A New Hardware Design Scheme of Symbol Synchronization for an Underwater Acoustic Receiver

Li Ying , Benson Bridget ,Kastner Ryan , Chen Lan

Abstract— Conventional time domain symbol synchronizers typically use channel noise models to set static synchronization thresholds. However, static thresholds may prove insufficient in the underwater environment as the underwater acoustic channel can be highly variable and plagued by multipath, variable noise levels, path-loss, Doppler shifts, and long propagation delays. The paper provides a new hardware scheme using real time orthogonal sliding correlation as an adaptive threshold to build a symbol synchronizer suitable for an underwater FSK acoustic receiver. The design has been implemented and evaluated on an FPGA board. Both the simulation and real data experimental results show that the design provides accurate synchronization while consuming only 9% of the FPGA's slice resources.

I. INTRODUCTION

Symbol synchronization is a critical issue in the design of wireless communication systems. It is used to set accurate sampling and decision timing for subsequent demodulation. Conventional time domain symbol synchronizers typically use channel noise models to set static synchronization thresholds [1,2]. However, static thresholds may prove insufficient in the underwater environment as the underwater acoustic channel can be highly variable and plagued by multipath, variable noise levels, path-loss, Doppler shifts, and long propagation delays.

Although there is a wealth of literature on the theory of symbol synchronization[4-6], and some others describing its use in underwater communication systems[7-9], there is a lack of references describing how to approach designing a hardware symbol synchronization scheme for specific underwater applications. Therefore, the purpose of this paper is to describe and present experimental test results for a complete and innovative hardware symbol synchronizer for our underwater FSK acoustic receiver. Unlike conventional symbol synchronizers, the design sets a dynamic synchronization threshold based on a real-time orthogonal sliding correlation to effectively adapt to the underwater environment.

The rest of this paper is organized as follows: Section 2 describes our overall acoustic receiver design and its parameters that govern many design choices for symbol synchronization. Section 3 describes our symbol synchronization algorithm and presents software simulation results to prove its functionality. Section 4 describes the hardware implementation of our symbol synchronization scheme and presents the area, timing, and synchronization results of the design with both simulated and experimental data. We conclude in section 5.

II. UNDERWATER FSK RECEIVER DESIGN

Figure 1 illustrates a block diagram of our underwater acoustic receiver design. The input signal *adc_in* is the received analog signal, which consists of a modulated FSK waveform (when data is present) and noise. The following analog to digital converter (ADC) and Digital Down Converter (DDC) recover the signal to the digital 'baseband' based on the FSK modulation scheme. A synchronizer is then applied to locate the index of the beginning of a symbol. This index corresponds to the synchronization point allowing the demodulator to begin demodulation with accurate symbol timing. The demodulator follows the coherent FSK demodulation scheme described in [3] to decode the sequence. The decoded bit stream *data_out* is then sent to the host computer and translated to a readable message.

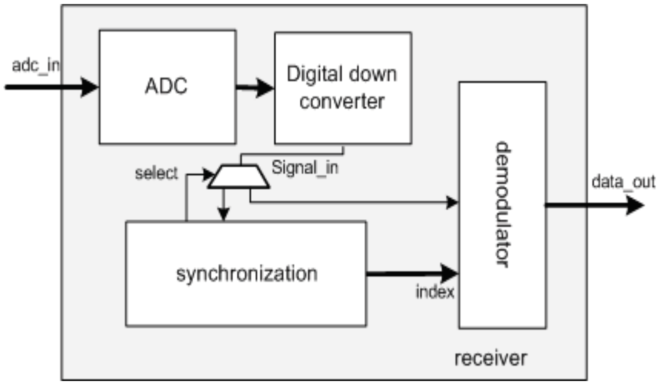

Figure 1. Block Diagram of Underwater Acoustic FSK Receiver

The receiver makes use of the frequency and rate parameters shown in Table 1. These parameters govern the timing constraints of the symbol synchronization scheme described in the following sections.

Table 1 Timing constraint and parameters for modem design [3]

properties	**assignment**
Modulation	FSK
Carrier frequency	40 kHz
Mark frequency	1 kHz
Space frequency	2 kHz
Symbol period	5 ms
Sampling Clock	800 kHz
Demodulator Clock[1]	16 kHz

III. SYMBOL SYNCHRONIZATION SCHEME FOR UNDERWATER ACOUSTIC MODEM

The symbol synchronization approach relies on the transmission of a predefined sequence of symbols, often referred to as a training, or reference sequence. The transmitter sends a packet that begins with the reference sequence and the receiver compares the received sequence with the known reference sequence in order to locate the start of the packet [4]. The comparison is effectively performed by sliding and correlating the received sequence with the reference sequence each time a new sample is received. When the reference and receiving sequence exactly align with each other, the correlation reaches a maximum peak and the synchronization is accomplished.

