, dt \\ \xi_3 = \int_0^{T_s} w_n(t)\bar{w}(t)\, dt \end{cases}$$

From Figure 17, the difference value (discriminator) between the correlator outputs in the ELG tracking serves as a timing error indicator. For example, if the discriminator (e) is close

to zero, then $y(t)$ and $r(t)$ are synchronized perfectly ($\hat{\tau} \approx \tau$). Otherwise, $y(t)$ is too late or too early. The discriminator error e at the output of DLL tracking is expressed by:

$$e[\epsilon, n] = R_e[\epsilon, n] - R_l[\epsilon, n]. \tag{27}$$

Substituting (25) in (27), we get:

$$e[\epsilon, n] = s[n] \underbrace{(\Gamma_e[\epsilon] - \Gamma_l[\epsilon])}_{\mathbb{S}[\epsilon]} + \underbrace{n_e[\epsilon, n] - n_l[\epsilon, n]}_{\eta[n]} = s[n] \cdot \mathbb{S}[\epsilon] + \eta[n] \tag{28}$$

where $\eta[n] = n_e[\epsilon, n] - n_l[\epsilon, n]$ is the equivalent additive noise for DLL tracking system, and $\mathbb{S}[\cdot]$ is the loop discriminator characteristic (S-curve), which is the useful term for tracking τ. From (28), we notice that the noise-free part of discriminator error (e) depends on the time offset ϵ and also on the data sign $s[n]$. To avoid the influence of the received date $s[n] \in \{\pm 1\}$ on the tracking performance, we add an absolute value block inside the DLL diagram. The absolute value method is chosen to make the tracking error indicator independent of the unknown data sign, is as follows:

$$e[\epsilon, n] = (R_e[\epsilon, n] - R_l[\epsilon, n]) \times sign(R_e[\epsilon, n] + R_l[\epsilon, n]) \tag{29}$$

Hence, the discriminator error e in (28) becomes as follows:

$$e[\epsilon, n] = \mathbb{S}[\epsilon] + \eta[n] \tag{30}$$

After series of mathematical manipulation, we found that the noise $\eta[n]$ could be treated as a zero mean Gaussian noise:

$$\eta[n] \sim \mathcal{N}\left(0, \sigma_\eta^2\right) \quad : \sigma_\eta^2 = \frac{N_0(M+1)}{M}(\mathcal{E}_{max} - \Gamma_{rr}(2\Delta)) \tag{31}$$

where $\mathcal{E}_{max} = \int_0^{T_s} p_R(t)^2\, dt$ represents T_s-long received segment energy, and $\Gamma_{rr}(2\Delta) = \int_0^{T_s} p_R(t)\, p_R(t - 2\Delta) dt$.

Assuming that the time offset ϵ is small to be inside the linear range of the S-curve $\mathbb{S}[\epsilon]$. In this case, the DLL behaves approximately like a linear filter ($\mathbb{S}[\epsilon] = A\epsilon$, where $A = \dot{\mathbb{S}}[0]$ is the gradient of the discriminator characteristic at $\epsilon = 0$). The DLL tracking system could be represented by the equivalent model shown in Figure 18. We focus now on designing a fit offset decision block, which is used to update the current estimate of time offset $\hat{\tau}$, to reduce then the offset error ϵ and to maintain timing synchronization. Let us consider the case where the dynamic input $\tau(t)$ has a ramp form:

$$\tau(t) = \tau_0 + mt \tag{32}$$

where τ_0 represents a constant time delay and m is a Doppler shift which is produced by a constant velocity relative to transmitter and receiver. For tracking efficiently a ramp input, the second-order DLL is selected [17]. Figure 18 depicts the suggested DLL structure with filter $F(s) = (1 + k_2 s)/(k_1 s)$, where the constants $\{k_1, k_2\}$ are loop parameters which will be found to optimize the tracking performance. Letting the closed loop function behave like

lowpass filter with noise equivalent bandwidth $B_L \ll 1/T_s$, which can be found as (see [17, Table 3.3.1]):

