

*Real Time Implementation of Passive Acoustic Source
Localization System on an Autonomous Underwater Vehicle*

Submitted by

Yow Zhi Hung Joshua

Department of Electrical & Computer Engineering

In partial fulfillment of the
requirements for the Degree of
Bachelor of Engineering
National University of Singapore

1. Abstract

This project focuses on the problem of localizing an underwater acoustic pinger in a reverberant environment under the constraints of a real time implementation on an Autonomous Underwater Vehicle.

This paper proposes a novel algorithm to make use of both distinctive features of an acoustic ping, the rising edge of the ping and the phase of the sine wave portion, to identify the location of the source. A combination of a step detector working on the envelope of the signal and the high resolution MUSIC algorithm is used to gain a more accurate and robust estimate of source location than can be obtained from each of these techniques alone. The algorithm is implemented on MATLAB and verified using data recorded by the hardware on BBAUV.

An implementation of the proposed algorithm is carried out on the Bumblebee Autonomous Underwater Vehicle (BBAUV). The resulting implementation is able to maintain the performance of the algorithm while maintaining low latency under a resource-constrained embedded hardware system.

2. Contents

Abstract	2
List of Figures	5
List of Tables	6
Acknowledgements	7
Background	8
Objective	9
System Specifications	9
Challenges	10
Literature Review	12
Near Field Signal Model	12
Far Field Signal Model	14
Steering Vector	15
Evaluation of both Signal Models	15
Review of Existing Source Localization algorithms	17
Rising Edge based Algorithms	18
Phase Difference Based Techniques	21
Algorithm Description	23
Time of Arrival Estimation	23
MUSIC Search	24
TDOA Refinement	25

Range search	25
MATLAB Implementation	26
Implementation	27
Hardware System	27
Teledyne Reson TC4013 Hydrophones	28
Custom Pre Amplifier and Band-pass Filter Board	28
NI 9223 Analog Input Module.....	29
National Instruments sbRIO-9602	31
Embedded Software System	32
Goals and Challenges.....	32
Software architecture	32
Real Time Component	34
FPGA Component.....	38
Results.....	41
Conclusion	42

3. List of Figures

- Figure 1: Example of a recorded underwater ping from an acoustic pinger 8
- Figure 2: Multipath interference where the Line of Sight signal and 1 reflected signal each from the surface and bottom of the water is shown. 10
- Figure 3: Illustration of the source localization problem with an equispaced linear array: 13
- Figure 4: Illustration of an equispaced linear array 14
- Figure 5: Initial start of an acoustic ping on 4 different hydrophones 17
- Figure 6: Two graphs of the start of a recorded acoustic pulse showing the effects of the band-pass filter. 20
- Figure 7: 3D plot showing the MUSIC Spectrum across azimuth and elevation. 25
- Figure 8: Hardware System Flow Diagram. 27
- Figure 9 Front 3 Hydrophones arranged into a square array of separation 1.5cm **Error!**
- Bookmark not defined.**
- Figure 10 Full Hydrophone Array Setup **Error! Bookmark not defined.**
- Figure 11: Printed Circuit Board Layout of the Custom Preamplifier and Band-pass Filter Board 29
- Figure 12: NI9223 Analog Input Module from National Instruments 30
- Figure 13 National Instruments sbRIO-9602 31
- Figure 14: Software Architecture of the embedded software system 33
- Figure 15: Front Panel of the LabVIEW Real Time Top Level VI 34
- Figure 16: Comparison of Sampling rate and time taken for the algorithm to run. In the region of 100 kHz to 125 kHz, the processing time does not reduce by much, especially in comparison with the 200 kHz to 250 kHz range. 35

Figure 17: LabVIEW FPGA Functions Palette on the left is much smaller than the full LabVIEW Functions Palette on the right. The difference in size indicates the difference in number of functions available on both platforms 39

4. List of Tables

Table 1: Interleaving of 4 Analog Channels into a single DMA FIFO. 40

5. Acknowledgements

First of all, I would like to thank my supervising professor Prof Mandar Anil Chitre for his invaluable guidance and insights on this project.

I would also like to thank the BumbleBee Autonomous Underwater Vehicle Team and team captain Huan for giving me the opportunity to work with the vehicle and develop the passive acoustics system and providing the necessary hardware, software and administrative support.

I would also like to thank my predecessors Huang Yong Chang and Vivek for laying the groundwork for this system. The team sponsors National Instruments and Teledyne Reson also provided hardware and software for the purpose of this project.

6. Background

Acoustic source localization is the task of locating a sound source using a passive sensor array. This has applications in a wide range of underwater tasks, including passive sonar and underwater navigation.

A scenario of particular interest is when the signal source is a narrowband pinger emitting acoustic pulses, or 'pings', as shown in Figure 1. These pingers are often equipped on underwater equipment to allow a listener to deduce the location of the pinger. This scenario commonly occurs in two major areas of interest in passive acoustic source localization; firstly, when searching for equipment underwater, like an aircraft black box with an equipped pinger, or in underwater navigation, where the position of the receiver is deduced based on the relative position to pingers with a fixed known location. Therefore, the task of localizing an acoustic pinger has wide-ranging and important applications.

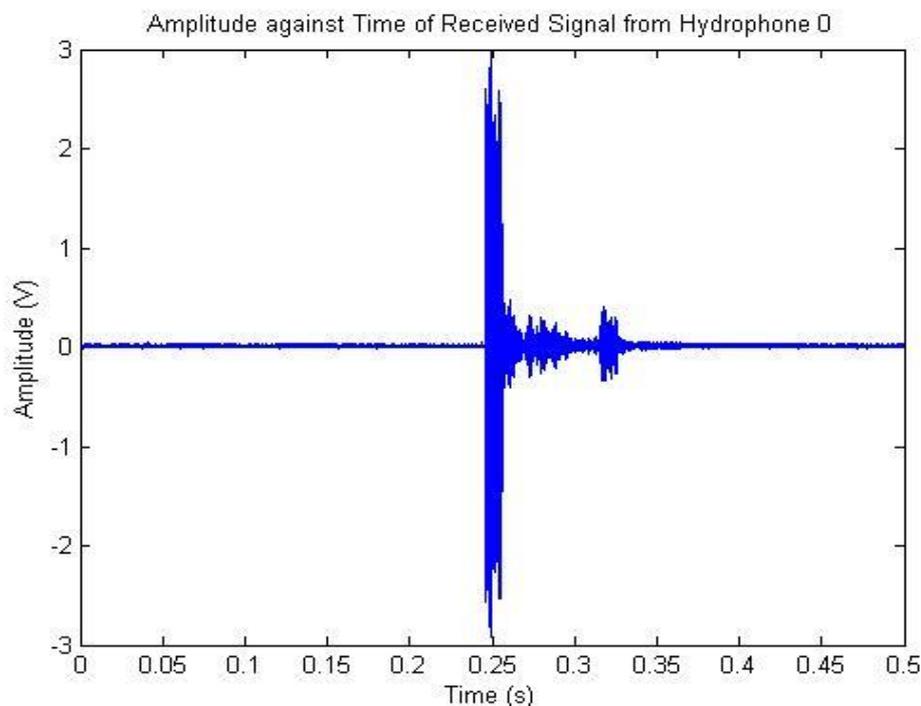


Figure 1: Example of a recorded underwater ping from an acoustic pinger

7. Objective

The goal of this project is to design and implement an underwater acoustic localization system on the Bumblebee Autonomous Underwater Vehicle (BBAUV), to enable the AUV to navigate to an acoustic pinger in the quickest possible time.

7.1. System Specifications

In order to help the vehicle navigate to the pinger in, the system has to meet several specifications.

1. The system needs to be able to estimate the Direction of Arrival (DOA) of the signal accurately and robustly. DOA is made of 2 parts: the azimuth, or angle on the x-y plane to the source and the angle of elevation of the source.
2. The system needs to perform in shallow water channels with depths of between 2m to 10m.
3. The system needs to track a pinger pinging at a rate of 2Hz and a frequency of between 20 kHz to 45 kHz.
4. The system needs to have a low enough latency ($<0.2s$) to allow the vehicle to get real time updates while moving (about 1m/s). Latency in this system is defined as the delay between reception of the acoustic signal and delivery of the source location information to the main computer of the AUV. A high latency would cause mismatch between the source location information received by the computer and actual location, causing the vehicle to make decisions based on out of date information. .
5. The system needs to be able to detect and localize a pinger located up to 20m away.
6. The system needs to be able to give an estimate of the distance to the pinger. This helps the vehicle tune its movements to reach the pinger in the least amount of time possible.

