A Platform for Delay Hopped Transmitted Reference UWB Communication System Prototype Development

Yiyin Wang, *Student Member, IEEE*,
Andreas Schranzhofer, Rene van Leuken and Alle-Jan van der Veen, *Fellow, IEEE*

***Abstract*—Delay-Hopped Transmitted-Reference(DHTR) is an attractive scheme for Ultra-wideband(UWB) systems. It relieves the channel estimation and the synchronization problems, which simplifies the UWB system. This paper presents a platform developed at Delft University of Technology. The purpose of the platform is to supply a hardware environment to implement DHTR UWB components. It can emulate real UWB signals as inputs of our digital UWB receiver and make efficient use of a USB link between the hardware and the PC. This platform provides a flexible environment for exploring and testing the realization of novel digital receiver algorithms. Currently it is being used to implement acquisition and demodulation algorithms. The obtained results are promising.**

***Index Terms*—Ultra Wideband(UWB), Delay-Hopped Transmitted-Reference(DHTR), Platform, ADC, USB**

I. Introduction

ULTRA wideband(UWB) technology can provide high speed, low cost and low complexity wireless communications with the capability to overlay on existing frequency allocations. Delay-Hopped Transmitted-Reference(DHTR) [1] is a practical scheme for UWB communications. It transmits pulse pairs composed by data pulses and reference pulses. The reference pulses can be used as correlation templates, since they experience the same channel distortion as the data pulses. It elegantly avoids complicated channel estimation and doesn't suffer from the stringent pulse level synchronization requirements as traditional RAKE transceivers. The analog parts can run independently without any feedback control of the digital parts. These advantages make it less complex to implement the DHTR transceivers than the traditional RAKE transceivers.

In order to explore the hardware architecture of the digital receiver algorithm for the DHTR scheme, we developed a platform (Fig. 1). In the platform, an Arbitrary Waveform Generator(AWG) is programmed to generate UWB signals. These represent the transmitted DHTR UWB signals, convolved with a measured multipath channel impulse response and correlated by themselves with certain delay. A dual-channel, 12-bit 125MSPS A/D Convertor(ADC) samples the signals at 100 MHz. The data samples are transferred through a USB link from the FPGA board to the PC and demodulated by Matlab programs. The function block diagram of the platform is depicted in Fig. 2. It indicates the case, in which the signal processing is executed in the PC and the FPGA board is employed as a pipe.

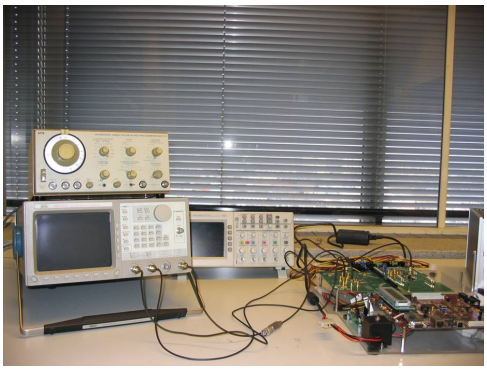

Fig. 1. The platform set up

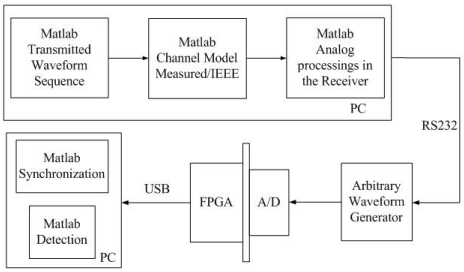

Fig. 2. The block diagram of the platform for a DHTR UWB system prototype development

Since we are using an FPGA, it is possible to programme it with changing functionality. In a first experiment we programmed the FPGA with a two-stage acquisition and demodulation algorithm. The algorithm takes advantage of the statistical properties of the UWB signal as well as of training sequences(header) in the data packet. In the first stage a chip level synchronization is achieved, and in the second stage a header is searched to find the start of a data packet. The two stage strategy is a tradeoff between hardware resource usage and acquisition time.

The paper is organized in the following way. Detailed description of the platform is presented in Section II. The two-stage acquisition and demodulation algorithm is explained in

Y. Wang, R. van Leuken and A.J. van der Veen are with Circuit & System Group, in Department of Electrical Engineering, Delft University of Technology, Delft, The Netherlands e-mail: Yiyin.Wang@tudelft.nl.