$$B_L = \frac{\xi\omega_n}{2}\left(1+\frac{1}{4\xi^2}\right) \tag{33}$$

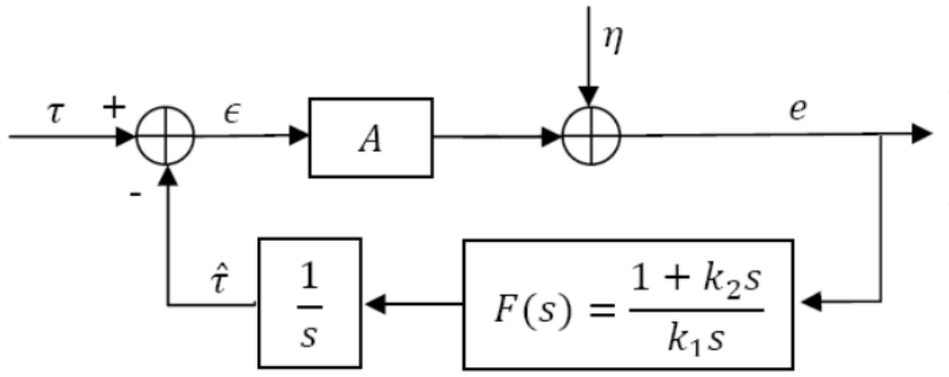

Figure 18. Linear Model of the second-order DLL

where the natural frequency $\omega_n = (A/k_1)^{1/2}$ and the loop damping ratio $\xi = (k_2\omega_n)/2$. With ramp input signal, DLL tracking system is subject to two error sources: additive noise error and transient error. So we are seeking to determine suitable values of the loop parameters $\{\omega_n, \xi\}$ for optimizing the performance criterion: minimizing noise energy effect as well as maintaining specified transient error. For selecting $\{\omega_n, \xi\}$, we apply Wiener-filter theory [25]. This optimization method allows us to minimize the following design criterion: $\sigma_\epsilon^2 + \lambda^2\epsilon_T^2$, where σ_ϵ^2 is the error energy (variance) due to noise, $\epsilon_T^2 = \int_0^\infty [\tau_0(t) - \tau(t)]^2 dt$ is the error energy due to transients, and λ is the Lagrange multiplier (considered as relative weight between noise and transient error energies). Based on Wiener-filter theory, the optimal second-order loop parameters are given as follows (see [17, Table 3.7.1]):

$$\omega_n^2 = m\,\lambda\,\sigma_\eta^{-1}, \quad \xi = \sqrt{2}/2 \tag{34}$$

Substituting (34) in (33), we get:

$$B_L = 0.53\omega_n \ll 1/T_s \ , \ k_2 = \sqrt{2}/\omega_n \ , \ k_1 = A/\omega_n^2 \tag{35}$$

This section has shown how to design a suitable UWB tracking system and to select its parameter values for maintaining the synchronization in the presence of ambient noise and Doppler effects. Figure 19 summarizes the synchronization system based on dirty template technique, which consists of three main blocks: signal detector, timing acquisition, and tracking. Relying on this synchronization system, the received symbols $\hat{s}[n]$ are demodulated, using:

$$\hat{s}[n] = \text{ sign } \left[\int_0^{T_s} r(t+nT_s)y(t-\hat{\tau})dt\right] \tag{36}$$

In the presence of the noise and with the channel variations, the DLL might be derived out of lock. We have mentioned in (2) that the received UWB signal $r(t)$ consists of a large

