Ultra-Short Baseline Acoustic Positioning System

Timothy Joel Soppet
Computer Engineering Program
California Polytechnic State University
San Luis Obispo, CA
tsoppet@calpoly.edu

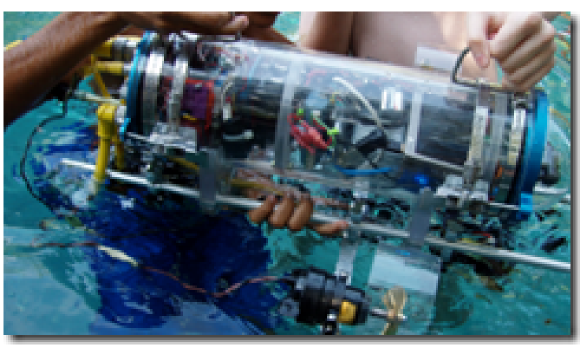
***Abstract*—This paper explains the design, implementation, and testing of an ultra short baseline (USBL) acoustic positioning system for the Amador Valley High School (AVHS) Robotics Clubs Barracuda Mark-X AUV. The system will be used to locate an underwater transducer beacon representing the final waypoint in an obstacle course designed for the AUVSI/ONR RoboSub international collegiate competition.**

***Index Terms*—Ultra Short Baseline (USBL), Transducer Beacon, Autonomous Underwater Vehicle (AUV), Hydrophones**

I. Introduction

Background Information

The Association for Unmanned Vehicle Systems International (AUVSI) created an international collegiate student competition, in 1998, to advance research and interest in Autonomous Underwater Vehicles (AUVs). In 1999, students at Amador Valley High School discovered the RoboSub competition and after a petitioning process entered their AUV into the 3rd annual AUVSI AUV competition. Since then, the AVHS Robotics Team consistently placed in the top seven among the elite competitors and is held in high regard amongst competitors and competition coordinators.

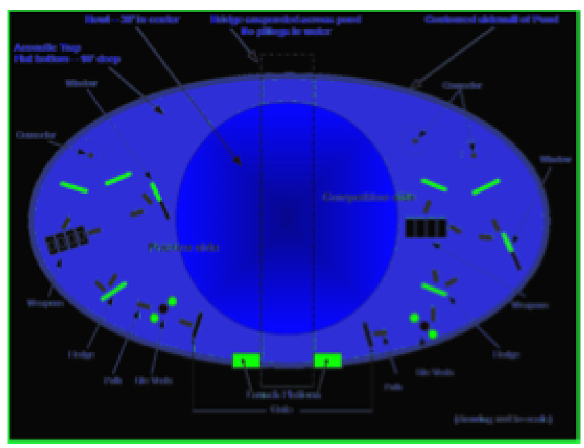


The current autonomous submarine shown above competes in the annual competition held at the Space and Naval Warfare Systems Center SSC SD TRANSDEC Facility in San Diego California shown on the next page.

An underwater obstacle course, shown below, is setup in the TRANSDEC pool, which the autonomous vehicles must navigate. The mission specifications for AUVSI and ONR's 14th International RoboSub Comptition requires entrants to navigate through an underwater gate, follow an orange pipe on the pool floor to a series of buoys, identify a specific colored buoy and proceed to hit it. The sub must then proceed to a

Transducer Evaluation Center (TRANSDEC)

series of colored window panes, locate a specific pane and shoot a marker through the window. Next, the sub must locate a series of boxes on the floor of the arena and drop a marker in a predefined box.



The final obstacle requires that the vehicle be capable of detecting and navigating to an acoustic beacon on the bottom of the pool. The high school team has lacked sufficient educational background to implement a reliable passive hydrophone detection system. The goal of this project is to implement such a system.

Related Works

The increased prevalence of AUVs in military, industrial and research applications has revolutionized the area of underwater

acoustic signal positioning. The three major categories of underwater acoustic positioning systems are Long Baseline (LBL), Short Baseline(SBL) and Ultra-Short Baseline(USBL) systems [3]. Baseline classes are differentiated based on the distance separating the active sensing elements.

TABLE I: Baseline Lengths

Type of Acoustic Positioning System	Baseline Length
Ultra-Short Baseline	<10cm
Short Baseline	20m to 50m
Long Baseline	100m to 6000m+

The advantage of using the USBL class acoustic positioning system on AUVs is the compact baseline sensor array configuration (¡10cm) in comparison to the SBL system which requires 20 meters minimum baseline distance, as shown in Table 1.