A. Schranzhofer is with Signal Processing and Speech Communication Laboratory, Department of Electrical Engineering, Graz University of Technology, Graz, Austria email: schranzhofer@student.tugraz.at.

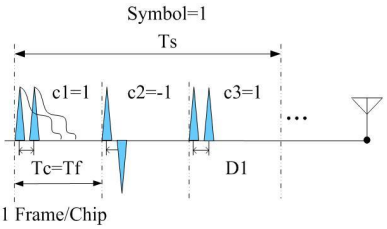

Fig. 3. The transmitted UWB signal

Section III. Some experimental results are shown in Section IV. The conclusions are drawn in Section V.

II. Platform

A. UWB Signal Generation

The DHTR scheme transmits pulse pairs, for example as shown in Fig. 3. The pairs of pulses are called doublets. The first pulse in the doublet is the reference pulse. The second one is the data pulse. Each frame contains one doublet. Several frames compose one chip and several chips construct a symbol $s_i \in \{-1, +1\}$. Every chip has a chip code $c_j \in \{-1, +1\}$. The polarity of the data pulse is modulated by the product of the chip code and the symbol. There is a certain delay interval between the reference pulse and the data pulse. The delay can be different for every chip. The chip code and the delay interval are known by both transmitter and receiver. They facilitate the DHTR system to accommodate multiple users.

We use Matlab to generate signals at a sampling frequency corresponding to 20 GHz. The pulse shape of the signal is a second derivative Gaussian pulse of 0.5 ns width. The frame time T_f is 50 ns. One frame is one chip. The delay interval D is 6 ns for all chips to avoid inter-pulse interference and simplify the system [2]. Each symbol is made up of $N_f = 8$ chips, resulting a symbol time $T_s = N_f \cdot T_f = 400$ ns. The chip codes are randomly generated. All these parameters can be modified easily in Matlab. The platform provides the flexibility to change the experimental parameters.

A transmitted signal propagates through a channel [3]. The channel does not only include the psychical UWB channel but also the antenna's impulse response. The length of the channel T_h is truncated to 90 ns. Most of the channel energy concentrates on the first 20-30 ns from the measurements. Even if $T_h > T_f$, the inter-frame interference can be ignored. Fig. 4 shows an example of a signal after traveling through the channel. The received signal without any noise is generalized in (1):

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

where s_i is the transmitted i_{th} symbol; $c_{j,j=1,\ldots,8}$ is the chip code repeated for every symbol. $h(t)$ is the aggregate UWB channel and can be represented by $h(t) = h_p(t) \star g(t) \star a(t)$, where $h_p(t)$ is the UWB physical channel of length T_h, $g(t)$ is the pulse shape filter and $a(t)$ stands for the antenna response. D represents the previously mentioned delay.

The received signals are correlated with delayed versions of themselves using the predefined delays at the transmitter. The

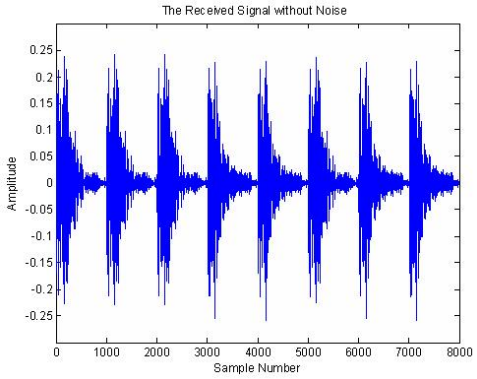

Fig. 4. The UWB channel and the received signal

usage of multiple delays requires multiple correlation branches in the analog part of the receiver. The digital receiver samples the output of the correlation and processes the signal. The analog correlation is done by Matlab programs instead of real analog circuits. One analog correlation branch is indicated in Fig. 5. The received signal is multiplied with its delayed version. An integrate and dump(I&D) component is used to generate discrete time results. The integration time window with length 10ns is the inverse of the sampling frequency(100 MHz) of the ADC. The I&D and the ADC are synchronized by the same clock as shown in the upper part of Fig. 5. Since we have an ADC and virtual I&D(Matlab), the synchronized clock can't be applied. The solution is to construct an ideal sliding window integrator in Matlab to generate continuous output. When sampling this output at 100 MHz, the equivalent I&D results can be achieved. The output of the sliding window integrator is shown in (2).