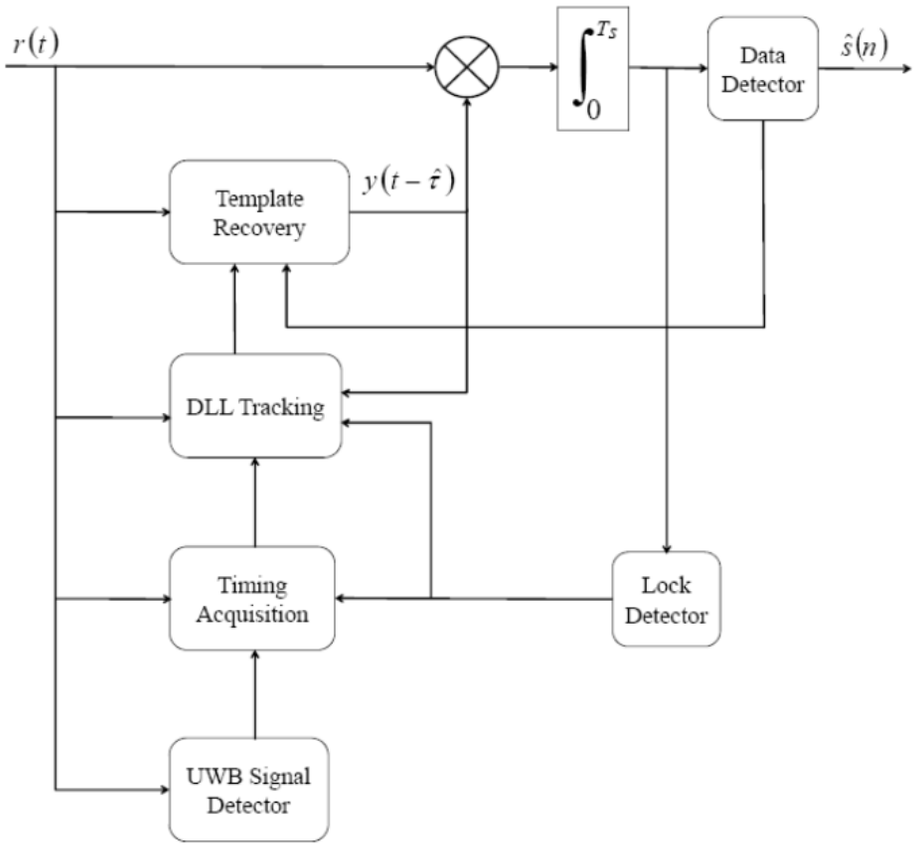

Figure 19. Block diagram of the synchronization system based on Dirty Template technique

number of channel paths. So when the DLL falls out of lock, it begins to track with a wrong path instead of the first one. It has been found in [24] that dirty template demodulator is able to perform efficiently symbol detecting, even if DLL is tracking another path close to the first one. In Figure 19, a lock detector is employed to measure capture energy at the correlator output and to compare it versus a threshold, for deciding if the DLL remains in lock or it is out of lock. Once the lose-lock signal is generated, the receiver turns back to the acquisition mode. After executing successfully the acquisition phase, the receiver enters again the tracking phase and DLL is brought back in lock. In UWB channel, due to the relative movement between the transmitter and the receiver (or object movement inside the signal propagation field), the multipath channel parameters vary slowly with time. Consequently, the reference signal $y(t)$, extracted in (22), becomes outdated and is not efficiently similar to T_s-long received segments. This leads to degrade gradually synchronization performance and increase BER. To avoid this problem, The dirty template $y(t)$, which is used for tracking the received signal and for estimating $\hat{s}[n]$, should be updated, as follows:

$$y(t) = \frac{1}{K}\sum_{n=1}^{K} \hat{s}[n] r(t + nT_s)\,; \quad t \in [0, T_s) \tag{37}$$

where K denotes the averaging size. Update of the template $y(t)$, does not need to be continuous, only periodic with a frequency $(1/T_y)$, where update period T_y must be less than the channel coherence time ($T_y \ll T_{coh}$). On the other hand, by considering that UWB

pulses are ideal candidates for indoor communication applications, where the expected speed of objects or users is typically in the order of $v = 1\ m/s$. With maximum carrier frequency for IR-UWB $f_c = 6$ GHz, the maximum Doppler spread is given as: $f_D = v/\lambda = (v/c) \times f_c = 20$ Hz, where c is the speed of light, and λ is the wavelength. As discussed in [26], the channel coherence time is about $T_{coh} = 0.423/f_c \approx 21$ msec. Thus, updating the DT reference $y(t)$ must be done periodically with T_y less than 21 msec.

