2011

Efficient Time of Arrival Calculation for Acoustic Source Localization Using Wireless Sensor Networks

Prashanth G. Reddy
Cleveland State University

EFFICIENT TIME OF ARRIVAL CALCULATION FOR ACOUSTIC SOURCE LOCALIZATION USING WIRELESS SENSOR NETWORKS

PRASHANTH G. REDDY

Bachelor of Science in Computer Science

Georgia State University

December 2006

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

December 2011

This thesis has been approved for the

Department of **ELECTRICAL AND COMPUTER ENGINEERING**

and the College of Graduate Studies by

Thesis Committee Chairperson, Dr. Nigamanth Sridhar

Department/Date

Dr. Murad Hizlan

Department/Date

Dr. Wenbing Zhao

Department/Date

To my parents...

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Dr. Nigamanth Sridhar for all his support to make this thesis happen and for not making my graduate life miserable. I've learnt so much thanks to the freedom he gave his students in allowing them to explore their interests. Also, I enjoyed all the intellectually stimulating discussions and brainstorming sessions I've had with him for the past three years.

I would like to thank everyone at Department of Electrical Engineering for all their support they'd given me. I'd be completely lost if it weren't for the help from the department's staff, especially Adrienne Fox.

I can't thank my lab partners and friends Greg Glazer, Gaurav Konchady, Sriram Sanka and Sharmila Kollipara enough, for making the past three years as enjoyable as it was. I cannot forget the important life lessons I've learned from them. A particular thanks to Sriram Sanka for the numerous intellectual discussions we had together for the past few years. It was an interesting discussion with him that lead to the ideas behind this thesis, among many others.

Most importantly, I owe everything I am to my parents. Their unwavering support and tolerance reminds me everyday how lucky I am in exploring my dreams, unfettered. My brother has been and will continue to be a role model for me in many good thing in life that only a few have control over.

EFFICIENT TIME OF ARRIVAL CALCULATION FOR ACOUSTIC SOURCE LOCALIZATION USING WIRELESS SENSOR NETWORKS

PRASHANTH G. REDDY

ABSTRACT

Acoustic source localization is a very useful tool in surveillance and tracking applications. Potential exists for ubiquitous presence of acoustic source localization systems. However, due to several significant challenges they are currently limited in their applications. Wireless Sensor Networks (WSN) offer a feasible solution that can allow for large, ever present acoustic localization systems. Some fundamental challenges remain.

This thesis presents some ideas for helping solve the challenging problems faced by networked acoustic localization systems. We make use of a low-power WSN designed specifically for distributed acoustic source localization. Our ideas are based on three important observations. First, sounds emanating from a source will be free of reflections at the beginning of the sound. We make use of this observation by selectively processing only the initial parts of a sound to be localized. Second, the significant features of a sound are more robust to various interference sources. We perform key feature recognition such as the locations of significant zero crossings and local peaks. Third, these features which are compressed descriptors, can also be used for distributed pattern matching. For this we perform basic pattern analysis by comparing sampled signals from various nodes in order to determine better Time Of Arrivals (TOA). Our implementation tests these ideas in a predictable test environment. A complete system for general sounds is left for future work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

Introduction

Acoustic source localization is a potentially very useful technique for surveillance. It is also a challenging task even for a resource rich system due to its heavy reliance on signal processing. Acoustic Source Localization (ASL) is the act of localizing an acoustic source in space with respect to a known coordinate system [6]. ASL is traditionally performed on a resource rich system due to the complexity of processing multiple acoustic signals simultaneously. Due to ASL's requirement of distributed sensing, it is not hard to envision an acoustic Wireless Sensor Network (WSN). The challenges however are daunting especially if any significant accuracy is to be maintained [10]. In this thesis we present methods to perform acoustic source localization in an efficient manner using a wireless sensor network.

Localization has become an ubiquitous accessory today with the advent of Global Positioning System (GPS). Though GPS involves the act of localizing oneself with respect to fixed satellites, the underlying principle is similar to other forms of source localization [25]. Most accurate localization systems rely on electromagnetic signals due to their narrow band nature. Narrow band signals are simpler to localize because only a single or a select group of frequencies need to be used [19]. Acoustic phenomena however are inherently wide band signals and as such are much more complex to deal with.