Conventional methods to detect a correlation peak involve setting a static correlation threshold in time domain where the correlation result only rises above the threshold when the receiving sequence and reference sequence are perfectly aligned [1,2]. The static threshold must be large enough to minimize the number of false alarms, but small enough to guarantee the peak will still be detected when the signal level is low. However, the underwater acoustic channel is highly variable and plagued by strong multipath, variable noise levels, path-loss, and long propagation delays [10,11]. Thus, selecting a static correlation threshold for the underwater receiver is impractical as the received signal strength and noise level is highly variable.

We therefore need to set an adaptive correlation threshold based on a real time estimation of the environmental noise level.

By using the character of cross correlation, we develop a method to dynamically estimate the noise level through the use of an orthogonal reference sequence. As shown in the equations below, the received complex signal sequence, R(t), contains both a delayed version of the transmitted sequence (the reference sequence followed by data), S(t-τ), and environmental noise, n(t) (1).

$$R(t) = S(t-\tau) + n(t) \tag{1}$$

When we express the correlation between the received sequence R(t) with the reference code, r(t), and the orthogonal reference code $r_o(t)$ at the receiver separately, the correlation result of the reference branch contains both signal and noise potion(2), whereas the correlation result of the orthogonal branch contains solely the noise correlation result ,where N is the number of samples in sequence. (3)

$$C(t) = R(t) * r(t) = \sum_{j=0}^{N-1} R'(t) r(t+j) \tag{2}$$

$$\begin{aligned} C_o(t) &= R(t) * r_o(t) \\ &= \sum_{j=0}^{N-1} S'(t-\tau) r_o(t+j) + \sum_{j=0}^{N-1} n'(t) r_o(t+j) \\ &= \sum_{j=0}^{N-1} n'(t) r_o(t+j) \end{aligned} \tag{3}$$

We can therefore estimate the real time noise level by calculating the standard deviation of the correlation result of the orthogonal branch and set it as an adaptive threshold to detect a reliable synchronization peak.

In order to make use of the scheme described above, we must select a reference code that has an excellent periodic and a periodic autocorrelation function[8], and an excellent cross-correlation function (with values close to zero) with its orthogonal code. Two orthogonal Gold codes are selected as they show reasonable cross-correlation and off-peak autocorrelation values while providing perfect orthogonality in the zero-offset case[12].

As the underwater acoustic receiver uses FSK modulation, we can represent the Gold code using the mark frequency to represent a ‘1’ and a space frequency to represent a ‘-1.’ However, representing the Gold code as a sequence of sinusoids removes its desired auto-correlation and cross-correlation properties. Thus we design a ‘signal shaping’ function before computing a correlation.

We select an appropriate reference length for the Gold code by simulating the symbol synchronization method in Matlab with a packet consisting of Gold code of lengths 3, 7, 15, and 31, followed by 80 symbols of data with a value of 15dB SNR (the same setting as in [3]). Software simulations produced correct results and a sharp, distinct peak for the synchronization point of packets with codes of length 15 and 31, but failed to produce accurate results for packets with 7 and 3. Thus, we select 15 as our minimum reference code length. Figure 2 shows the simulation result for the scheme. The dashed line shows the result of the orthogonal correlation, the thick line shows the resulting threshold and the solid line shows the reference correlation result. The correlation result first rises above the threshold when the packet is present and the synchronization point is accurately detected as the highest point above the threshold.

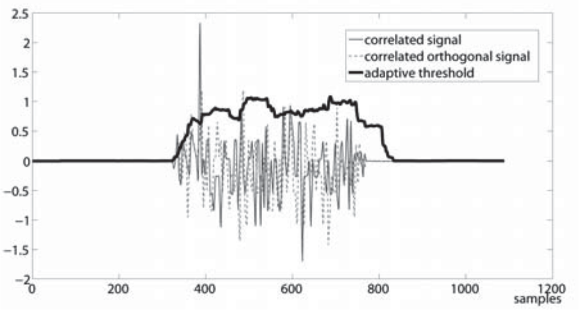

Figure 2. Matlab simulation of symbol synchronization for underwater FSK with a packet consisting of a 15-bit Gold Code sequence followed by 80 bits of data at 15dB SNR

IV. HARDWARE IMPLEMENTATION AND EXPERIMENTS ANALYSIS

Our symbol synchronization scheme for underwater FSK is therefore given in Figure 3. The baseband input signal, *Signal_in*, first passes through a signal shaping block which shapes the input signal from a sinusoid to a signal resembling a square wave. The signal shaping block includes two band pass filters, a down sampler, absolute value operation, and subtraction. The effect of the filters is to remove out of band noise and begin to shape the signal as having regions of different amplitudes. The down sampler and absolute value operations further smooth the signal into an all positive square-like wave. The subtraction produces the desired square like wave consisting of positive and negative numbers ready for correlation with the reference code consisting of ‘1s’ and ‘-1s.’