For Matlab simulation, we select the pulse $p(t)$ as the second derivative of the Gaussian function with unit energy and duration $T_p \approx 1$ns. Each symbol contains $N_f = 10$ frames with duration $T_f = 100$ns, the symbol duration is $T_s = N_f T_f = 1$μs. We employ TH spreading codes of period N_f, which is generated from a uniform distribution over $[0, N_c - 1]$, with $N_c = 9$, and $T_c = 10$ ns. The DT reference signal $y(t)$ is enhanced by passing it through the filter $W(t)$ with window filter size = $1.5T_c$. The simulations are performed in a Saleh-Valenzuela channel [20]. Figures 20-21 show the mean square error (MSE) for various values of natural frequency ω_n and data-aided bit number M. In these figures, the MSE results are normalized by T_p^2, and plotted versus the signal-to-noise ratio (SNR) per pulse. Doppler effects are neglected $m = 0$. The simulations confirm that an increase in M or a decrease in ω_n lead to reduce noise energy effect and improve tracking performance. We can notice from Figure 20 that for SNR≥ 5 dB the MSE lines at various M are matched. In order to evaluate the transient performance, the input signal time varies as follows $\tau(t) = mt$ with $m = T_p/100T_s$. We select $\lambda = 10^7$, and $\sigma_\eta^2 = 10^{-16}$; thus, the optimum natural frequency in (34) is $\omega_n = 0.1$ GHz. Since DLL operates digitally with sampling rate T_s, we have $\tau(n) = 0.01nT_p$. Figure 22 presents DLL behavior in the presence of input time variations $\tau(n)$ for different ω_n. It is seen that DLL follows the input signal $\tau(n)$ rapidly when the natural frequency increases. Another way of representing DLL transient performance is exhibited in