8. Challenges

There are several challenges which must be overcome to meet the specifications stated above.

These are described in detail below.

Firstly, in shallow water, acoustic signals tend to reflect off the surface and bottom of the water and any nearby surfaces, including the vehicle itself. This is illustrated in Figure 2. The reflections result in multiple copies of the original signal corrupting the signal received at the hydrophones. These copies are difficult to remove due to their similarity with the Line of Sight signal and make it difficult for phase-difference based algorithms to compute the source location.

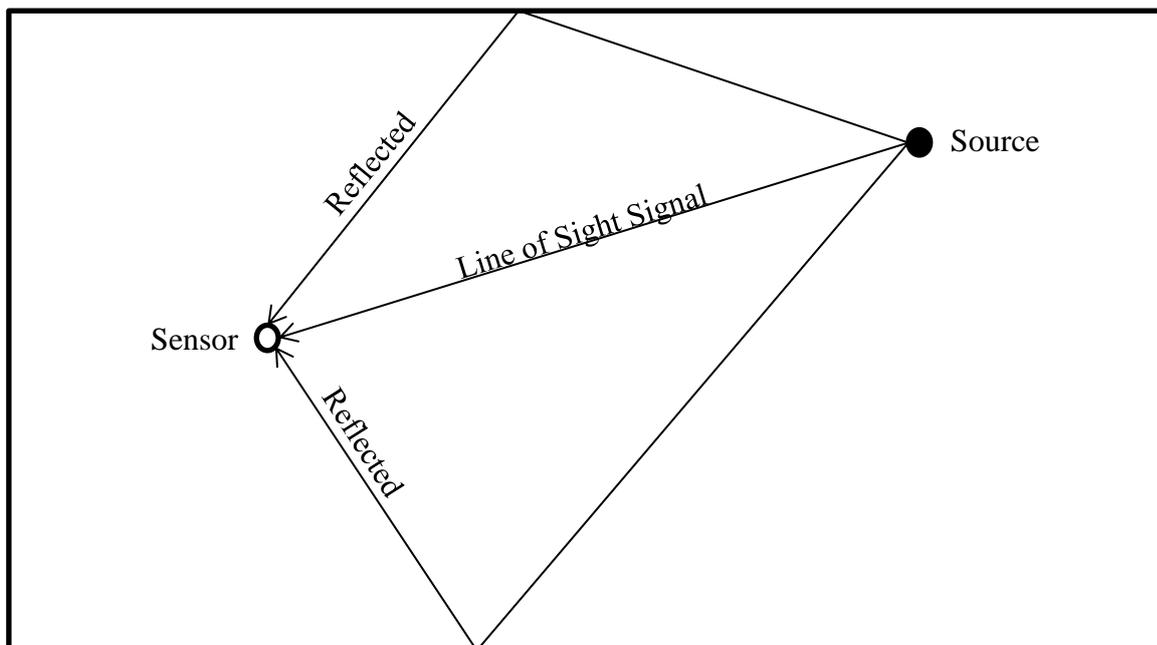


Figure 2: Multipath interference where the Line of Sight signal and 1 reflected signal each from the surface and bottom of the water is shown. In reality, the signal can take many different paths to the sensor by reflecting off surfaces, resulting in multiple copies of the signal corrupting the Line of Sight signal.

Secondly, the size of the hydrophone array is limited by the size of the AUV it is being deployed on. The size of BBAUV is 1.1m by 0.7m by 0.4m. This puts limits on the size of the

hydrophone array and therefore limits the magnitude of the Time Difference of Arrival (TDOA) of the signal at the hydrophones. This makes any error in the TDOA estimate larger in comparison to the magnitude of the TDOA estimate and hence makes the algorithm less accurate.

Thirdly, the hardware resources available for the implementation are very limited. Due to space and power constraints on the AUV, The entire algorithm is to be implemented on an embedded hardware system, with limited disk space, RAM, and CPU bandwidth. This affects the speed at which the algorithm can run and ultimately poses a limit on the complexity of the algorithm.

Lastly, noise present in the environment can affect the accuracy of the algorithms. One of the most significant sources of noise is noise generated by the thruster. The Signal to Noise Ratio (SNR) can go to as low as 5dB when the pinger is up to 20m away from the AUV and the thrusters are running. This can affect the accuracy and robustness of the algorithm.

9. Literature Review

Before proceeding with the analysis of the current literature, it is necessary to define the signal model. Two models will be used in this paper: the Near Field Signal model and its far field approximation, the Far Field Signal model.

9.1. Near Field Signal Model

The narrowband acoustic pulse can be described by the following equation:

$$s(t) = \begin{cases} A \sin(\omega t - \phi) , & 0 \leq t \leq T \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

where A is the amplitude of the signal and T is the duration of the acoustic pulse.

As previously illustrated in Figure 1, in a reverberant environment like underwater environments, the signal can travel to the receiver via several different paths. This is known as multipath propagation and results in several delayed and attenuated copies of the source signal appearing at the receiver in addition to the line-of-sight signal. The signal received at a receiver is therefore written as:

$$x(t) = \sum_{m=1}^M \alpha_m s(t - \tau_m) + n(t) \quad (2)$$

where M is the total number of paths (reflected and direct) that the source signal can take, α_m is the attenuation associated with the signal travelling along the m th path, τ_m is the delay associated with the m th signal due to time taken to travel to the receiver along the m th and $n(t)$ is the noise received at each signal.

Figure 3 below shows a source in the near field of 3 sensors. The differences between the paths r_1 , r_2 and r_3 correspond to the Time Difference of Arrival of the line of sight signals. Since the relative locations of sensors in the array is known, once TDOA is obtained, we can

easily solve for the exact location of the source through a combination of trigonometry and one of many robust linear algebra methods like Gaussian Elimination [1].

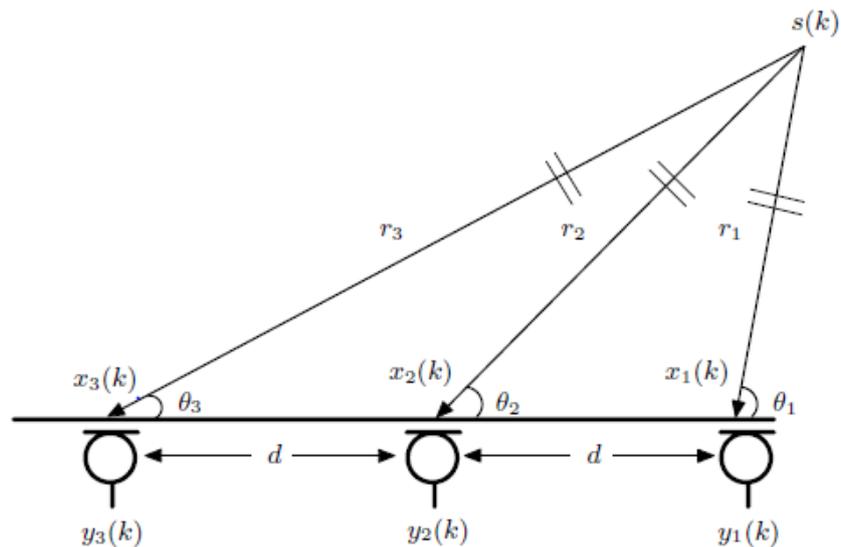


Figure 3: Illustration of the Near Field source localization problem with an equispaced linear array: the source $s(k)$ is located in the near-field, and the spacing between any two neighbouring sensors is d . [J. Benesty, J. Chen and Y. Huang, *Microphone Array Signal Processing*, Berlin Heidelberg: Springer, 2008.]

9.2. Far Field Signal Model

The Far Field signal model is an approximation of the Near Field Signal model used when the size of the sensor array is small compared to the distance to the source. In such a case, the incoming wavefronts of the signal can be considered to be straight and parallel to each other rather than spherical about the source in the Near Field signal model. This can be seen in Figure 4 below.