$$\begin{aligned} x(t) &= \int_t^{t+T_w} r(\tau) r(\tau - D)\, d\tau \\ &= \int_t^{t+T_w} [h(\tau) + s_i c_j h(\tau - D)] \\ &\quad [h(\tau - D) + s_i c_j h(\tau - D - D)]\, d\tau \\ &= \int_t^{t+T_w} [h(\tau)h(\tau - D) + h(\tau - D)h(\tau - 2D) \\ &\quad + s_i c_j h(\tau)h(\tau - 2D) \\ &\quad + s_i c_j h(\tau - D)h(\tau - D)]\, d\tau \end{aligned} \tag{2}$$

Since the delay interval D is chosen to be much larger than the correlation time of the channel, the cross correlation term in (2) can be ignored. The output of the sliding window integrator is simplified to:

$$x(t) = s_i c_j \int_t^{t+T_w} h(\tau - D)h(\tau - D)\, d\tau \tag{3}$$

The AWG is used to emulate the output signal of the sliding window integrator. According to the Nyquist sampling theory, the AWG can reconstruct the signal, if the data used to trigger the AWG is the samples resulting from sampling the signal beyond its Nyquist rate. The bandwidth of the sliding window output is smaller than 100 MHz. It is sampled at 200 MHz. The AWG can reconstruct the sliding window output exactly.

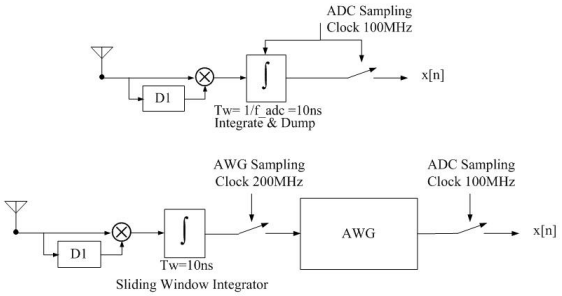

Fig. 5. The equivalent analog parts

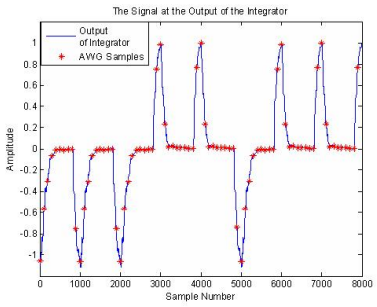

Fig. 6. The signal after the integrator

The continuous signal in Fig. 6 is the output of the sliding window integrator and the star samples are used to trigger the AWG.

The AWG has two channel outputs. Channel 1 is used to output the UWB signal and Channel 2 is used to output the 100 MHz clock for the ADC. Since both the UWB signal and the clock are generated by the AWG, they are coherently synchronized. To break up this synchronization effect, another clock source can be used to trigger the ADC to make it sample freely. An other way is to randomly start to sample the sliding window integrator in Matlab for every experiment.

B. Data Sampling

An AD10226 evaluation board [4] is utilized in the platform. It has a dual-channel, 12-bit 125 MSPS IF sampling A/D convertor module. Two AD9433 chips are integrated on the module. The channels can sample separately, and can also operate in an interleaving mode to get higher sampling frequency or reduce the workload for each channel. The ADC is connected to an FPGA board. The output of the ADC is used as input to the FPGA, hence the FPGA should only read data when the ADC's output is stable.

C. USB Link

The Spartan-3 Development Kit from Avnet Design Service [6] facilitates the design with Xilinx Spartan-3 FPGA. This board integrates additional hardware to help the user to implement complete applications. There are three types of interface to communicate with the outside: RS232, 10/100 Ethernet and USB 2.0. A Cypress EZ-USB FX2 [7] chip is included on the board for USB 2.0 communication. This device integrates a

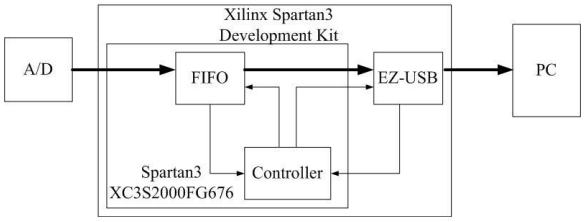

Fig. 7. Function block diagram for USB communication

USB 2.0 transceiver, a Serial Interface Engine(SIE) and a 8051 micro-controller. It can provide a data throughput as high as 480Mbps . Open software resource [5] is available to configure the EZ-USB FX2.