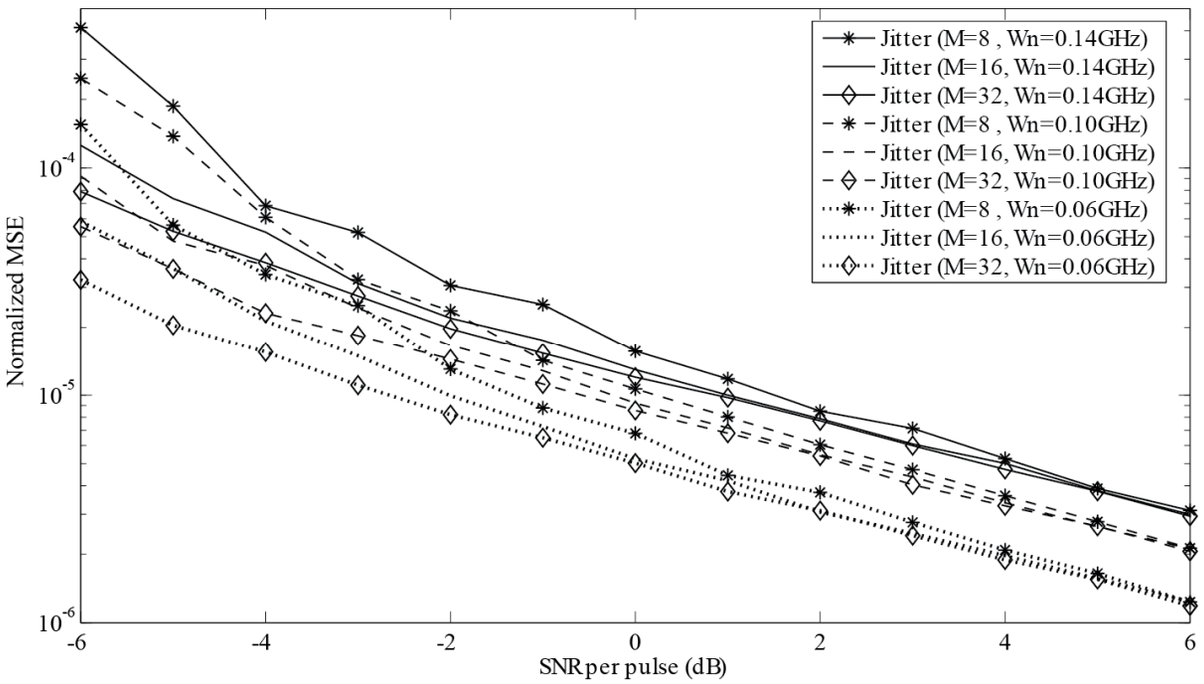

Figure 20. Normalized MSE vs. SNR per pulse for second-order DLL

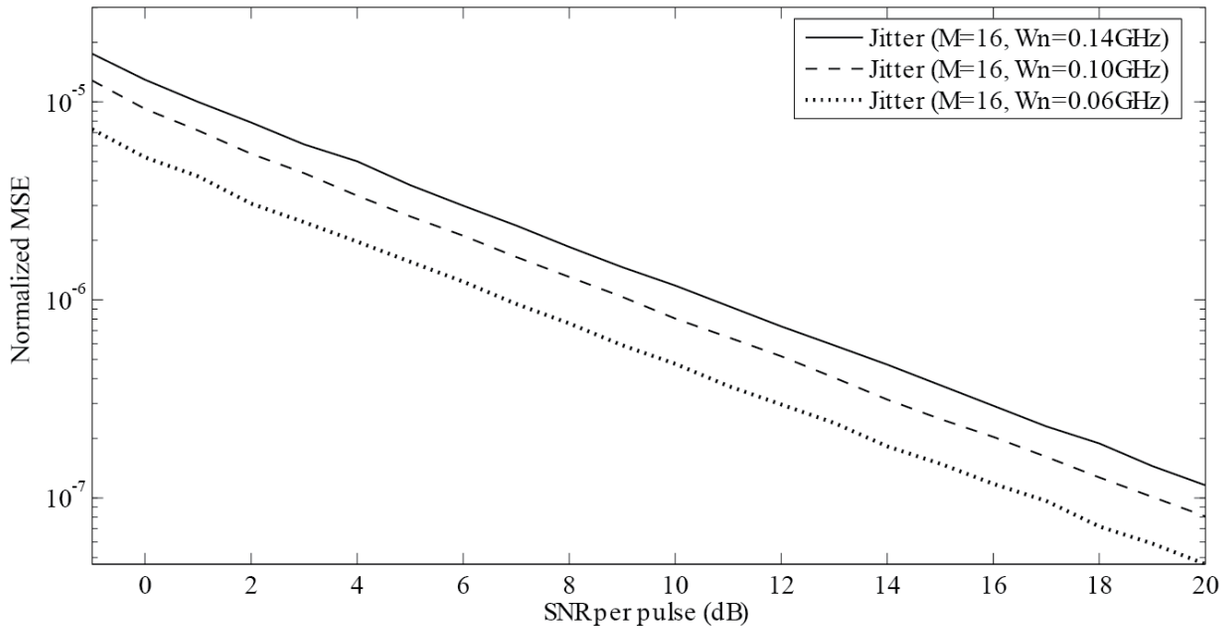

Figure 21. Normalized MSE vs. SNR per pulse for second-order DLL with data-aided bit number $M = 16$

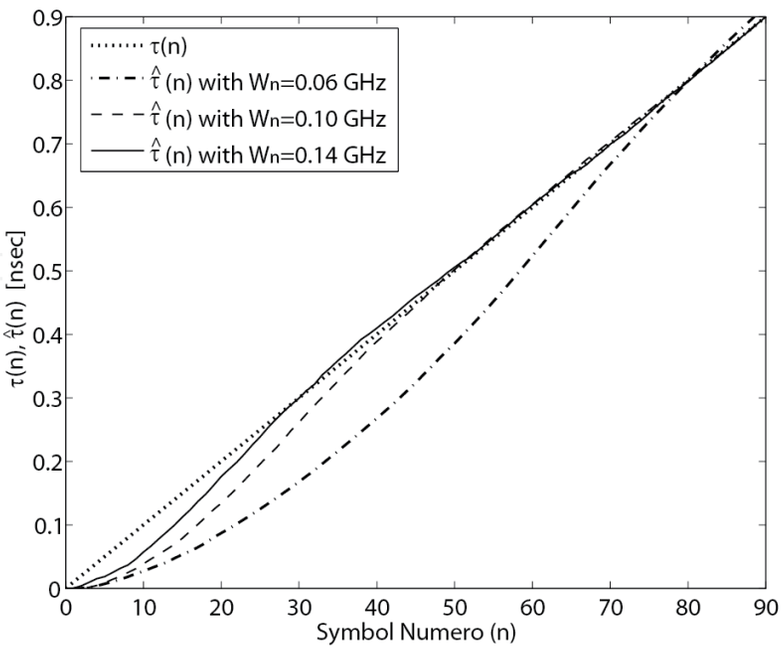

Figure 22. DLL transient reponse for ramp input (SNR=3 dB)