Acoustic localization has long been of interest to biologists studying how animals acoustically perceive their environment at least since the 1940's [15]. Acoustic localization for surveillance has been of research interest for several decades. More recently numerous acoustic localization systems have been developed for the purpose of gunshot localization [5]. It would be of great importance if gunshots alone could be localized in an environment for military and law enforcement needs. Systems have even been developed for biologists studying animal behaviors based on their vocalizations in the wild [32]. Biologists often seek to monitor animal vocalizations and in particular their migratory behavior. All these however fall in the large category of surveillance be it for the purpose of finding an infraction of peace or tracking animals in the wild.

It makes sense to have a large distributed sensing system for large area coverage. With this in mind most recent acoustic localization and surveillance systems are wirelessly networked acoustic sensors. It is possible to envision a very large city instrumented with acoustic sensors to localize and track various phenomena encountered. Recently mobile phones have been used to track the sources of noise pollution in an urban environment [26]. Wireless acoustic sensor networks can be a powerful tool for the greater good of civilized societies primarily in the areas of law enforcement, sociological and environmental surveillance. Such a task would have tremendous social and technical complications. Dealing with a small part of the technical complexities will be the concern of this thesis.

1.1 Problem

The biggest challenge for acoustic source localization in wireless sensor networks (WSN) is the computational and energy constraints characteristic of WSNs. Desktop computers, for example, have powerful multi-core processors capable of complex signal processing tasks and gigabytes of RAM. Such a network node, should one be used, would have minimal resource constraints for performing networked ASL. Wireless sensor network nodes, on the other hand, are designed to be cheap, hardly noticeable, hardly significant sensing devices. The expression "strength in numbers" readily comes to mind when describing wireless sensor networks. This paradigm places computation and energy constraints that desktop class systems do not face, on each node of a WSN. Large wireless sensor networks typically consist of nodes with highly limited resources in the range of 10 Kilobytes of RAM and 1 - 10 MHz range fixed point processors [21]. For the purpose of signal-processing-intensive tasks such as ASL it is often the case that customized nodes are developed which are much less restrictive specifically for ASL. This application specific customization however comes at the cost of greater resource usage and more significant node cost. This inevitably places a greater restriction on the scale of deployments.

Acoustic signals are almost always wide band signals composed of ranges of frequencies. This is significantly different from the narrow band signals commonly used in electromagnetic-based localization systems such as GPS and RADAR. This makes acoustic signals computationally complicated to deal with as signals from various source nodes need to be compared with each other. The standard approach for this comparison is by using cross-correlation on two separate signals. This is however problematic for several reasons. First, large sections of audio signals need to be transmitted to a location where they will be correlated. Data transfer in large quantities is perhaps the Achilles heel to WSN energy consumption minimization. Radio transmissions are usually the most energy consuming tasks and as such are avoided whenever possible. Second, cross-correlation is a computationally intensive process for a resource constrained system such as the typical low power wireless sensor nodes. This problem can rival the energy costs of radio transmissions in WSNs. Thirdly, cross-correlation relies on a large section of acquired signal. This is fine in large open environments with no possibilities of reflections. However, reflections in the form of echoes and reverberations make cross-correlation methods somewhat inaccurate. This, in spite of the development of Generalized Cross Correlation with Phase Transform (GCC-PHAT) [3] method that is much better at handling reverberations. These reasons make the use of cross-correlation techniques infeasible in large scale WSN deployments for ASL.

Unlike cross-correlation based techniques, Time Of Arrival (TOA) based methods are much more efficient especially for WSNs. Most gunshot localization systems based on WSNs rely on time of arrival using significant changes in signal amplitudes. While time of arrival methods in gunshot localization is highly predictable, it is seldom used for localizing general sounds. Threshold based methods, have seen great use in WSN based ASL for highly impulsive sounds. However the problem here is that impulsive sounds only comprise a small portion of sounds naturally occurring in the environment.