The EZ-USB chip can only be driven at any frequency between 5MHz and 48MHz. The ADC samples at 100MHz. Therefore it is impossible to transfer the ADC outputs directly to the PC. A FIFO is designed as a data buffer to be inserted between the ADC outputs and the EZ-USB chip. Fig. 7 shows the function block diagram for the USB communication. The controller first lets the ADC outputs write into the FIFO until the FIFO is full. In the second step it makes the EZ-USB chip read out the data from the FIFO until the FIFO is empty and send them to the PC. The length of the data block is determined by the length of the FIFO. If a digital receiver is implemented on the FPGA, it takes the place of the FIFO to prepare data for the USB link. The USB link provides a continuous transfer rate of 35Mb/s. Now we can easily verify our receiver algorithms in Matlab using real signal samples. The algorithms can be evaluated more thoroughly.

III. Digital Receiver Implementation

A first receiver prototype has been designed, implemented and realized. The function of this digital receiver is to acquire the start of the data packet at the chip level and demodulate the signal symbols. To indicate the beginning of each consecutive data packet, a header sequence consisting of $N_h = 8$ symbols is used, each of which are made up of $N_f = 8$ chips as described earlier. There are 64 candidate positions for the first chip of the first training symbol. If we check all the possibilities concurrently, insufficient resources are available. On the other hand, if we check sequentially, the synchronization time might be too long. We have developed a hybrid solution[8]. The acquisition procedure is scheduled into two steps. The first step is to match the chips. The second step is to identify the header.

As we mentioned before, the acquisition procedure is at the chip level. The ADC samples at 100MHz(1 sample/10ns). Since the chip length is defined as 50ns, we get 5 samples out of each chip. We assume sample level synchronization is already achieved. The five samples, which belong to one chip, are added together to get one combined sample $x_i[j]$ to represent the jth chip of the ith symbol. It is identical to extend the integration time to T_f(50ns). The $x_i[j]$ can be presented as:

$$\begin{aligned} x_i[j] &= s_i c_j \int_{(i-1)*T_s+(j-1)*T_f}^{(i-1)*T_s+j*T_f} h(\tau - D)h(\tau - D)\, d\tau \\ x_i[j] &= s_i c_j A \end{aligned} \quad (4)$$

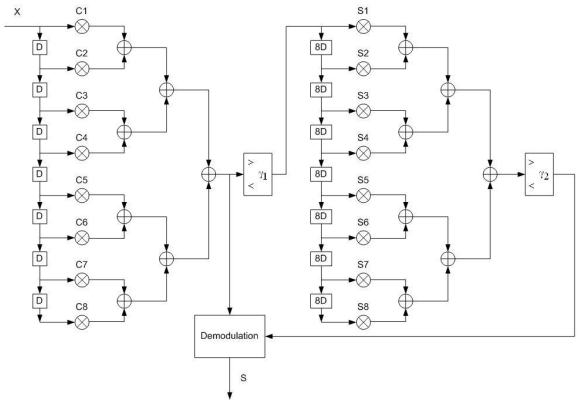

Fig. 8. The architecture for two stage synchronization digital receiver

where $A > 0$ is the aggregate channel energy and remains the same for every chip.

Both of the two steps follow the maximum likelihood principle. The first step is done by correlating the input signal $x_i[j]$ by the chip code c_j sequence with $N_f = 8$ different offsets. The correlations are executed concurrently. The maximum result is compared with a preset threshold γ_1 to distinguish the signal and the noise. The demodulated symbol values $s_d[i]$ are the signs of the correlation results.

$$s_d[i] = \text{sign}(\sum_{j=1}^{8} x_i[j]c_j) \tag{5}$$

If the chip code matches, the second step is carried out to identify the header of the data packet. The demodulated symbols are correlated by the training symbols with $N_h = 8$ different offsets concurrently. The maximum result compares with another preset threshold γ_2 to distinguish between signal or noise one more time. Once the value exceeds γ_2, the acquisition is accomplished. The output of the digital receiver are the demodulated symbols. The results can be shown in the PC through the USB link. The architecture for this two stage acquisition digital receiver is shown in figure 8.