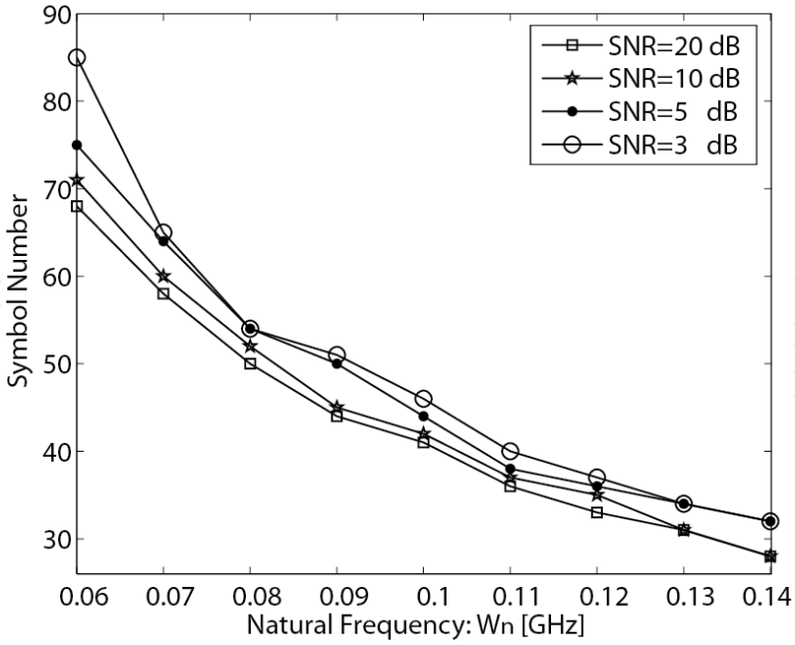

Figure 23. DLL transient performance (symbol number vs. natural frequency)

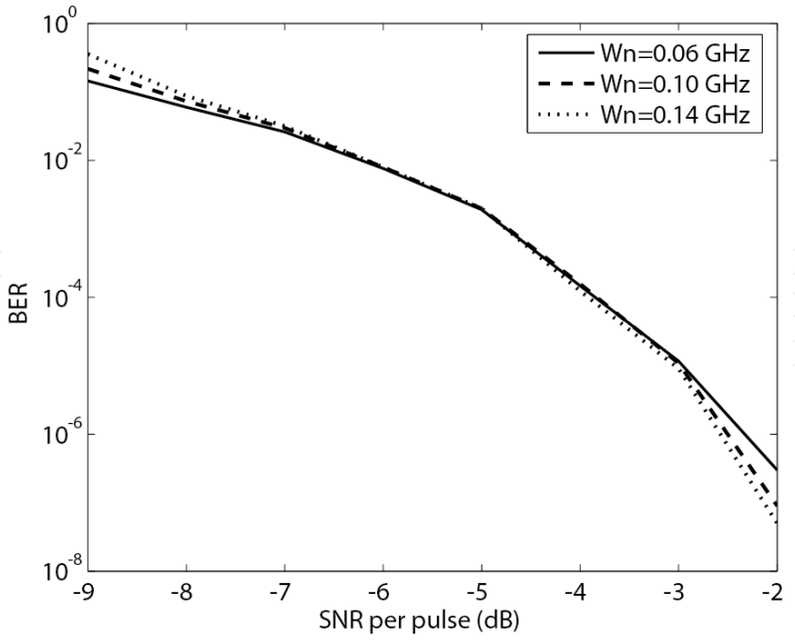

Figure 24. BER for the proposed DLL ($M = 16,\ \tau(n) = 0.01nT_p$)

Figure 23, which shows how many symbols (iteration steps) DLL requires to match efficiently $\tau(n)$. It is clear that when ω_n increases, the required number of symbols would decrease and transient response would be better. The BER comparisons for various ω_n are depicted in Figure 24. We observe that for high SNR values, an increase in ω_n (Simultaneously, B_L augments) leads to speed up the tracking operation and to improve BER performance. On the other hand, for small SNR values, the bandwidth B_L should be decreased for alleviating the noise effects, but that degrades the transient performance. Consequently, increasing ω_n helps to reduce the transient error effect, but in the price of noise handling ability.