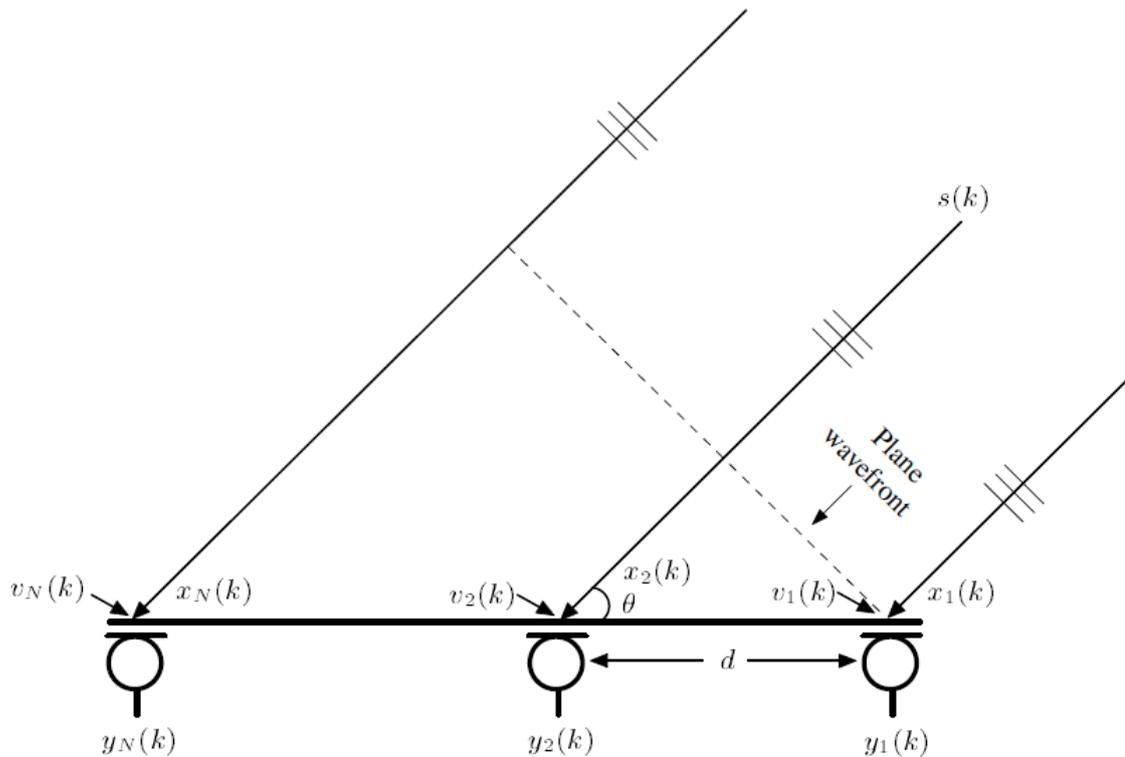


Figure 4: Illustration of an equispaced linear array, where the source $s(k)$ is located in the far field, the incident angle is θ , and the spacing between two neighboring sensors is d . [J. Benesty, J. Chen and Y. Huang, *Microphone Array Signal Processing*, Berlin Heidelberg: Springer, 2008.]

Using this model, the time delay (τ_n) between first sensor and the n th sensor is given by the equation below:

$$\tau_n = \frac{(n-1)d \cos \theta}{v} \quad (3)$$

where d refers to the spacing between the sensors, θ is the angle of arrival of the signal at the array and v is the speed of sound in water.

As can be seen from (3), the time delay is a function of only the angle at which the signal is arriving at the sensor array and does not involve the distance between the array and the source. Hence, the Far Field signal model is an approximation which discards information about the distance to the sensor in exchange for simplicity in calculating the DOA of the signal.

9.3. Steering Vector

For complex analysis of the signal, the time delays are often expressed in the form of a vector known as the Array Steering Vector. The Array Steering Vector is a vector which contains the associated phase shifts of a narrowband signal given that the source is assumed to be in a certain location. The phase shifts are due to the Time Difference of Arrival of the signal between the first sensor and the n th sensor. The Array Steering Vector is given in the following form:

$$\begin{pmatrix} e^{i\omega\tau_1} \\ \vdots \\ e^{i\omega\tau_n} \end{pmatrix}$$

Where ω is the angular frequency of the narrowband signal and τ_n is the time delay between the first and the n th sensor.

9.4. Evaluation of both Signal Models

Both models have their advantages and disadvantages. With the Far Field Signal Model, it is much easier to compute the DOA of the signal. However, it can only provide Direction of Arrival (DOA) information and no information about the distance to the source can be obtained.

On the other hand, if the Near Field Signal model is used, we can find the exact location of the source in 3D based on the wave-front curvature of the incoming signal. However, the Near Field Signal model is much more computationally complex to deal with. It is also only applicable to situations where the array size is significant to the distance to the source; if not, the Near Field Signal model can only estimate distance poorly and has no significant advantage over the Far Field model.

Given the size limitations and the computational hardware limitations of the platform it is to be deployed on, the Far Field Signal model is likely to be most applicable model. However, the Near Field Model will be required to provide distance estimate of the source. Thus the algorithm will have to find a way to integrate both models.

9.5. Review of Existing Source Localization algorithms

There are 2 main features of the acoustic ping which are used to estimate their Time Difference of Arrival (TDOA). The first is the rising edge portion of the ping where there is a sharp increase in amplitude of the signal. The second is the sine wave portion of the ping and the difference in their phases when they are received by each of the hydrophones. These are aptly illustrated in Figure 5 below.

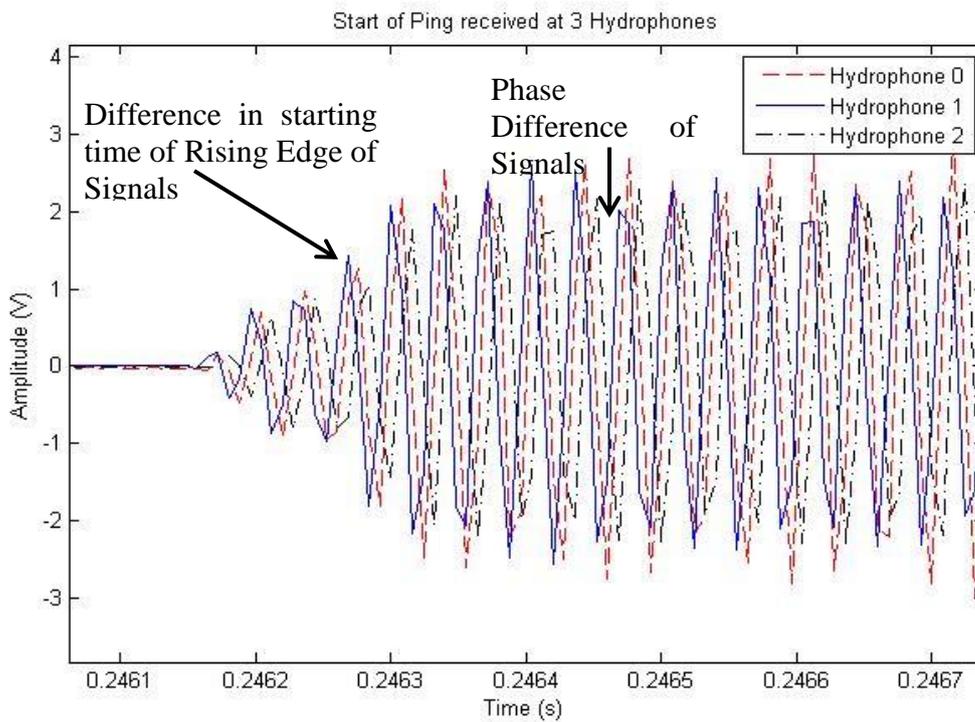


Figure 5: Initial start of an acoustic ping on 4 different hydrophones, highlighting the 2 features which are used to identify the Time Difference of Arrival at each hydrophone.

The algorithms to estimate TDOA and hence source location are separated into two classes based on which feature they use to distinguish TDOA. The first kind is referred to in this paper as Rising Edge based algorithms while the second is referred to as Phase Difference based algorithms.

9.5.1. Rising Edge based Algorithms

Rising Edge based algorithms attempt to compute the Time of Arrival (TOA) of the incoming acoustic pulse. With the Time of Arrival of the ping at each sensor, their Time Difference of Arrival can be found by subtracting the respective Times of Arrival. The Time Difference of Arrival can then be used in either the Near Field or Far Field signal models to compute the Direction of Arrival.

Matched Filtering

For acoustic pulse detection under zero-mean, white Gaussian noise conditions, the optimal detection scheme is the matched-filter [2], which is commonly used in radar. However, in a multipath environment, the incoming signal is corrupted with multiple copies of itself, making it hard to distinguish the peak of the matched filter output and hence the TOA of the pulse.

9.5.2. Thresholding

The most straightforward rising-edge detection scheme is the standard threshold-crossing technique whereby the time at which the incoming signal rises above a pre-set threshold is used as the TOA estimate [3]. It is a quick technique and requires very little computation power to perform. The threshold level chosen here is important. A threshold which is too low results in the possibility of false triggering due to noise spikes while a threshold too high may limit the use of the system, as at longer ranges, the SNR of the signal decreases. However, this method results in a biased estimator of TOA [3] which is SNR dependent and hence cannot be easily corrected for.

