Acquisition of Wideband Direct-Sequence Spread Spectrum Signals In System C

Zhila Amini Sheshdeh
University of Tarbiat Modares
Zh.Amini@gmail.com

Abdolreza Nabavi
University of Tarbiat Modares
abdoln@modares.ac.ir

Mohammad Bagher Ghaznavi Ghoushchi
University of Tarbiat Modares
ghoushchi@modares.ac.ir

Abstract: *Wideband (UWB) radio is gaining increasing attention thanks to its attractive features that include low-power low-complexity base band operation and ample multipath diversity. In another side using system C for implementing systems on a chip has a lot of advantages.*
The goal of this paper is to implement a proper algorithm for DS_UWB timing acquisition in system C.
In this paper we implement an algorithm for a direct sequence ultra wideband receiver with single user and we use IEEE standard channel model, therefore interference such as MUI and ISI are present. In this design data rate, chip rate and sampling rate are about 110Mb/s, 660Mchip/s and 4GS/s that are proper for DS_UWB and also the achieved BER is acceptable.

Keywords: Acquisition, DS_UWB, system C.

1 INTRODUCTION

A class of spread spectrum techniques known as ultra wideband (UWB) communication has recently received a significant amount of attention from academic researchers as well as from the industry.
UWB signaling is being considered for high data rate wireless multimedia applications for the home entertainment and personal computer industry, as well as for low data rate sensor networks involving low power devices. It is also considered a potential candidate for alternate physical layer protocols for the high-rate IEEE 802.15.3 and the low-rate IEEE 802.15.4 wireless personal area network (WPAN) standards [3].

In any communication system, the receiver needs to know the timing information of the received signal to accomplish demodulation.
The subsystem of the receiver which performs the task of estimating this timing information is known as the synchronization stage. Synchronization is an especially difficult task in spread spectrum systems which employ spreading codes to distribute the transmitted signal energy over a wide bandwidth.
The receiver needs to be precisely synchronized to the spreading code to be able to despread the received signal and proceed with demodulation. In spread spectrum systems, synchronization is typically performed in two stages.
The first stage achieves coarse synchronization to within a reasonable amount of accuracy in a short time, and is known as the acquisition stage.
The second stage is known as the tracking stage and is responsible for achieving fine synchronization and maintaining synchronization through clock drifts occurring in the transmitter and the receiver.
Tracking is typically accomplished using a delay locked loop. Timing acquisition is a particularly acute problem faced by UWB systems [3], and here we discuss a coarse synchronization algorithm.
This algorithm develops low-complexity training (data-aided) schemes for rapid timing acquisition in UWB. The novel approach relies on a special cross-correlation pattern among received waveforms, which is enabled by our training sequence design.
Similar to pilot waveform assisted modulation, and the so-termed transmitted reference approaches to

channel estimation, our timing acquisition scheme exploits the rich multipath diversity provided by UWB channels. The resulting algorithm entails simple integrate-and-dump operations per symbol.

This reduces complexity and markedly improves acquisition speed by one or two orders of magnitude [1].

In another side nowadays because of valuable advantages of SOC technology, designers try to implement their design in a way to use this new technology. For this purpose in this paper we use system C to implement DS_UWB receiver synchronizer that has an important role in this receiver. Design in system C prepares the opportunity for using SOC technology.

The paper is organized as follows. In section II, the proposed algorithm is described. Section III describes implementation in system C and simulation results are reported in section IV. Finally, section V provides the concluding remarks.

2 BLIND TIMING ACQUISITION

In UWB radio, every information symbol is conveyed by N_f data modulated ultra short pulses p(t), each over one frame of duration T_f .The resultant symbol duration is thus $T_s=N_fT_f$ seconds. With p(t) having duration T_p($<<T_f$) at the sub-nanosecond scale, the transmitted signal occupies UWB with bandwidth $B_s\sim1/T_p$ [1].UWB radio generally adopts modulation methods such as pulse position modulation (PPM), and pulse amplitude modulation (PAM) and binary phase shift keying (BPSK). In this paper, we will deal with BPSK.

When multiple users are present the transmitted UWB signal which consists of a train of short pulses (monocycles) may be dithered by a time hopping (TH) sequence to facilitate multiple access and to reduce spectral lines or the polarities of the transmitted pulses may also be randomized using a direct sequence (DS) spreading code to mitigate multiple access interference (MAI) [1].

The UWB signal transmitted during the acquisition process for single user in duration of T_s in DS_UWB can be expressed as [3]:

$$P_T(t)=\sum_{n_f=0}^{N_f-1}\alpha_{[n_f/N_{ds}]}p(t-n_fT_f) \qquad (1)$$

That $\alpha_{[n_f/N_{ds}]}$ is the randomized coefficient that is different (+1 or -1) for each frame.