After the signal passes through the shaping block, it then correlates with the expected reference code and orthogonal reference code. The result of the correlation with the orthogonal branch is squared and summed to compute its standard deviation and serve as a threshold for the reference code correlation operation. The ‘find max index’ logic compares the reference correlation result with the threshold. When the reference correlation result rises above the current threshold, the logic looks for the highest peak above the threshold over two reference length periods to ensure it can find the highest peak that denotes the synchronization point. The peak position’s index is recorded and sent to the demodulator for accurate symbol timing.

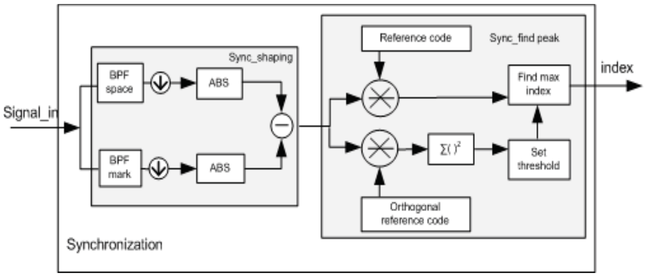

Figure 3. block diagram for symbol synchronization

In order to implement our symbol synchronization block in hardware, we need to consider the design parameters of our acoustic receiver. Recall that the ADC sampling rate is 800kHz and the symbol period is 5ms. These parameters give a total of 4000 samples per symbol. With a reference length of 15 symbols, the entire reference sequence is 60,000 samples. Multiply that by 2 to include the orthogonal reference sequence. Correlating 120,000 samples requires 120,000 multiplications and 119,998 additions. These operations are computationally infeasible to complete in one 800kHz sampling clock period (1.25 us). We thus focus our hardware design implementation on reducing the complexity of the correlation.

Now consider the down sampling ratio. As a square wave only requires one bit to represent the sequence, ideally we could down sample the shaped signal by a factor of 4000, thus reducing the number of samples for correlation to 15 (one sample per symbol). However, in an implementation of a communications system, signals should be oversampled by at least a factor of 4 to account for noise. Thus we down sample the original signal by 1000, to create a reference sequence consisting of 4 samples per symbol, or 60 samples. Performing 120 (including the orthogonal branch) multiplications and 118 additions in one 800Hz (800kHz / 1000) clock period is more feasible, but still computationally intensive.

As the reference sequence and orthogonal reference sequence are represented by ‘1’s and ‘-1’s, we can simplify the correlation operation by replacing the multiplications with a sign selection method: when the reference code is a ‘1’, add the received sample, when the reference code is a ‘-1’ subtract the received sample. Each correlation operation is thus reduced to 59 additions in one 800Hz clock period. We implement the correlation in a 15-stage pipeline, thus requiring only 8 adders (4 per branch). The correlation requires 60 registers to correlate, another 60 registers to shift, and additional internal registers to apply the pipeline. The symbol synchronization has a delay of eighty 16kHz clock cycles, 1 for clocking data in, 60 for shifting the signal into the correlator, 15 for the pipelined additions and 4 for shifting the result to the output.

The rest of the symbol synchronization design is computationally inexpensive as the filters are reused from the demodulator (the synchronizer and demodulator never run simultaneously) and the threshold calculator only requires one multiplier and one adder per 800Hz cycle. The 800Hz clock used for synchronization is source synchronous with the 800kHz sampling clock and 16kHz demodulator clock to avoid clock confliction.

We used both hardware simulation signal and real underwater signal to test our symbol synchronization scheme described by verilog HDL. For both of the two tests, the input to the synchronization block is the output of a digitally down converted signal consisting of the Gold Code of ‘011001010111101’ followed by 80 bits of data with 15dB SNR. The orthogonal code in tests is ‘111010110011011’. The simulation was repeated with SNR values of 5, 10, and 20 dB, each giving the accurate result.

Figure 4 shows the lab set up to obtain a real underwater signal. We use two laptops connected to M-audio data acquisition devices (DAQ) [13] to send and receive signals to and from our analog circuit. The analog board is connected to the acoustic transducers (in the bucket). The real signal is stored in the DAQ as a wav file and then sent to the hardware synchronizer. Figure 5 shows the underwater signal obtains the same accurate synchronization index. Moreover, the performance of the whole modem design (including the synchronizer) has been verified in a swimming pool and fresh water lake. All test results achieved accurate synchronization these different environments with different SNR and multipath spread[14].