McMullan and Delanghe [3] proposed a modified technique which compensates for this uses the technique analytically fits the rising edge of the pulse using two threshold levels and the

corresponding threshold crossing times to give an asymptotically unbiased estimate of the TOA. However, the system is still susceptible to errors due to the variation in SNR.

Step Detection

A step detection algorithm can also be used to detect the TOA of the pulse. The envelope of the pulse is first detected using a Hilbert Transform. Many step detection algorithms can then be used to good effect to preserve the edge while removing the noise, including the median filter and total variation denoising. While complex and robust methods like Canny or Sobel edge detection exist, these prove too computationally intensive for the purpose of this paper. A simple technique like a gradient detection or a variance measurement over a sliding window can then distinguish the edges quickly and reliably. For a simple purpose like this, a median filter and variance measurement over a sliding window was determined to be sufficient to detect the edge and reject noise spikes effectively while not being too computationally intensive.

Additional Notes

While simple and robust Rising Edge detection techniques exist, they perform best when the array size is large and their accuracy over small array sizes decreases. This is because as the distance between hydrophones decreases, the difference in distance the sound has to travel is reduced and hence the magnitude of the TDOA decreases. As such, any error in TOA estimation results in a larger percentage error of TDOA estimation and has a larger effect on the accuracy of the eventual estimate of the source location. As the size of BBAUV is small, the size of the array is restricted. Thus these techniques may not work well enough.

Another point to note is that due to the square pulse making up the rising edge having many frequency components, the signal cannot be put through a tight band-pass filter as this would spread out the rising edge and result in a less precise estimate, as shown in Figure 6. As such,

the signal should preferably not be put through a tight band-pass filter. However, this leaves the algorithm suspect to noise spikes triggering false detections and also reduces the SNR of the signal which the algorithm has to deal with.

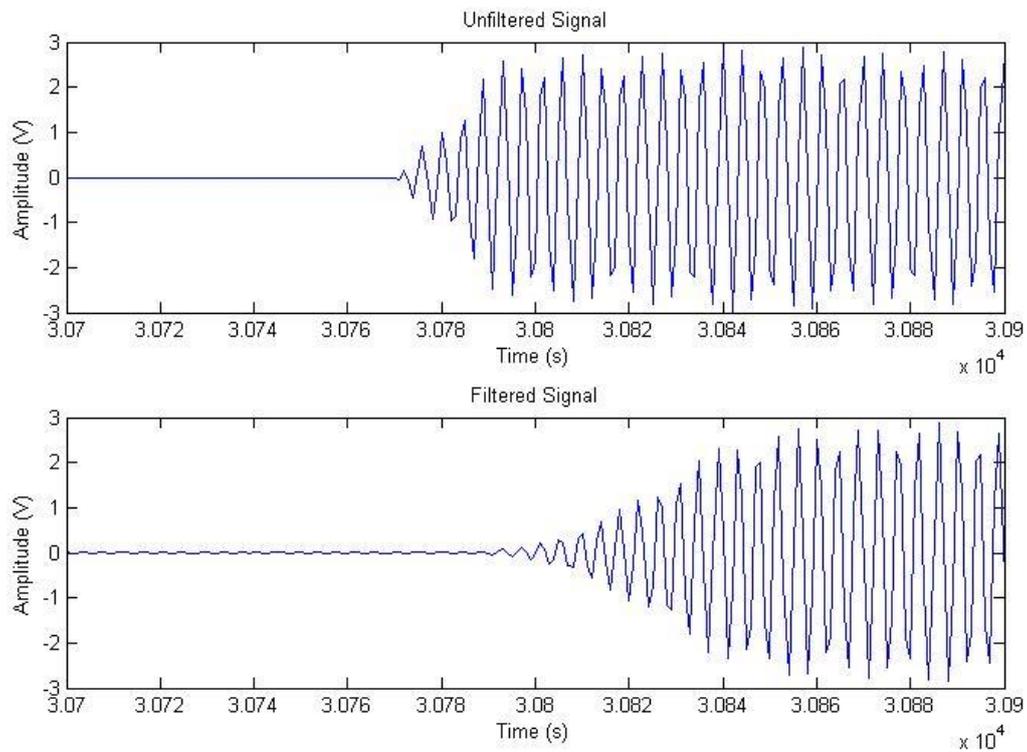


Figure 6: Two graphs of the start of a recorded acoustic pulse showing the effects of the band-pass filter. The top graph shows the original start of the acoustic pulse. In the bottom graph, the signal has been put through a tight band-pass filter, resulting in less noise but also a less defined edge.

9.5.3. Phase Difference Based Techniques

Phase Difference based techniques make use of the phase difference between the signals to deduce their Time Difference of Arrival and hence the Direction of Arrival of the signal. These techniques are applicable not only to the narrowband acoustic ping, but also to wideband signals. While multipath interference poses a problem to these algorithms, these can be easily circumvented by only extracting only the very start of the ping, when multipath interference has not set in, for analysis.

Conventional Beamforming and Capon's Beamformer

Classical beamforming attempts to locate the Direction of Arrival using the array steering vector to “steer” the array in across the all directions one at a time and measuring the output power in the direction being evaluated [4]. The directions which result in the maximum power are then chosen as the DOA estimates. However, the resolving power and statistical performance of conventional beamforming is poor [4].

Capon's beamformer [6] improves on this by attempting to minimise the power contributed by noise and signals from directions other than the one being evaluated. This improves the resolution of the beamformer at the cost of noise suppression capability. However, the resolving power is still limited compared to other methods like MUSIC [4]. .

Deterministic Maximal Likelihood

Deterministic Maximal Likelihood is a popular algorithm due to its optimal performance [8]. The algorithm decomposes the observed data into its components and then estimates the parameters of each signal component iteratively to solve the complicated multiparameter optimization problem. However, this algorithm is computationally intensive.

MUSIC

The MUSIC algorithm [5] invokes the spatial covariance matrix to estimate the DOA of the signal. By projecting the noise eigenvectors calculated from the spatial covariance matrix of the received signal onto the array steering vector, one can measure the orthogonality of these vectors. The direction in which the orthogonality is maximised gives the DOA estimate. This yields a high resolution and is not computationally intensive as it only involves a 2-D search

Additional Notes

Phase difference based techniques perform better in small arrays than Rising Edge based techniques as they are not dependent on a very accurate TOA estimate; only a rough TOA estimate is needed in order to extract the start of the signal for analysis. Conversely, the size of the array must be small in order to avoid spatial aliasing.

Spatial aliasing is a phenomenon that occurs when sensors are placed more than half a wavelength of the carrier wave away from each other. This is because any phase difference could be a result of more than one time difference, each differing from the other by an integer multiple of the period of the signal. Thus, there is no longer a unique solution for the DOA of the signal.

In order to avoid spatial aliasing, the sensors have to be placed within half a wavelength of the expected signals it is to analyze. For a frequency range of up to 45 kHz, this means a maximum separation of about 2cm. As the number of sensors is also limited, this means that the array size is very small. The small array size means that the Near Field Signal model is not applicable when such techniques are applied in the scenario this paper presents. As such, these algorithms can only apply the Far Field Signal model and hence can only provide DOA information with no information on the distance to the source.

10. Algorithm Description

This paper proposes a novel technique of estimating source location of the pinger by supplementing a Rising Edge based technique with a Phase Difference based technique in order to enhance the source location estimate. This would enable us to use both useful features of the acoustic ping to enhance accuracy, robustness and computational simplicity beyond what either technique could achieve alone.

The algorithm uses a MUSIC search based on the Far Field model to first determine Direction of Arrival of the signal to an array of three hydrophones situated in the front of vehicle. A step detector is then used to compute TDOA between the fourth hydrophone located in the rear of the vehicle and the hydrophones at the front (See **Teledyne Reson TC4013 Hydrophones** for more details on the hydrophone arrangement). This is then combined with the DOA information to compute the exact location of the source using the Near Field model.

The detailed steps of the algorithm are described below.

10.1. Time of Arrival Estimation

The algorithm first takes the readings from each hydrophone over a time period of 0.5s, which is the time between each ping. This is the ideal window size because every window should theoretically have at least 1 ping within it, while not being too large that the data is unnecessarily large.