In Fig.1 you can see a signal transmitted with DS_UWB modulation.

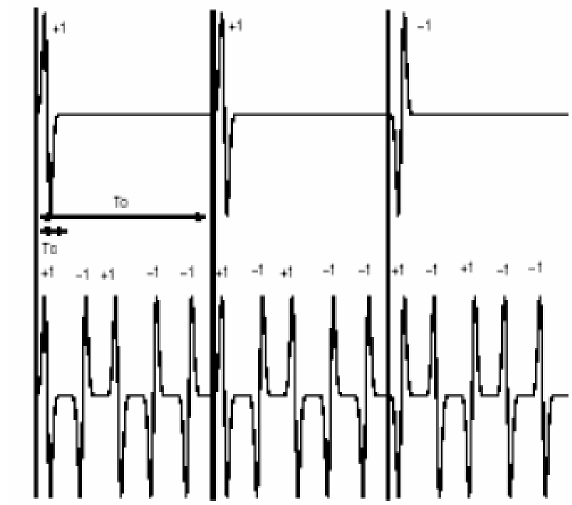

Fig 1. DS_UWB modulation in UWB systems.

The multipath channel is modeled as a tapped-delay line, with L +1 taps $\{\alpha_l\}_{l=0}^{L}$ and delays $\{\tau_l\}_{l=0}^{L}$ satisfying $\tau_l<\tau_{l+1}$. Being quasi-static, the channel coefficients and delays remain invariant over one transmission burst, but are allowed to change across bursts.

To isolate the multipath spreading effects from the propagation delay τ_0, all path delays can be uniquely casted into $\tau_{l,0}:=\tau_l-\tau_0$.

Focusing on a single user link, and treating multiuser interference (MUI) as noise, the waveform arriving at the receiver is given by:

$$r(t)=\sqrt{\varepsilon}\sum_{l=0}^{L}\alpha_l\sum_{k=0}^{+\infty}s(k)p_T(t-kT_s-\tau_{l,0}-t_2)+w(t),$$

Where the noise term w (t) includes the MUI, and the first arrival time t_2 is nothing but the transmission starting time t_1 augmented by the propagation delay τ_0 .

The first step of our blind timing acquisition algorithm is to take from the received waveform a segment of duration T_s, starting at time (t_3+ nT_f + mT_s), for integers n$\in$ [0, N_f), and m $\in$[0,M-1], with MT_s being the observation interval.

Denoted by $x_{n,m}(t)$, this waveform can be expressed as:

$$x_{n,m}(t)=r(t+T_s+nT_f+t_3),t\in[0,T_s) \qquad (2)$$

The proposed r (t) is received signal contain of several P_T (t) with different coefficient.

The blind algorithm base on dirty template can be carried out as follow:

Step 0. Set n=0.

Step 1. For a given n, take M segment $x_{n,m}(t)$ each of duration Ts from the received signal as in (2). Integrate-and-dump the product of adjacent segments $x_{n,m}(t)$ and $x_{n,m+1}(t)$ as below:

$$R_{xx}(n;m) := \int_0^{T_s} x_{n,m}(t) x_{n,m+1}(t) dt \qquad (3)$$

Step 2. Form an estimate of Rxx(n) by averaging over all pairs the absolute value of the integral obtained in Step 1 in this way:

$$\hat{R}_{xx}(n) = \frac{2}{M} \sum_{m=0}^{M/2-1} \left| \int_0^{T_s} x_{n,2m}(t) x_{n,2m+1}(t) dt \right| \qquad (4)$$

If $n < N_f - 1$, set $n = n + 1$, and go to Step 1. Otherwise, go to Step 3.

Step 3. By peak-picking $\hat{R}_{xx}(n)$, we find an estimate for synchronization time. If $\hat{n}_{\in}$ = arg $\max_n\{\hat{R}_{xx}(n)\}$ then $\hat{n}_{\in} T_f$ is an estimate of a symbol starting time that with applying this time shift, receiver will get synchronized.

It is necessary to mention that in this paper we are not supposed to explain the proposed algorithm completely because it need another paper for complete explanation .for more information you cane refer to [1].

In [1] the algorithm has been used for a single user TH_UWB system with multipath channel and it is also in simulation level not hardware implementation. But in this paper we developed this algorithm for a DS_UWB system with single user and IEEE channel in system C.

3 Implement In System C

In order to implement this algorithm in system C, we have designed four blocks: acquisition, driver, reset generator and monitor.

The duty of driver is to generate the two important needed signals named as input and template signal.

Input signal is data that has passed from channel but template signal is that which is before channel and we use them for correlating with received signal for finding BER. These signals are generated in each positive edge of clock.

```
SC_CTOR (driver)
      {
        SC_METHOD (prc_driver);
        sensitive_pos<< clk;
      }
```

Because there is no model for IEEE channel in system C and we have only its model in simulink, we send the transmitted data to IEEE channel in simulink and then use the data after IEEE channel in our system C code.