Problem Statement

Design, implement and test an ultra-short baseline underwater acoustic positioning system to detect a 22 40 kHz ultrasonic acoustic signal at a frequency of 0.5 Hz produced by a single static transducer at a range of approximately 108 m, at a maximum depth of 11.4 m within a 2.7 m2 surfacing area, for the Amador Valley High School (AVHS) Robotics Clubs Barracuda Mark-X AUV. The USBL underwater acoustic positioning systems operating environment will be the SPAWAR Systems Center TRANSDEC sonar testing facility, in San Diego, CA [7].

II. Implementation

System Overview

Fig. 1, shown below, depicts the functional environment that the USBL acoustic subsystem will fit into and includes all interface points. The system begins with the mission software contained on a BeagleBoard in the submarine. The mission software interacts with the various hardware components through purpose-built driver software, which in turn interacts with the specific hardware components. One of those hardware components is the Acoustic Subsystem, which is also resident in the submarine.

The USBL acoustic subsystem, shown in Fig. 2, obtains input from the external environment via an array of four RESON TC4013 Miniature Reference Omni-directional hydrophones. The piezoelectric sensor element contained inside the rubber nibble produces an electrical signal on the order of approximately 40mv, which propogates up the coaxial cable. The electrical signal is amplified to satisfy the requirements of the dspblok ad96k42 audio I/O platform. The dspblok stack converts the signal to a digital representation, filters extraneous noise and outputs the signal difference time of arrival via a UART connection to a BeagleBoard, the primary CPU on the Barracuda Mark-IX AUV.

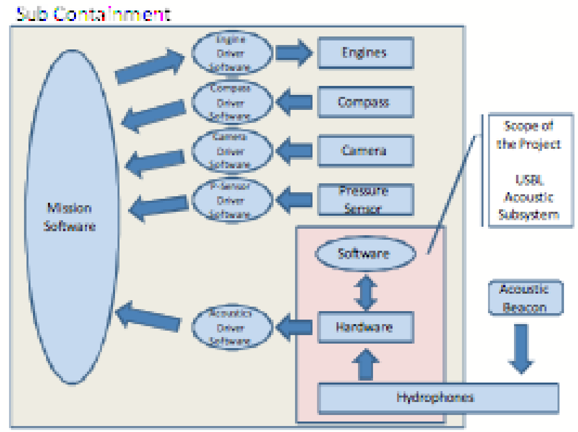


Fig. 1: USBL Acoustic Subsystem Functional Environment

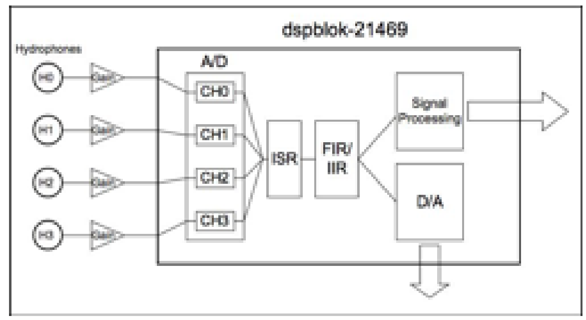


Fig. 2: USBL Acoustic Subsystem Flow Diagram

III. Hardware Subsystem

Signal Conditioning

The signal produced by the hydrophones is approximately 40mV; however, the Wolfson WM8731 ADC on the dspblok ad96k42 platform requires a minimum input signal voltage of 1.8Vpeak. In order to satisfy the requirement, I designed a simple 50x non-inverting amplifier, as shown in Fig. 3, using a LM711JN operational amplifier.

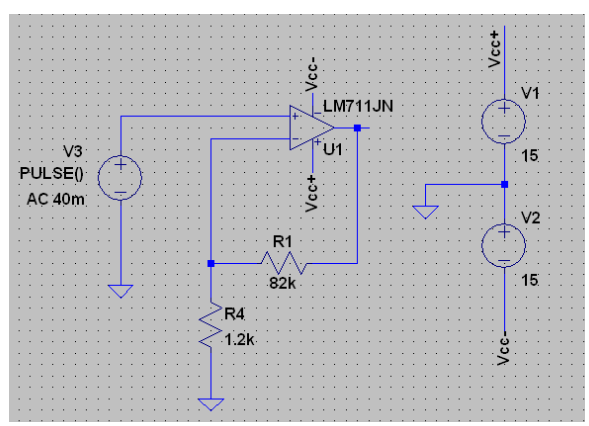


Fig. 3: 50x Amplifier Circuit Schematic

Digital Signal Processor

1) dspblok 21469: The dspblok 21469 is a base-line 60 mm2 integrated dsp platform supporting the ADSP-21469

SHARC DSP from Analog Devices. The ADSP-21469 processor features a 450 MHz internal clock, 16-bit DDR2 external memory and DMA engine. Supported interfaces include serial ports (SPORTs) registers for transmitting/receiving signal interrupts, UART interfacing for transmitting/receiving character strings to/from the BeagleBoard primary processor and SPI interfaces for cross-platform communication [1].