The readings are put through a Hilbert transform to obtain the envelope of the signal. The envelop is then put through a median filter, which helps preserve the sharp rising edge while removing as much noise as possible. The envelope is then analyzed for steps using a sliding

window which computes the variance of the values in that window. The first step detected is taken as the TOA of the pulse.

10.2. MUSIC Search

The MUSIC algorithm from the above section was chosen as the technique used to compute DOA. The MUSIC algorithm was chosen because it is simple enough to implement and computationally not as complex as other algorithms like DML or Adaptive Eigenvalue Decomposition. It also has offers a higher resolution than the Classical and Capon's Beamformer. The higher resolution allows a lower precision to be used in calculating the MUSIC spectrum, hence saving resources in the FPGA implementation (See **Challenges** for more details).

128 samples from the Time of Arrival are taken from the signal for analysis to avoid multipath interference. A Fast Fourier Transform is performed to verify that the sample is of the correct frequency. Once this is done, eigenvalue decomposition is performed to compute the eigenvectors and the noise eigenvectors are extracted.

A 2 Dimensional search is then performed to determine the most likely DOA of the signal. This is done by comparing the orthogonality of the noise eigenvectors with the array steering vector for the DOA being evaluated. The orthogonality of all the available DOAs with the noise eigenvectors then forms the Music Spectrum, as shown in Figure 7. The peak of the MUSIC Spectrum represents the DOA of the signal.

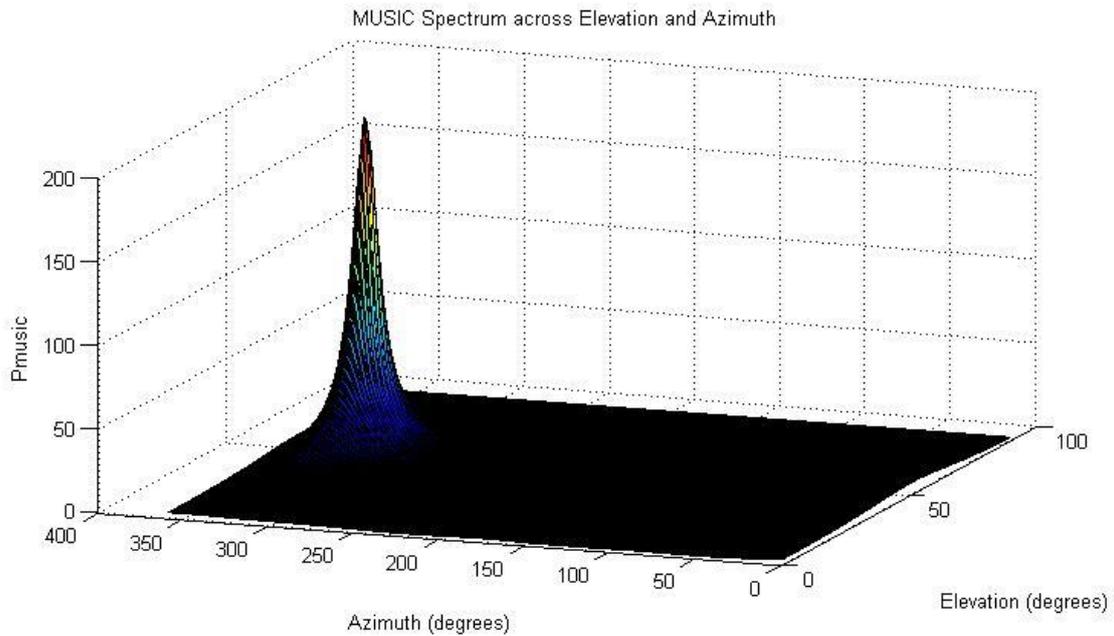


Figure 7: 3D plot showing the MUSIC Spectrum across azimuth and elevation. The peak shows the estimate Direction of Arrival

10.3. TDOA Refinement

The initial TDOA between the front hydrophones and rear hydrophones is further refined through comparison of the phase. A Cross Correlation was performed considering only the time differences within half a period of the initial TDOA to avoid spatial aliasing. This allows us to correct the TDOA estimate using the phase difference of the signal to supplement the Rising Edge detection information.

The quantization error due to the low sampling rate also limits the accuracy. This quantization error is 8 μ s. In order to reduce this, interpolation was performed on the sample before cross correlation to yield a sampling rate of 500 kHz, or 4 times the initial sampling rate. The quantization error is hence reduced by 4 times to only 2 μ s.

10.4. Range search

The TDOA between the front and rear hydrophones is then combined with DOA information to estimate range. This is done by finding the closest TDOA match along the line represented

by the DOA within a distance of 0m to 5m between the front hydrophones and the source. Beyond 5m, the distance to the source is too far to be evaluated accurately.

10.5. MATLAB Implementation

The entire algorithm was first implemented on MATLAB on a desktop computer. This enabled us to test the algorithm with either simple simulated data as a first step check in the initial development stage or recorded data when later in the implementation when it became available. The MATLAB implementation also serves as reference to cross check with the implementation on BBAUV to confirm the accuracy and help with debugging.

11. Implementation

This section covers the implementation of the algorithm described above onto the Bumblebee Autonomous Vehicle (BBAUV), which is built by an undergraduate team of students from National University of Singapore. It will describe the hardware setup on the BBAUV and the software system implemented on the embedded controller of the system.

11.1. Hardware System

This section will describe the hardware setup on BBAUV for the passive acoustic system. The entire hardware system flow is summarized in Figure 8 below. 4 TC4013 hydrophones are used to record the acoustic signals underwater. The signal from the hydrophones is then passed through a custom designed pre-amplifier and band-pass filter board. The amplified and filtered signal is then sampled using the NI 9223 Analog to Digital Converter. The resulting digital signal is then sent to the NI sbRIO-9602 for processing. Each of these components is described in further detail below.

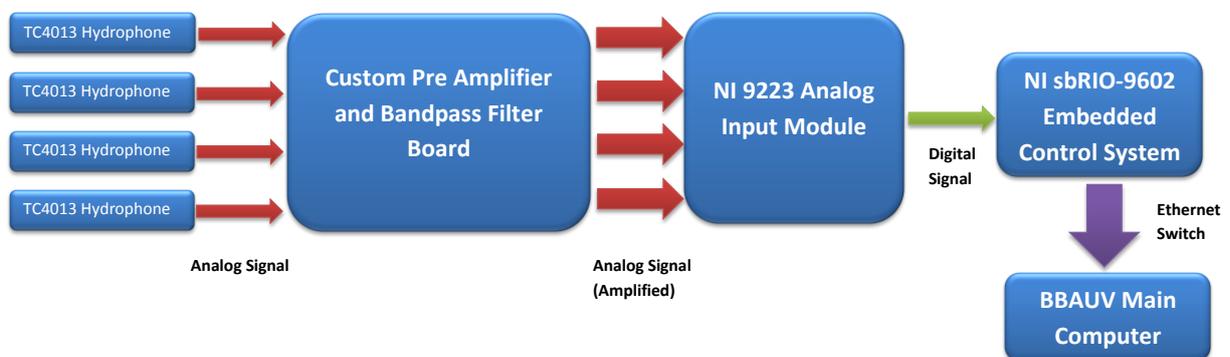


Figure 8: Hardware System Flow Diagram showing how the different hardware components interact with each other.

11.1.1. Teledyne Reson TC4013 Hydrophones

The TC4013 miniature reference hydrophones are uniform omnidirectional hydrophones with a usable frequency range of 1 Hz to 170 kHz. They were chosen for their small size which allows them to be placed in close enough proximity to avoid spatial aliasing.

The first 3 hydrophones are arranged into the 3 corners of a square array with a separation of 1.5cm. The last hydrophone is situated at the rear of the vehicle 48.5 cm away from the furthest forward hydrophones.

11.1.2. Custom Pre Amplifier and Band-pass Filter Board

Due to its small size, the TC4013 hydrophones do not come with a preamplifier to amplify the signal to a suitable level for Analog to Digital Conversion. Hence, a custom designed Printed Circuit Board (PCB) was made for amplifying the signal. DipTrace was used to design the PCB and the results can be seen in Figure 9 below. The schematics for the board were based off schematics from the previous iteration and redesigned to be smaller and more compact.