Figure 4. lab setup to get real underwater signal

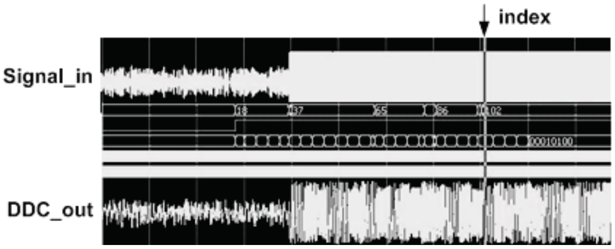

Figure 5. real signal experimental result

Table 2 below shows the FPGA hardware resources required for each component of the acoustic receiver design with standard optimization. These resources are reported for a Xilinx Virtex IV FPGA with the percent of total resources shown in parentheses. The design was implemented and successfully tested on a DINI Group DNMEG AD-DA evaluation board which includes a Virtex IV chip as a prototype [15]. The Virtex IV FPGA is good for prototyping designs as its abundant resources allows designers to quickly implement a design and determine a more suitable (smaller and lower power) FPGA for future optimization.

Table 2 Occupied FPGA resources of design component

	Slices (%)	**LUTs**	**BRAMs**
DDC	309 (1%)	498 (1%)	9
Demodulator	785 (1%)	1874 (1%)	0
Synchronizer	9405 (9%)	33819 (34%)	4

V. CONCLUSIONS

This paper provides a new scheme to build a symbol synchronizer suitable for an underwater FSK acoustic modem by setting an adaptive threshold using orthogonal correlation. The algorithm, design choices and implementation method are described in detail. Both the hardware simulation and real underwater signal experiments show the design meets the timing requirements of the underwater modem and provides accurate synchronization while occupying only 9% of slices and 34% of LUTs in a Virtex IV FPGA. As there are few references describing how to design a hardware symbol synchronizer for an underwater acoustic system, especially how to evaluate real time noise variance, we hope the design choices and implementation details described in this paper will give other researchers a starting point for implementing similar systems.

VI. ACKNOWLEDGMENTS

This material is based upon work partially supported by the China Scholarship Council and partially supported under USA National Science Foundation Grant #0816419 and a USA National Science Foundation Graduate Research Fellowship.

REFERENCES

[1] B.Raul, P. N. Puneet, S. L. Fred, P. C. Anantha, “A Baseband Processor for Impulse Ultra-Wideband Communication,” IEEE Journal of Solid-State Circuits, 2005 vol.40(no.9), 1821-1828.

[2] Xie yongjun, H.xiaoyi, X jing, W deqing, L wangwei, “Implementation of timing synchronization for OFDM Underwater Communication System on FPGA,”. 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication, ASID, 2009, 568-570.

[3] Li ying, B Bridget, K Ryan and Zhang xing, “ Bit Error Rate, Power and Area Analysis of Multiple Implementations of Underwater FSK,**”** International Conference on Engineering of Reconfigurable Systems and Algorithms, ERSA, 2009, 136-142.

[4] B.Stefano, Synchronization of Digital Telecommunications Networks, WILEY, 2002, 210-221.

[5] S. H. Yoon, J. W. Chong, “Packet Detection and Symbol Timing Synchronization Algorithm for Multi-Band OFDM UWB,” IEICE Trans on Communication, E89-B: 2006,.1433-1435.

[6] D. B. Kilfoyle, A. B, Baggeroer, “The State of the Art in Underwater Acoustic Telemetry,” IEEE Journal of Oceanic Engineering, 2000 vol. 25(no.1), 4-27.

[7] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K Ball, “The WHOI Micro-Modem: An acoustic communications and navigation system for multiple platforms,” In Proc. of OCEANS Conference, 2005, 1086-1092.

[8] R. Jurdak, C. V. Lopes and P. Baldi, “Software Acoustic Modems for Short Range mote-based Underwater Sensor Network,” OCEANS 2006-Asia Pacific, 2006, 1-7.

[9] F. Tricia, D. Daniel, U. Chris, I. Ronald, K. Ryan, L. Hua, “Design and Development of a Software-Defined Underwater Acoustic Modem for Sensor Networks for Environmental and Ecological Research,” MTS/IEEE Oceans, 2006, 1-6.

[10] E. M. Sozer and M. Stojanovic, J G. Proakis, “Underwater Acoustic Networks,” IEEE Journal of oceanic engineering, 2000, vol. 25(no. 1), 72-83.

[11] I. F. Akyildiz, D. Pompili and T. Melodia, “Challenges for Efficient Communication in Underwater Acoustic Sensor Networks,” ACM Sigbed Review, 2004, vol. 1(no. 2), 3-8.

[12] C.Chien, Digital radio system on a chip: a system approach. Springer, 2001, 146-168.

[13] M-audio Inc.

http://www.m-audio.com/products/en_us/ProFire610.htm

[14] B. Bridget, Design of a Low-cost Underwater Acoustic Modem for Short-Range Sensor networks, PhD thesis of University of California, San Diego, 2010.

[15] DINI Group.

http://www.dinigroup.com/index.php?product=DNMEG_ADDA