2) dspblok ad96k42 audio I/O board: The dspblok ad96k family of audio I/O platform supports the Wolfson WM8731 96kHz 2&4 channel low power stereo CODEC with stereo 24-bit multi-bit sigma delta ADCs and DACs. The WM8732 CODEC supports ADC and DAC sampling rates ranging from 8kHz to 96kHz with an ADC SNR of 85dB at 1.8V and a DAC SNR of 95dB at 1.8V [8]. Analog I/O is provided via a maximum of 4-input / 4-output SMA audio connectors for analog signal processing; however, the project requirements only specified audio signal input requirments. The dspblok ad96k42 platform supporting 4-input / 2-output audio connectors was selected in order to provide useful analog signal debugging as well as future application requirement flexibility [6].

3) dspblok ps-uart pwr & misc I/O board: The PS-UART pwr & misc I/O combination platform supports UART connection for external communication and power supply for powering the entire stack of dspblok platforms.

Software Filter Algorithm

The software filter algorithm is based on a finite impulse response (FIR) digital filter design. The FIR filter was designed using MATLAB, creating an equirripple band-pass filter centered from 70kHz to 76kHz, with a pass gain of 1dB and a stop gain of -80dB. The designed filtered had a total of 258 taps. The hydrophones voltages are polled at a constant sample rate of 150 ksps per hydrophone from the analog to digital converters. These sample values are converted to floats and stored in a buffer. After the buffer is filled, the FIR filter algorithm is run on the samples, and the filtered values are stored in an output buffer. The output buffer is iterated over, checking to see if the voltage passes a given threshold, if the ratio of output vs. input is less than a given threshold, and if the number of high signals surpasses a given threshold. This decreases the likelihood of false positives occurring.

When a signal is received from the first hydrophone, a flag is set in the program and the time the signal received is recorded. The DSP will then check for a signal from the second hydrophone, and when received will record the time. It will then compare the difference between the first and second hydrophone, and send a frame over RS-232 which contains the time difference between the pings, as well as which hydrophone received the ping first. To prevent synchronization issues which can occur from receiving a ping on one hydrophone and missing it on another, a timeout occurs if both hydrophones dont receive a ping within a maximum time period.

IV. SOFTWARE SUBSYSTEM

The amplified signal is then input into the digital signal processor where the analog to digital converter converts the analog signal into a corresponding digital signal and stores the resulting signal representation before triggering a signal interrupt. The system catches the interrupt, pushes the stored input signal representation to the digital to analog converter and the resulting analog signal is output from the system. In parallel, the stored signal representation is also sent to the filtering module. The signal filter module removes higher and lower noise from the signal represenation and then pushes the, software, conditioned signal to the Signal Processing module where the signal is parsed to check if a pulse was detected. If a pulse was detected, the system calculates the time of reception and sends via a UART connection to a main processor.

Interrupt Service Routine

The initialization routine unmasks the SPI_SP0 register enabling interrupts to be triggered. If an interrupt occurs the system refers to the interrupt vector table which maps the interrupt register to a corresponding interrupt service routine. The interrupt service routine takes the input buffer specified in the init_spi_device routine for the analog to digital converter device and saves the address of the tail of the input buffer.

For debugging purposes a transmit routine may be called which copies the tail of the input buffer into a transmit buffer. The transmit buffer is then forwarded to the digital to analog converter for transmission.

Software Filter

The USBL system requirements specify a range of frequencies from 22kHz to 40kHz for use in the competition. Therefore, multiple filter coefficient files are required for the system to support the required range. The software implementation of the software filter algorithm is based on a finite impulse response (FIR) digital filter design. The FIR filters were designed using MATLABs FDATOOL user interface, creating equirripple band-pass filters centered at 22kHz to 40kHz respectively +/- 1kHz, with a pass gain of 1dB and a stop gain of -60dB. The FDATOOL user interface produced coefficient header files for each of the frequencies specified in the requirements, containing 129 taps per frequency selection.

After the ISR stores the digital representation of the input signal in the input buffer corresponding to the hydrophone from which the signal originated, the FIR function is called taking the coefficient array, input signal buffer and an output buffer as parameters and digitally filters the signal representation in the input buffer.