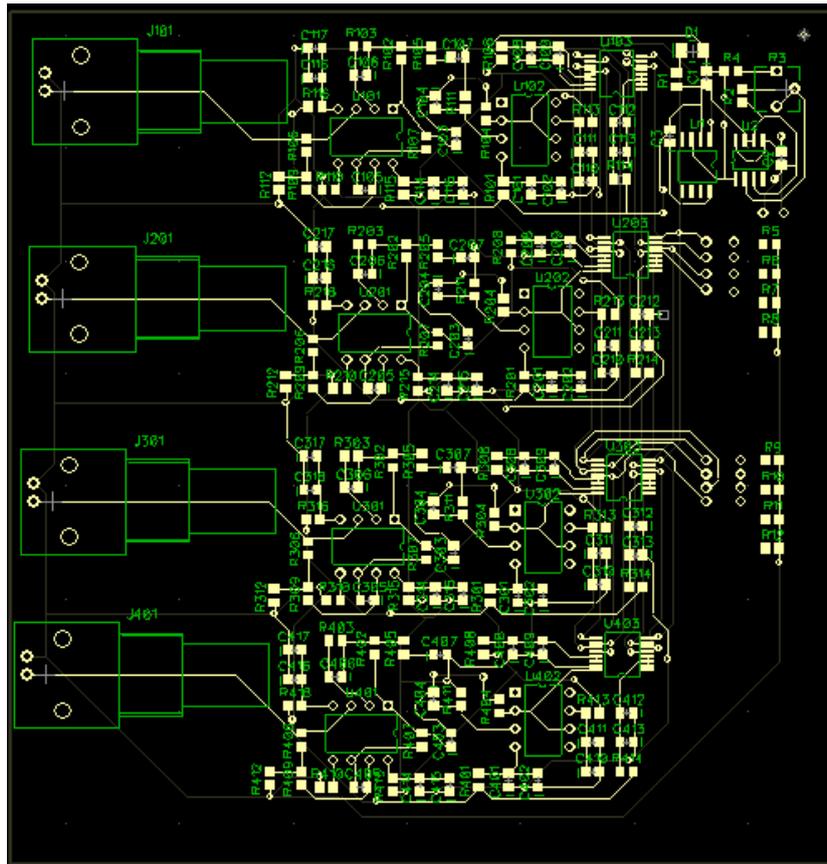


Figure 9: Printed Circuit Board Layout of the Custom Preamplifier and Band-pass Filter Board

The board has 4 channels which connect to the hydrophones using a BNC connector. The input signals from the hydrophones first go through a Low Noise Amplifier with a gain of 20 built from a LT1169 JFET Input Op Amp. Following this, the signal is filtered by a Sallen Key High-pass Filter with a cutoff frequency of 10 kHz built from an AD797 Ultralow Noise Op-Amp. Finally, the signal passes through an LTC1564 Antialiasing Low-pass filter with a passband gain of 16 and a cutoff frequency of 150kHz.

11.1.3. NI 9223 Analog Input Module

The NI 9223 is a 4 Channel Analog to Digital Converter which supports simultaneous sampling at 1MS/s with 16-bit resolution within a measurement range of ± 10 V. Since it is also from National Instruments, the NI 9223 can easily be integrated with the sbRIO-9602 used as the main processor in this system.



Figure 10: NI9223 Analog Input Module from National Instruments

11.1.4. National Instruments sbRIO-9602

The sbRIO-9602 from National Instruments (shown in Figure 11 below) is used as the main processor to implement the MUSIC algorithm. The sbRIO-9602 consists of a 400MHz Freescale Real Time processor and a 2M gate Xilinx Spartan 3 Field Programmable Gate Array (FPGA). It is programmable using the LabVIEW 2012 software suite provided National Instruments.

The sbRIO-9602 processes the incoming signals and handles communications with the main computer of BBAUV via Ethernet, providing the results of the processing as well as communicating any errors which may have occurred.



Figure 11: National Instruments sbRIO-9602

12. Embedded Software System

This section describes the software system implemented on the embedded controller (sbRIO-9602) used to control the entire acoustics hardware system.

12.1.1. Goals and Challenges

The goal of the embedded software system is to implement the algorithm described in the first section onto the hardware setup described above. The software system handles acquiring and processing the signal from the hydrophones, computing the MUSIC algorithm and communicating with the main computer of BBAUV.

The main challenge of the embedded software is to implement the algorithm with a low latency in spite of the limited hardware resources. This is especially challenging as the embedded hardware system is much more resource constrained than typical desktop PCs, with limited disk space, RAM, and CPU bandwidth all posing a restriction on algorithm performance.

12.1.2. Software architecture

The software architecture is divided into 2 major components: the Real-Time component handled by the real time processor on the sbRIO-9602 and the FPGA component which is handled on the Xilinx Spartan 3 FPGA on the sbRIO-9602. This is shown in Figure 12 below. The entire software is programmed with LabVIEW 2012 from National Instruments and is made up of Virtual Instruments (VI), which are the LabVIEW equivalent of functions in other typical programming languages.

The architecture division between the Real Time and FPGA components occurs for several reasons. The Real Time processor on the sbRIO-9602 is not a very powerful processor, making it unsuitable to handle functions which are computationally intensive. On the other

hand, the Spartan 3 FPGA is able to handle computationally intensive operations quickly due to its parallel processing power. However, it is not as flexible as a processor and cannot perform very complex operations. Hence, complex or high level functions are computed in the Real Time Component while the more repetitive and intensive but simpler calculations are computed on the FPGA.

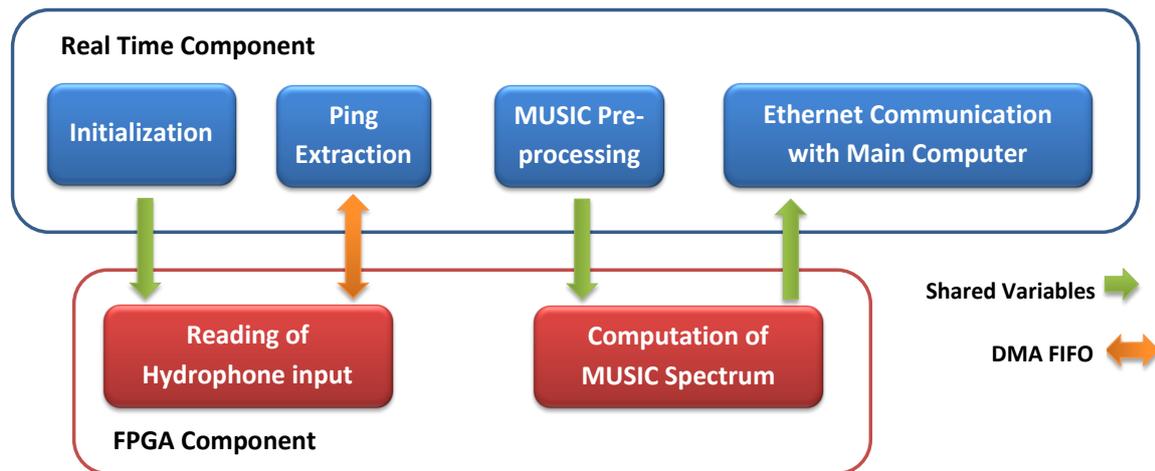


Figure 12: Software Architecture of the embedded software system

The Real Time component handles the high level functions of the algorithm as well as the more complex computations which are not easily implemented on the FPGA. These components are Ping extraction, MUSIC pre-processing and Ethernet communication with the main computer. The functions are elaborated further below.

The FPGA component handles the low level functions which require high speed but do not involve complex operations. These functions are sampling the analog input and computation of the MUSIC spectrum. The FPGA then communicates with the Real Time Processor through the use of a DMA FIFO for large data sets and Shared Variables for flags and small data sets.

12.1.3. Real Time Component

The real time processor runs the main functions of the program. It commands the FPGA and handles commands and communications with the main computer. The top level Virtual Instrument is shown in Figure 13 below, with the necessary displays on the front panel to tell what is happening throughout the entire embedded software system.