This two step strategy avoids exhaustively search in a wide range. It compresses 64 chip candidates to 8 chip candidates and 8 symbol candidates, whereas the latter are only computed conditionally, that is, when the first step results in a chip code match. The hardware resources are dramatically reduced compared to a parallel solution. In each step, all correlations are executed concurrently, which speeds up the acquisition procedure compared to the sequential solution. It is a good trade-off between the resource utility and the acquisition time.

IV. Results

Two experiments have been carried out on the platform. In the first experiment, the FPGA board is used as a pipe as shown in Fig. 7 to transfer the data samples of the ADC to the PC. Fig. 9 shows the ADC samples collected by the FIFO implemented on the FPGA board. It is similar as the waveform shown in Fig. 6. It proves the functionality of the platform. The second experiment is to implement the digital receiver described in Section III on the FPGA board. It uses

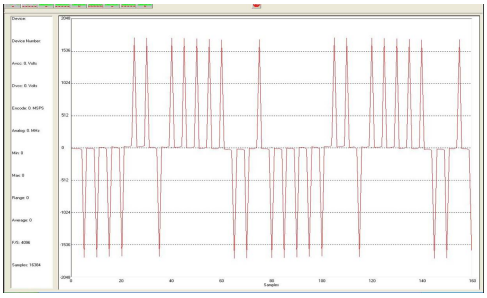

Fig. 9. The ADC samples of the UWB signal generated by AWG

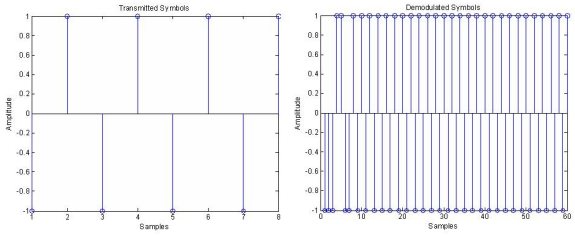

Fig. 10. The transmitted symbol and the received symbol

1575 LUT units of Xilinx Spartan3(xc3s2000FG676-5) FPGA chip and runs at a frequency of 50MHz. Fig. 10 shows the transmitted symbols $\{+1, -1, \ldots, +1, -1\}$ on the left side and the received demodulated symbols on the right side. The first few symbols are not correct, since the receiver is still in the acquisition procedure. Thereafter, the remaining symbols are correctly demodulated.

V. Conclusion

The resulting platform delivers a flexible experimental environment for the DHTR UWB communication system. The parameters of the UWB signal can be easily modified in a Matlab program. The novel digital receiver algorithms could be evaluated using real UWB signals. The acquisition and demodulation algorithms were implemented and verified in the platform.

Currently we are working on novel algorithms for synchronization and channel equalization.

References

[1] R.Hoctor and H.Tomlinson, "Delay-Hopped Transmitted-Reference RF communication", *IEEE conference on Ultra Wideband Systems and Technologies*, pp.265-269, May 2002

[2] Q.H.Dang and A.J. van der Veen, "A Decorrelating Multiuser Receiver for Transmit-Reference UWB Systems " *to appear, IEEE J. Sel. Topics in Signal Processing*, Volume 1, Issue 3, Oct. 2007.

[3] Z.Irahhauten, G.J.Janssen, H.Nikookar, A.Yarovoy, and L.P.Ligthart, "UWB channel measurements and results for office and industrial environments", *IEEE conference on Ultra Wideband Systems and Technologies*, pp.225-230, Sept. 2006

[4] Dual-Channel, 12 Bit 125 MSPS IF Sampling A/D Converter – AD 10226, 2002 ANALOG DEVICE, Inc.

[5] http://www.cip.physik.uni-muenchen.de/˜wwieser/elec/periph/USB-FX2/

[6] Xilinx Spartan -3 Development Kit User's Guide, 2004 Avent Design Service, AvnetAvenue Inc.

[7] EZ-USB Techinical Reference Manual, Cypress Semiconductor Inc.

[8] Andreas Schranzhofer, "Acquisition for a transmitted reference UWB receiver", Master's Thesis, TU Delft, May 2007