In this section, we design a second-order DLL used for tracking timing offset variations in the received waveform, taking Doppler effects into account. We combine DLL with TDT, which enables enhancing the received energy capture even when the multipath channel and the TH codes are both unknown. Consequently, the proposed approach contributes to improve tracking performance for UWB systems and to reduce receiver structure complexity. For selecting the optimum parameter values for the proposed DLL, we apply Wiener-filter theory. Simulation results show the performance of the proposed DLL and conform that increasing ω_n helps to reduce the transient error effect, but in the price of noise handling ability.

6. Conclusion

In this chapter, we use TDT algorithms for carrying out low-complexity high-performance timing synchronization, which constitutes a major challenge in realizing the UWB communications. TDT technique is based on correlating the received signal with "dirty template" extracted from the received waveforms. This template is called dirty; because it is distorted by the unknown channel and by the ambient noise. TDT allows enhancing received energy capture and reducing receiver structure complexity. The described technique can be applied to UWB systems even in the presence of time hopping TH codes, Inter-Frame Interference (IFI) and rich multipath environment, where Inter-Symbol Interference (ISI) is absent.

TDT synchronization system consists of three main blocks: signal detection, timing acquisition and tracking. Each block of them is explained in a separated section of this chapter. In signal detection section: the dirty template detector is derived by applying the Neyman-Pearson theory. Then, Monte Carlo simulations are performed to find the probabilities of false alarm (P_{FA}) and detection (P_D). The results of simulation analysis show that the detection performance of the dirty template approach is improved by increasing the signal-to-noise ratio (SNR) or data-aided (DA) number. After detecting the UWB signal, the symbol-level timing offset estimation relies on searching the best statistics T.

Next section presents the timing acquisition in both DA and NDA modes. Then, we improve the timing estimator by adding a suitable window filter to the structure of the cross-correlation. Both the theoretical analysis and Matlab simulation results show the

performance of the proposed TDT estimator, and confirm that DA mode has the high performance and fast timing, compared to NDA mode, but that is at the price of bandwidth efficiency.

In tracking section, we demonstrate tracking system design taking into consideration the relative motion between Transmitter and receiver. We combine DLL with TDT, which enables to enhance the received energy capture and to improve tracking performance. Simulation results illustrated that increasing natural frequency parameter ω_n helps to reduce transient error effect, but in the price of noise handling ability.

Author details

Rshdee Alhakim and Emmanuel Simeu
TIMA Laboratory, Grenoble University, Grenoble, France

Kosai Raoof
LAUM Laboratory, Maine University, Le Mans, France

7. References

[1] Pagani P, Talom F, Pajusco P, Uguen B (2007) Communications Ultra Large Bande: le canal de propagation radioélectrique. Hermes. 246 p.

[2] Tian Z, Giannakis G.B (2006) Timing Synchronization for UWB Impulse Radios. In: Shen X, Guizani M, Qiu R.C, Ngoc T.L, editors. UWB Wireless Commun. and Networks. John Wiley & Sons. pp. 53-81.

[3] Win M.Z, Scholtz R.A (1998) On the energy capture of ultrawide bandwidth signals in dense multipath environments. IEEE Commun. Letters. pp. 245-247.

[4] Homier E.A, Scholtz R.A (2002) Rapid acquisition of Ultra-Wideband signals in the dense multipath channel. Proc. IEEE Conf. Ultra Wideband Systems and Techno. pp. 105-110.

[5] Fleming R, Kushner C, Roberts G, Nandiwada U (2002) Rapid acquisition for Ultra-Wideband localizers. Proc. IEEE Conf. Ultra Wideband Systems and Techno. pp. 245-250.

[6] Yang L, Tian Z, Giannakis G.B (2003) Non-data aided timing acquisition of Ultra-Wideband transmissions using cyclostationarity. Proc. IEEE Conf. ASSP. pp. 121-124.