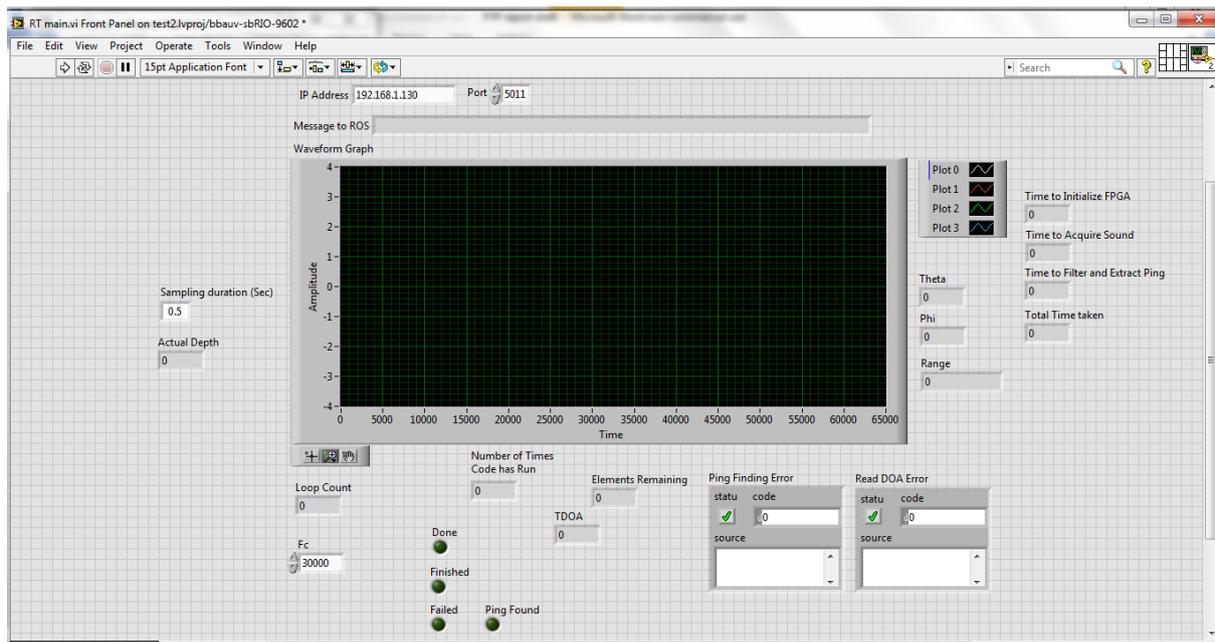


Figure 13: Front Panel of the LabVIEW Real Time Top Level VI

Challenges

The sbRIO-9602 has only 128MB of system memory and runs a simple Real Time Operating System. This limits its capabilities in handling large amounts of data quickly. Hence, computationally intensive functions are avoided on the FPGA and steps are taken reduce the size of data sets that need to be manipulated.

Initialization

The program begins by configuring the FPGA with the bitstream and initializing the DMA FIFO. Following this, it triggers the Start Reading flag to indicate to the FPGA to begin reading data from the hydrophones. Following this, it will sleep until the Done flag is raised,

upon which the data from the DMA FIFO is read. The data is then unpacked into the 4 different channels (See FPGA Component for more details on how data from 4 channels is packed into 1 FIFO).

In order to improve the performance of the algorithm, the sampling frequency was lowered to decrease the number of data points in each set. An evaluation was conducted between 300 kHz and 100 kHz to decide the optimal sampling frequency. Beyond 300 kHz, it was found that the processor could not handle so much data. The Nyquist frequency of the system was 90 kHz (given the maximum pinger frequency of 45 kHz), hence the lowest sampling frequency evaluated was 100 kHz. It was found that below a sampling frequency of 125 kHz, the decrease in processing time is not significant, as shown in Figure 14 below. Hence, 125 kHz was set as the sampling frequency.

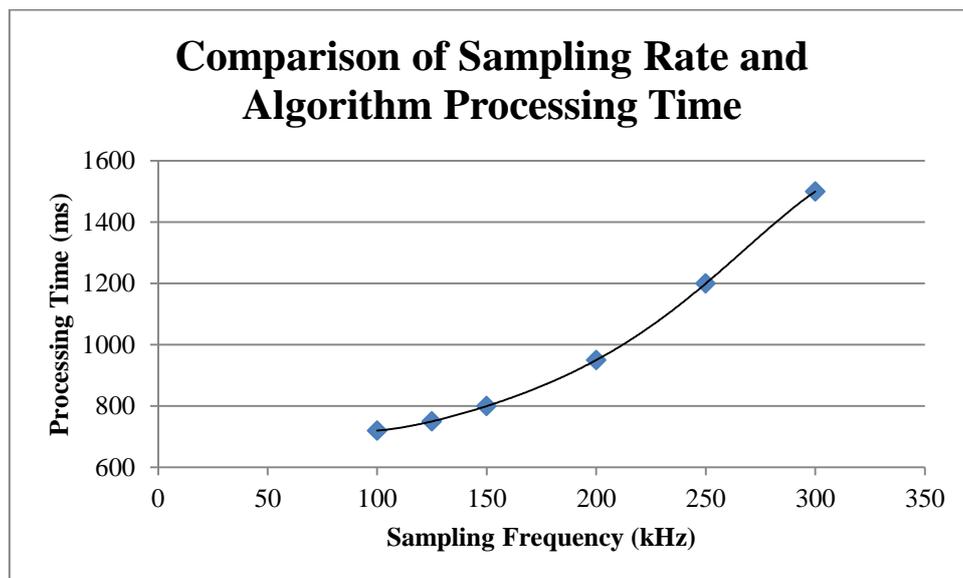


Figure 14: Comparison of Sampling rate and time taken for the algorithm to run. In the region of 100 kHz to 125 kHz, the processing time does not reduce by much, especially in comparison with the 200 kHz to 250 kHz range.

Ping Extraction

The data from each of the 4 channels is then processed in the Ping Extraction function to determine if a ping is present and to extract the ping out from the data. The data is first passed

through a 10th order Elliptic band-pass filter to minimize noise. A dynamic threshold is then used to determine if the data requires further examination; if a ratio (set at 1:5) of the maximum point of the data is larger than a noise threshold (set at 0.01V), it is assumed that a ping could possibly be in the data and the point where the data crosses the noise threshold is used as the start of the ping.

128 samples after the start are taken from the data in order to collect the data uncorrupted by multipath interference. This data is then put through a Fast Fourier Transform (FFT) and the peak of the FFT is compared to the desired frequency. If they match, the complex FFT point at the peak is extracted and passed to the MUSIC Preprocessing Function.

The Time Difference of Arrival of the ping at the 1st and 4th hydrophones is also calculated here. The unfiltered raw data from the first and fourth hydrophones are first put through a Hilbert transform to detect their envelope. The envelope is then put through a median filter to remove the noise and the Time of Arrival is found using an edge detector which computes the first peak in variance across a sliding window. The Time of Arrival is then fine-tuned further by doing a Cross Correlation using only time differences within half a period of the initial Time of Arrival estimate to avoid spatial aliasing.

MUSIC Preprocessing

The MUSIC Preprocessing function computes the values needed for the computation of the MUSIC spectrum. The covariance matrix of the 4 complex FFT points from the Ping Extraction function is calculated first. Eigenvalue decomposition is then done on the covariance matrix to extract the noise eigenvectors. The noise eigenvectors are then passed to the FPGA to compute the MUSIC spectrum.

These complex mathematical operations are not easily implementable on the FPGA and hence have to be performed on the processor. Due to the small number of values involved,

they take only about 20 milliseconds to perform and hence do not significantly affect the latency of the system.

Ethernet Communication with Main Computer

The sbRIO-9602 communicates with the main computer of BBAUV through an Ethernet connection. Messages are sent to the main computer using User Datagram Protocol (UDP). UDP was chosen as it allows messages to be sent easily and quickly without any handshaking dialogues, thus minimizing complexity and time of the communication.

Besides communicating the Direction of Arrival and estimated distance to the source, this function must also communicate any errors which may have occurred on the sbRIO-9602 so that the main computer can take any appropriate action necessary.

12.1.4. FPGA Component

The FPGA implements two functions which supplement the Real Time program: the Reading of the hydrophone input and the computation of the MUSIC spectrum.

Challenges

The FPGA is a powerful resource because it allows a circuit to be tailored to the algorithm at hand. Therefore, all parts of the algorithm which can be optimized for parallel processing, in contrast to a standard processor which executes one instruction at a time. However, the FPGA is difficult to program and not as flexible as the processor. This limits the FPGA's abilities to do complex calculations.

Firstly, the FPGA supports much less functions compared to a processor. Only simple mathematical functions like Fast Fourier Transform and Butterworth Filtering are available on the LabVIEW FPGA module, in contrast to the LabVIEW 2012 used to program the processor. This can be seen from the difference in the size of the functions palette in LabVIEW FPGA and LabVIEW illustrated in Figure 15. This makes it difficult to do advanced processing on the FPGA.