For generating reset signal we have used a SC_THREAD process in reset generator block. In the code this process has no sensitivity list and it is done only one time.

```
SC_CTOR (rst_gen)
{
SC_THREAD (prc_wave);
}
```

Through monitor block the needed output is shown and in the acquisition block that is the main one, the performance start with reset signal.

Each received frame is made of three parts. First part is “preamble” that is for synchronization. It means that this part contains no information and it is used for synchronization between transmitter and receiver.

Because of the time that is consumed by synchronization algorithm we should define “before data” part that is the second one, and therefore when the synchronization algorithm is performed in another process, the block is receiving the “before data”.

The number of bytes in “before data” is determined base on the time estimation that the synchronization algorithm is going to take. It means that before the “before data” signals become finished, the synchronization algorithm had finished before.

Therefore after reset, the acquisition block goes to the state for saving the preamble part of the received frame.

```
case save_preamble:

      cout<<"preamble ,sample "<<g<< endl;
      if (g<(preamble_samplenum-1))
      {
        preamble[g]=input;
      }
```

After this state it starts to save the “before data” part.

```
case wait_for_data:

cout<<"bdata,sample "<<h<<endl;
if (h<(bdata_samplenum-1))
{
        bdata[h]=input;
```

```
        h=h+1;
}
```

Meanwhile in this time synchronization algorithm performs in another process in this block.

```
void acquisition::sync_algoritm()
{
……
}
```

After this state the blocks save the third part of frame that is "main data" part. This part is t the real data that has information.

```
case save_data:
cout<<"maindata,sample "<<p<<endl;
        data[p]=input;
```

In the middle of saving "main data", acquisition block correlates the samples in each received T_s with related samples in shifted template signal to calculate the BER.
The calculated BER determines the level of reliability of an algorithm.

```
for (t=0;t<bit_samplenum;t=t+1)
        {
        op1[t]=templat[((k-1)*bit_samplenum)+t];
        op2[t]=data[((k-1)*bit_samplenum)+t];
        }

        convol=correlation(op1,op2);
        if (convol<0)
          {
                error=error+1;
          }
```

If the result of correlation is negative, it means that the shape of data has changed after channel, and the sign of most samples in a bit has changed and it will cause error in determining the value of that bit.

4 SIMULATION RESULTS

The simulation result were accomplished with Microsoft Visual C++6.0.
For simulating the proposed design there is a .cpp file that assigns sub blocks ports to each other properly. This file contains SC_MAIN as below.

```
int sc_main(int argc, char * argv[])
{
rst_gen b4 ("reset");
        b4<<reset;
driver b2 ("Geninput");
        b2<<clock<<reset<<i2<<i1;
…
}
```

After compiling and linking the .exe file will be made and the result is shown.
Table 1 summarizes the performance with data rate about 111Mbit/sec.

TABLE I
PERFORMANCE ANALYSIS

Parameter	Value
N_f	6
T_s	9 nsec
T_f	1.5 nsec
T_p	1 nsec
Number of "preamble"	100
Number of "before data"	70
Number of "main_data"	900
Sampling rate	4GHz
BER	0.012

5 CONCLUSION

Implementation of a blind timing algorithm for UWB radios based on integrate-and-dump operations between adjacent symbol-long segments of the received waveform in system C has been presented in this paper.
Segments of such pair serve as "dirty templates" for each other.
The resultant timing algorithm exploits the ample multipath diversity inherent to UWB transmissions, without knowledge of either information symbols or the channel.
Simulations and comparisons confirm considerable improvement in both error performance and acquisition speed, over existing blind algorithms [1].
It is worth mentioning that this algorithm is readily applicable to non-UWB systems, when inter-symbol interference is absent [1].
Also in this paper we use the proposed algorithm for DS_UWB in somehow nearest condition to reality such as IEEE channel, reasonable data rate and sampling rate.
Another important point that should be mentioned is nowadays that using SOC has developed; design of some necessary block in the form of reusable code is so valuable and helpful.

REFERENCES

[1] Liuqing Yang and Georgios B. Giannakis, "Blind UWB Timing With a Dirty Template,"*IEEE, ICASSP*, 2004.

[2] Liuqing Yang and Georgios B. Giannakis, "Low-Complexity Training for Rapid Timing Acquisition in Ultra Wideband Communications," *IEEE, GLOBECOM*, 2003.

[3] Sandeep R. Aedudodla, Saravanan Vijayakumaran, and Tan F. Wong, "Timing Acquisition in Ultra-wideband Communication Systems," *IEEE Transaction on Vehicular Technology*, Vol. 54, No. 5, September 2005.