[7] Strohmer T, Emami M, Hansen J, Papanicolaou G, Paulraj A.J (2004) Application of time-reversal with MMSE equalizer to UWB communications. IEEE Global Telecommun. Conf. pp. 3123-3127.

[8] Zhang H, Goeckel D.L (2003) Generalized Transmitted-Reference UWB Systems. Proc. IEEE Conf. Ultra Wideband Systems and Techno. pp. 147-151.

[9] Pardinas-Mir J.A, Muller M, Lamberti R, Gimenes C (2011) TR-UWB detection and synchronization - Using the Time Delayed Sampling and Correlation detection method. Proc. 8th EuRAD Conf. pp.202-205.

[10] Alhakim R, Raoof K, Simeu E (2010) A Novel Fine Synchronization Method for Dirty Template UWB Timing Acquisition. Proc. 6th IEEE Conf. Wireless Commun. Networking and Mobile Computing. pp. 1-4.

[11] Alhakim R, Raoof K, Simeu E, Serrestou Y (2011) Cramer-Rao Lower Bounds and Maximum Likelihood Timing Synchronization for Dirty Template UWB Communications. Signal Image and Video Processing J. pp. 1-17.

[12] Alhakim R, Raoof K, Simeu E, Serrestou Y (2011) Data-aided timing estimation in UWB communication systems using dirty templates. IEEE Int. Conf. on Ultra Wideband. pp. 435-439.

[13] Alhakim R, Simeu E, Raoof K (2011) Internal model control for a self-tuning Delay-Locked Loop in UWB communication systems, Proc. 17th IEEE Conf. pp. 121-126.

[14] Yang L, Giannakis G.B (2005) Timing Ultra-Wideband Signals with Dirty Templates. IEEE Trans. On Commun. pp. 1952–1963.

[15] Yang L, Giannakis G.B (2003) Low-complexity training for rapid timing acquisition in ultra-wideband communications. Proc. Global Telecommun. Conf. pp. 769–773.

[16] Kay S.M (1998) Fundamentals of Statistical Signal Processing: Detection Theory. Prentice Hall. 672 p.

[17] Meyr H, Ascheid G (1990) Synchronization in Digital Communications: Phase-, Frequency-Locked Loops, and Amplitude Control. John Wiley & Sons. pp. 21-162.

[18] Meyr H, Moeneclaey M, Fechtel S.A (1997) Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing. John Wiley & Sons. pp 61.206

[19] , FCC First Report and Order: In the matter of Revision of Part 15 of the Commission's Rules Regarding Ultra-Wideband Transmission Systems. FCC 02-48, Apr. 2002.

[20] Saleh A.A.M, Valenzuela R.A (1987) A statistical model for indoor multipath propagation. Selected Areas in Commun. j. pp. 128-137.

[21] Lee J.S, Miller L.E (1998) CDMA Systems Engineering Handbook. Artech House Publishers. pp. 677-834.

[22] Salem F, Pyndiah R, Bouallegue A (2005) Synchronization Using an Adaptive Early-Late Algorithm for IR-TH-UWB Transmission in Multipath Scenarios. 2nd Int. Symp. on Wireless Comm. Sys. pp. 268-271.

[23] Zhang W, Shen H, Kwak K.S (2007) Improved Delay-Locked Loop in a UWB Impulse Radio Time-Hopping Spread-Spectrum System. ETRI j. Available: http://dx.doi.org/10.4218/etrij.07.0506.0032.

[24] Farahmand S, Xiliang L, Giannakis G.B (2005) Demodulation and Tracking with Dirty Templates for UWB Impulse Radio: Algorithms and Performance. IEEE Trans. On Vehicular Technology. pp. 1595 – 1608.

[25] Jaffe R, Rechtin E (1955) Design and performance of phase-lock circuits capable of near-optimum performance over a wide range of input signal and noise levels. IRE Trans. Inf. Theory. pp. 66–76.
[26] Rappaport T.S (2001) Wireless Communications: Principles and Practice 2nd ed. Prentice Hall. pp. 139-196.

IntechOpen