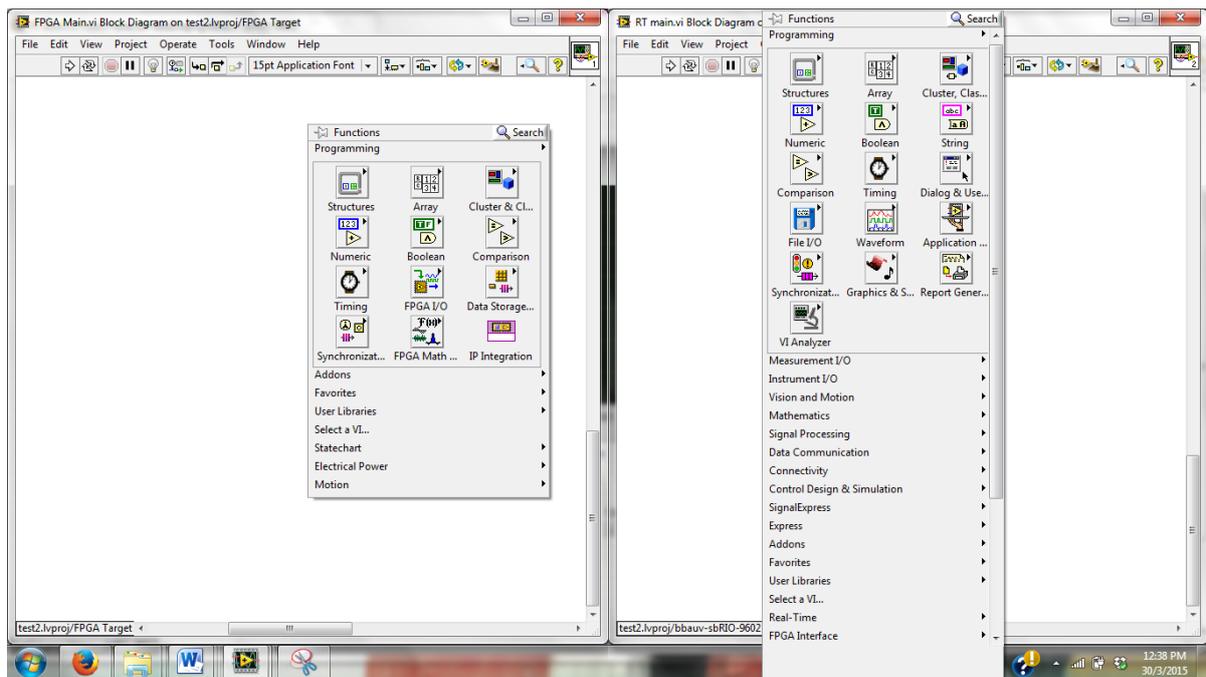


Figure 15: LabVIEW FPGA Functions Palette on the left is much smaller than the full LabVIEW Functions Palette on the right. The difference in size indicates the difference in number of functions available on both platforms

Secondly, floating point numbers and complex numbers are also not supported in an FPGA. Only Fixed Point numbers are available, greatly increasing the amount of effort required to implement mathematical operations. Therefore, operations like multiplying large numbers, which present no problem on a processor, become very computationally expensive when dealing with FPGA programming.

Lastly, compilation of a single FPGA program can require between 30 minutes to an hour. This makes iterating and debugging a program tedious.

As a result of these limitations, only operations which are simple but require high speed or are computationally intensive on a processor are implemented on the FPGA. These functions are detailed below.

Reading of Hydrophone input

The reading of hydrophone input is contained. When the Read Flag is set to true by the real time processor, the FPGA initiates a loop to read 0.5 seconds of data. The data is then interleaved into an array in the manner shown in Table 1 below and placed onto the DMA FIFO for transfer to the processor. Once the reading is complete, the Done flag is raised and the loop enters a sleep state waiting for the Read Flag to be set to true again.

Array Index	Element
0	Hydrophone 1
1	Hydrophone 2
2	Hydrophone 3
3	Hydrophone 4
4	Hydrophone 1
5	Hydrophone 2
6	Hydrophone 3

Table 1: Interleaving of 4 Analog Channels into a single DMA FIFO. This allows 4 channels worth of data to be sent through the only channel available between FPGA and Processor

This is done on the FPGA because the real time processor cannot handle sampling of analog inputs at such a high sampling rate, while at the same time servicing other processes. The reading is therefore executed on the FPGA.

Computation of MUSIC Spectrum

The MUSIC spectrum involves a 2-Dimensional search in the azimuth and elevation dimensions (collectively referred to as direction) in order to evaluate which direction is the source most likely to be in.

Each evaluation involves calculating the steering vector for that particular direction and multiplying it with the complex noise eigenvectors passed from the Music Preprocessing

function. The result is then inverted to find the peak of the spectrum, which gives the estimated direction of the source.

The 2-Dimensional search involves 32400 iterations. This puts a load on the processor since each of these are computed one at a time on the processor. The FPGA allows us to tailor a circuit to compute these iterations simultaneously, greatly reducing the time needed to complete the search.

13. Results

The implementation was extensively tested in the swimming pool at depths of 1.8 m to 5 m, with the pinger located up to 20m away. The estimate of the Direction of Arrival was accurate and robust. The DOA fluctuates by ± 2.5 degrees in azimuth and ± 5 degrees in elevation when the vehicle is stationary. There are also very few false triggers of the algorithm even at an SNR of 5dB, proving the robustness of the algorithm.

A low latency is also achieved with the implementation of the algorithm on BBAUV. The time it takes from the end of the recording of the signal to the delivery of source location information is about 120ms. This is in contrast to the MATLAB implementation on a desktop system. A naïve implementation takes up to 7 seconds, while an implementation which uses a hash table reduces this to about 4 seconds. This low latency allows the vehicle to move without long pauses.

The estimation of distance to the source was found to be accurate within a range of up to 2 meters. The refined TDOA estimate is more stable, fluctuating only between 1-2 μ s, than the initial TDOA estimate which usually fluctuates by about 5-10 μ s. Beyond 2m, the range estimate appeared to fluctuate by large amounts, making it unreliable for practical usage.

14. Conclusion

This paper proposed a novel algorithm using the rising edge in combination with the phase information of an acoustic ping to estimate its source location. The algorithm was validated through implementation as a real time system on the BumbleBee Autonomous Underwater Vehicle.

The resulting algorithm was found to meet the specifications of the system stated in the beginning of the project, in particular the requirements for. The estimation of distance, although outperforming what could be done with just a Rising Edge based algorithm, can still be improved upon further.

Future work can be done to further evaluate the use of other methods to further enhance the estimation of distance. A larger number of sensors would allow for more robust estimation with less fluctuation due to the increased redundancy in information, possibly allowing for reliable distance estimation at larger distances. A different sensor array configuration could

This paper proposed a novel algorithm using the rising edge in combination with the phase information of an acoustic ping to estimate its source location. The algorithm was validated through implementation as a real time system on the BumbleBee Autonomous Underwater Vehicle.

The resulting algorithm was found to meet the specifications of the system stated in the beginning of the project, in particular the requirements for. The estimation of distance, although outperforming what could be done with just a Rising Edge based algorithm, can still be improved upon further.

Future work can be done to further evaluate the use of other methods to further enhance the estimation of distance. A larger number of sensors would allow for more robust estimation with less fluctuation due to the increased redundancy in information, possibly allowing for

reliable distance estimation at larger distances. A different sensor array configuration could also possibly be implemented to improve the distance estimate.

15. Works Cited

- [1] Numerical Recipes, Numerical Recipes, The Art of Scientific Computing, Third Edition, Cambridge: Cambridge University Press, 2007.
- [2] H. L. V. Trees, "Detection, Estimation, and Modulation Theory, Part," Wiley, New York, 1971.
- [3] W. G. MaMullen, B. A. Delaughe and J. S. Bird, "A simple rising-edge detector for time-of-arrival estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, no. 4, pp. 823-827, 1996.
- [4] M. V. a. H. Krim, "Two Decades of Array Signal Processing Research," *IEEE Signal Processing Magazine*, pp. pp. 67-94, 1996.
- [5] J. Capon, "High-Resolution Frequency-Wavenumber Spectrum Analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408 - 1418 , 1969.
- [6] N. Kabaoğlu, H. A. Çırpan, E. Çekli and S. Paker, "Deterministic Maximum Likelihood Approach for 3-D Near Field Source Localization," *International Journal of Electronics and Communications*, vol. 57, no. No. 5, pp. 345-350, 2003.
- [7] R. Schmidt, "A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation," PhD Thesis, Stanford University, Stanford,CA, 1981.
- [8] B. A. Delanghe, "Acoustic pulse time of arrival estimate by rising edge detection," Simon

Fraser University (Canada), Ann Arbor, 1994.

[9] J. Benesty, J. Chen and Y. Huang, *Microphone Array Signal Processing*, Berlin Heidelberg: Springer, 2008.