A large problem with signal threshold based methods is the wide variations of sounds encountered by various nodes in the network. Acoustic signals being wideband in nature are prone to large variations in the sampled signal from node to node. What one node hears, while being similar to what its neighboring node hears, is different enough that when threshold detection is used, large variations in time of arrival are detected, making localization less accurate. Therefore methods are required to consistently match the signals across nodes with accurate time differences maintained, all while keeping computation and communication to a minimum. This thesis primarily aims to improve upon time of arrival based methods such as signal thresholding such that it is applicable to a wider variety of sounds.

1.2 The Thesis

Acoustic source localization despite its demanding requirements can be simplified to be performed in an efficient manner. In addition, unlike most current research work for localizing general sound sources, energy and cost-efficient devices can be used. This thesis describes some techniques that could be used to allow low cost sound source localization to be to performed, possibly even in real-time.

1.3 Solution Approach

Signal parameters that uniquely represent a short time signal provides a snapshot of the signal for comparison between nodes. While correlation methods can provide accurate time differences necessary for localization, they are too inefficient. In addition they fail when too many reflective paths are present. This thesis aims to solve these problems in three significant steps.

- Minimize interference from reflection by considering initial parts of a signal based on node distances.
- Use signal zero-crossing and peak detection instead of impulsive changes in amplitude, along with signal tracking.
- Networked pattern matching from tracked signals to allow for more reliable time differences.

To begin with, even though each node is collecting data continuously, only the initial part of a given sound occurrence is taken into consideration. The length of the considered part is dependent on the particular pair of nodes in consideration. Therefore the length of interest is dynamic. By choosing only the initial part of the sound, interference caused by reflection can be greatly reduced if not avoided altogether. Reflections occur when energy in the form of sound pressure waves bounces off of surfaces and follows a path different from that of the sound's source. These reflections when captured by a node in addition to the true source's sound already being experienced by that node, will cause significant distortions in the final captured signal. By dynamically changing the length of capture for a given two nodes, this data corrupted by reflections can be avoided in the next stages of localization.

Next, by looking at only the most significant part of the signal, the first few instants of the event, transmission of the complete signal between nodes can be avoided. Thresholding by itself is highly efficient because of its very minimal signal processing requirement. However, thresholding alone is prone to error between nodes due to the significant changes present from node to node. This thesis aims to minimize this problem by using peak and zero-crossing detection instead. This alone is however not sufficient. Because the signal strength of sound reduces as distance from source increases, two different nodes with different distances from the source will pick up different amplitudes. However, the signal pattern would be similar assuming other interferences such as secondary sources or reflections are not significant. By employing peak detection and signal pattern tracking much more accurate TOA information can be acquired.

The signal is then tracked individually on each node. Tracking the signal involves making note of parameters of the signal segment such as locations of maxima and minima along with their amplitude ratios. Change in frequencies by using zero-crossing rates are also possible features. Essentially, each node creates a pattern with a predictable path that represents that section of signal on each node. This is essentially a feature descriptor of that signal which greatly compresses the information for efficient transfer between nodes.

The final stage that this thesis introduces is the networked pattern matching stage. Here the feature set of the signal of interest provided by each node is transferred between node pairs for comparison. The node pairs are selected according to the standard requirements placed by localization techniques. In this pattern matching stage, the feature sets are to be compared by pattern matching techniques. Because the feature sets are highly compressed descriptors of the original signal, the processing required for this correlation is minimal. In addition, because of the distributed nature of node pair selection, the transmission of those feature sets and processing are also distributed. The resulting output is the times of arrival information of each node which is used in the localization stages. Localization will be discussed as a final part of the theory as this thesis does not contribute to the established methods of localization, rather only to the efficient determination of the TOAs vital to localization.

1.4 Contributions

- This thesis makes the following contributions: Presents novel techniques for efficient wireless sensor network based acoustic source localization.
- Efficient mechanisms of signal processing and matching necessary for such localization.
- Implement and evaluate our theory by testing core methodologies.

1.5 Organization of Thesis

This thesis is organized as follows. Chapter 2 provides necessary background information on wireless sensor networks relavent to acoustic source localization. Chapter 3 provides details on our architecture for an efficient networked acoustic localization system. Chapter 4 describes in detail the implemented theory along with the necessary work used to test validate our idea. Chapter 5 details the evaluation methodology along with results and problems encountered. Chapter 6 provides a review of similar work. And finally Chapter 7 concludes this thesis along with future work.