Signal Processing

A wide array of techniques and algorithms have been developed to determine the position of a target using USBL acoustic positioning systems. However for the scope of this project, a simplistic, robust and minimally computationally intensive method is desired. The most basic and minimally computational intensive method is differential time of arrival (DTOA). DTOA is simply an exercise involving capturing the time of arrival that each sensor detects a pulse and comparing arrival times to determine the order of arrival. Based on the

order, a relative bearing may be computed. After repeatedly sampling and modifying the AUVs bearing, it is possible to hone in on the target of interest within a meters error. The downside of DTOA is the amount of distance separation required in order to be able to compare the time of arrivals (approximately 1 meter), which while not unreasonable still is not ideal.

A more robust method of determining a targets bearing is phase comparison. The theory behind this method boils down to simply overlaying two signals and observing the order in which the signals were received. In order to implement this method, the signals much be converted into their respective digital representations. This is accomplished with built in analog to digital converters on the dspblok ad96k42 audio I/O board. Next, the zero-crossing point is chosen within a single phase of each signal being compared. The difference in the zero-crossing points correlates to the difference in time that the respective sensors producing the signal received the sound wave produced by the transducer beacon. If we extrapolate this method and compare every signal against all other signals received in a given sampling time, then it is possible to determine the exact order that the sensors received the sound wave from the beacon. Thus, it is possible to determine the exact bearing of the beacon in relation to the array of sensors [3].

V. Testing

Testing Hardware Signal Conditioning Subsystem

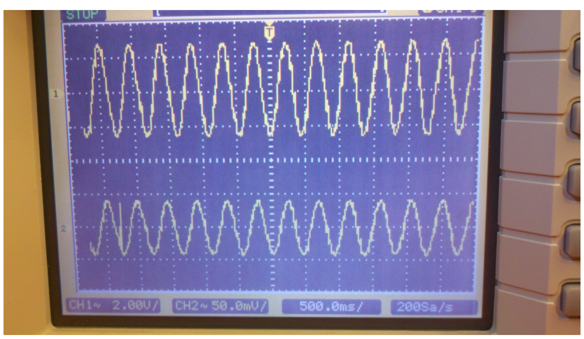


Fig. 4: Test signal amplification circuit on signals produced by function generator

The first step in testing the amplification circuit was to verify the gain factor by comparing the input signal produced by a function generator outputting a 40mV sinusoid signal against the signal output by the amplification circuit. As is shown in Fig. 4, the channel-2 signal is 40mV and the output signal labeled channel-1 is 2.4V which is approximately a 40x gain factor as was expected.

The next step was testing the amplification circuit using an input signal produced by the TC4013 hydrophone. Fig. 5 shows the change in amplification of the output signal from the input signal. The input signal while not consistent is on average greater than the 40mV expected input and the amplified signal remains above the minimum required voltage for the dspblok ad96k44 audio i/o board.

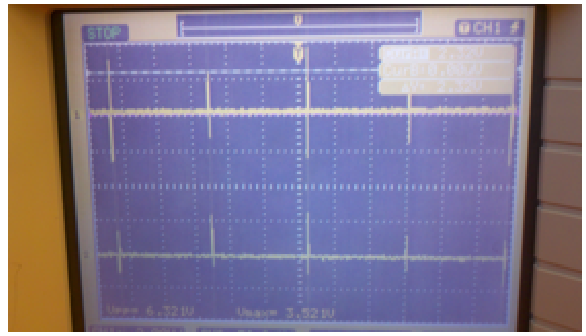


Fig. 5: Test signal amplification circuit using hydrophones

Testing System Wide Signal Transformation

As Fig. 6 shows, the end-to-end path of the signal is quite complex. In order to verify the input signal satisfied the dspbloks input signal requirement, I utilized a convenient signal output debugging feature to verify the signal successfully traverses the internal communication of the dspblok audio i/o board to the adsp-21469 dspblok and back to the audio i/o board.

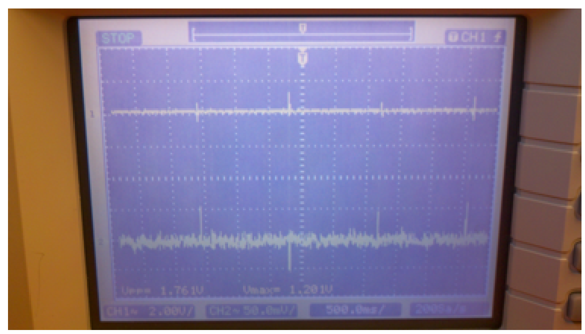


Fig. 6: upper signal: final output signal from dspblok
bottom signal: pre-amplified signal from hydrophones

The primary purpose of testing the end-to-end system is to determine the effect of noise in the system. Notice the ratio of noise in relation to the signal pulses. It is clearly shown in the snapshot that system results in a clear line on Channel-1 which shows the signal output after the signal is processed by the digital to analog converter. Fig. 7 and 8 are individual pulses sampled at a higher rate than those in Fig. 6.