CHAPTER II

Background

2.1 Wireless Sensor Networks

The field of Wireless sensor networks (WSN) is relatively new. Traditionally, problems that required sensing of a system were limited to wired sensors connected to a central computer. This model is widespread and has been critical to the electronic age. Examples of this are, sensors in an automobile monitoring and controlling the engine, security systems relaying video and audio data to a monitoring station, among numerous others. However these applications though relying on sensors, and typically are not considered as sensor networks. Sensor networks differ in that they are networked computers with each computer having its own set of sensors accessible directly only by its computer. Examples of wired sensor networks include some area surveillance networks, computer networks in general communicating with users among others. These networks however are largely wired and far too general in their use. The new field of wireless sensor networks deals largely with gathering data about an environment and relaying this information to a central location for further analysis. Adding wireless communication capabilities to small embedded computers with sensors allow them to be deployed in places that larger systems cannot be placed. The variety and range of applications that embedded system based networked wireless and wired systems are being applied to is vast and growing [7].

2.2 Wireless Sensor Network Hardware

Sensor networks gain a great degree of freedom when they become wireless allowing them to be deployed in places previously infeasible. However this added freedom also places a serious restriction because these wireless devices need to be powered by a portable energy source, usually a battery. WSNs are often designed to last for extended periods which places further restriction on the energy source. Due to these severe restrictions, WSN hardware is designed to be be very efficient and as a result limited in capabilities. Some of the hardware that make up WSN hardware are the radio for wireless communication, sensors for sensing the environment and an embedded processor for collecting data, performing basic processing and using the radio module to transmit the data [20].

2.2.1 Node Processors

Each node that collectively make up the network has one or more low power embedded processor for performing various operations relevant to the networks function. The most popular of the embedded processor have been limited to 8-bit or 16-bit processing, although some of the newer processors have 32-bit processing capabilities [14]. The processors are usually limited to fixed point processing and limited RAM sizes due to the severe energy restriction. One popular processor is the Texas Instruments MSP 430 microcontroller. The MSP 430 is a 16-bit processor that is very energy efficient for performing mixed tasks such as managing various sensors and performing basic processing on the sensed data when needed. Because of the popularity of this processor our thesis makes use of a network of nodes employing the MSP 430 microcontroller. There have been applications that require a low power processor for efficient resource management and one or more powerful processor for serious computations of the sensed data [24]. Many applications are relying on signal processing capabilities, for example, to perform data compression. These networks employ more energy expensive digital signal processors (DSP) such as the Intel PXA271. For our thesis we limit our usage to the popular TI MSP 430.

2.2.2 WSN Radio Modules

Wireless sensor networks require a wireless communication medium for all their communication needs. Most WSN communicate by the use of radio due to their feasibility in most applications. However usage of the radio is minimized due to their high energy usage. So far two major WSN radio frequencies have been used, 900 MHz and 2.4 GHz in WSNs. The current most popular radio the TI ChipCon CC2400 series of radio chips use a 2.4 GHz communication frequency. One of the main goals of our thesis is to minimize the radio usage and thus greatly extending network life.

2.3 Sensor Network Software

2.3.1 TinyOS

Because of the complexities involved data collection in WSNs, software systems specific to sensor networks are often used. One of the most popular operating system is TinyOS [30]. TinyOS has many software components that are important to accomplish various critical functions that make a WSN. One important function of software systems like TinyOS is to allow convenient access to the underlying hardware including ADCs, radio modules, storage and the processing mechanisms. Another critical component in WSN operating systems is the need for MAC protocols which will be discussed in Section 2.3.2. There are several other operating systems that are used primarily for wireless sensor network research work and applications.

2.3.2 MAC Protocols

The communication medium by radio is the electromagnetic medium where all nodes use a highly limited range of frequencies to communicate. Most communication hardware in WSN allow only one device to transmit at a time in the medium. However more than one node can listen to the medium at a given time as the listening process does not interfere with any other transmission process in the medium. Because of these restrictions a software communication protocol is needed to allows for reliable transmission and reception of data between nodes in the network. This is the job of the Media Access Controller (MAC).