Fig. 7: Signal post amplification and processing by dspblok

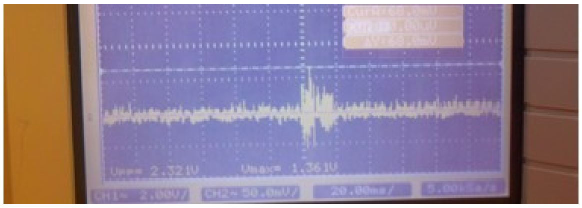


Fig. 8: Raw signal produced by the hydrophoes before amplification

VI. Conclusion

Concluding Statement

I was successful in demonstrating that the raw signal could be captured, filtered, and information about the signal could be delivered over the UART. However, I encountered some technical problems integrating the code written in assembly with other code written in C. These integration challenges prevented me from completing the end to end integration and testing required to achieve all the goals outline at the beginning of the project as Phase I.

Moving Forward

The tasks yet to be completed involve replacing the on-board signal filtering logic with software to invoke the FIR filter coefficient files that were developed using MatLab so various frequency ranges can be accommodated. A second task is to complete the implementation of the DTOA analysis algorithm that pinpoints the location of the acoustic beacon relative to the four hydrophone sensors. I may also need to rewrite the input/output processing logic in C to simplify the integration process. The completion of these tasks over the summer will enable the Barracuda Mark-IX AUV to successfully compete in the 2011 competition.

After conclusion of the 2011 competition in July, phase 2 of the project can commence, which is to implement the more robust differential phase analysis algorithm for pinpointing the location of the acoustic beacon relative to the hydrophone sensors.

On a personal level, this project brings me full circle. I started off as a high school student competing in this annual competition with a very narrow perspective. My college coursework coupled with my professional experience have filled in many of the gaps, which I am now positioned to use to influence the future high school students. The drop-out rate for engineering majors is well above fifty percent. There were many times when my high school engineering experience carried me through when coursework got difficult.

My personal goal, in addition to helping advance the AUV project, is to at a minimum give the team members an opportunity to encounter the material before entering college and at a maximum allow them to obtain hands on experience using the equipment to actively learn how to design, implement, test, and debug real world problems by doing.

Lessons Learned

I learned a great deal about the trade-offs required during the design and implementation process as competing constraints are evaluated and weighed. For example, this project had a series of design constraints ranging from the physical space available to house the solution to the choice of boards available on the market. With each choice, there were trade-offs that had to be considered; how much work was provided by each solution versus what were the integration challenges with getting each of the component pieces to work together. In many cases, components provided very powerful capabilities but the lack of concise documentation limited my ability to seamlessly integrate the parts into a comprehensive solution. This resulted in many early-morning calls to vendors in the United Kingdom and hours of time spent talking with technical support.

VII. Acknowledgment

I want to thank Professor Clark, my faculty advisor, for sponsoring and providing valuable guidance and advice on approaching and breaking down the project into managable and testable benchmarks. I also want to thank the Amador Valley High School robotics team for allowing me the opportunity to design and implement a hydrophone system for the Baracuda Mark-X AUV.

References

[1] Jaffre, F.M., Austin, T.C, Allen, B.G., Stokey, R., Von Alt, C.J. *Ultra Short BaseLine Acoustic Receiver/Processor*, Ocean-Europe 2005, 0-7803-9103-9/05, 2005

[2] Vickery, K. *Acoustic Positioning Systems. A Practical Overview of Current Systems*, Autonomous Underwater Vehicles, AUV.1998.744434, 1998

[3] *Acoustic Positioning Systems. A Practical Overview of Current Systems*, http://www.analog.com/en/processors-dsp/sharc/ADSP-21469/products/product.html, 2011

[4] http://www.analog.com/en/processors-dsp/sharc/ADSP-21469/products/product.html, 2011

[5] http://www.avbotz.com/about/, 2011

[6] http://www.danvillesignal.com/dspblok/dspblok-dsp-modules.html, 2011

[7] *Preliminary Mission for the 14th Annual 2011 AUV Competiton "RoboLove"*, http://www.auvsifoundation.org/AUVSI/FOUNDATION/Competitions/RoboSub/Defaul 2011

[8] *WM871: CODEC with headphone driver*, http://www.wolfsonmicro.com/products/codecs/WM8731/, 2007