There are several basic types of MAC protocols. Time Division Multiple Access (TDMA) is one type that requires various devices in the network to take turns transmitting data. TDMA is often used in WSN although not the most popular due to their high energy consumption which in turn is because of the requirement of precise time synchronization between nodes. Frequency Division Multiple Access (FDMA) and Code Division Multiple Access (CDMA) are other communication protocols that are seldom used in WSN due to their higher hardware and resource requirements. Carrier Sense Multiple Access (CSMA) protocols that allow for unrestricted usage of the communication medium as needed by a device in the network. CSMA protocols are such that a node first samples the communication medium before transmission. A node transmits only if the medium is unused. When not transmitting the nodes can be listening to the medium for other transmission if necessary. Because of CSMA's freedom of communication at the time of need, it is generally very efficient for networks that require communication at unpredictable times. Because of the unpredictable nature of the environment being sensed by the WSN, data transmission can occur at any time. These characteristics of WSNs and the simplicity of the CSMA protocols make them the most popular type of MAC protocols in WSNs. However WSN have many variations of CSMA MAC protocols depending on the application. Currently TinyOS 2.x makes use of the BMAC protocol by default. BMAC is designed to be an energy efficient protocol by allowing a flexible interface and adaptive parameters [9]. In this thesis work we use this default protocol adequately although collisions are a problem at certain instances.

2.3.3 Time Synchronization

Time synchronization is the act of synchronizing various nodes in the network relative to a reference clock common to all nodes. Time synchronization is often used in WSN depending on the application. TDMA protocols for example require relatively precise time synchronization. Many application require time synchronization not just for the MAC protocols but also for properly sensing the environment. In general the more precise the time synchronization requirement the greater the demand on the network hardware and resources, especially energy. Tight time synchronization is therefore avoided if possible. The need for time synchronization arises from a few basic facts. First, any two given nodes have oscillators that differ slightly for each other. Though the more precise oscillators based on mechanical oscillations such as MEMs, tuning fork or crystal oscillators offer relatively small difference on a large time scale, when small differences in timing matter, they fail. This is due to slight variations in the manufacturing process and material imperfections that cause two oscillators to be slightly different from each other. Second significant cause for further variations in oscillator frequency differences is due to variations in temperature at each node. Two nodes placed in an environment where even slight differences in temperatures can cause sufficient variation in oscillator frequencies. For example, even if two particular nodes are located in an outdoor setting such as a park, one node might be exposed to sunlight more than the other therefore resulting in a slight temperature difference. WSNs nodes placed in an indoor setting will likely see even greater temperature variations. Most sensor networks are placed in environments that will result in some clock drifts between nodes even if oscillators were finely tuned and selected before installation.

Besides TDMA communication protocols, WSN applications that perform acoustic event localization are some of the most demanding application in terms of the accuracy of time synchronization. Several types of time synchronization protocols have been developed for applications that require tight synchronization. Reference Broadcasting Schemes (RBS), for example rely on reference nodes that broadcast a beacon signal which each node in the network referee to [8]. This timing beacon does not contain any time stamp. The nodes use the time of arrival with respect to its own clock and make phase difference measurements relative to other nodes for synchronization. RBS is therefore a receiver-receiver synchronization protocol. Time Synchronization Protocol for Sensors Networked (TPSN) is a synchronization protocol based on sender-receiver synchronization unlike RBS [27]. TPSN has two steps where the first stage is the level discovery phase where a tree structure is create such that root nodes transmit a timing beacon to be received by its child nodes. The next step is the synchronization stage where each root node starting with the root node at level zero transmit a timing beacon used only for synchronizing its immediate children nodes. This process continues until all nodes are synchronized. The inherent tree structure allowed by TPSN allows for scalable multi-hop networks unlike RBS. Flooding Time Synchronization Protocol (FTSP) is another synchronization protocol popular in WSN requiring accurate synchronization. FTSP is very similar to TPSN in that it is also designed for multi-hop networks, uses a hierarchy although not a tree topology, and synchronization is performed sender to receiver [17]. FTSP however uses linear regressing for clock drift compensation and allows for dynamic shifting of the root node. These allow for better synchronization accuracy with FTSP than TPSN's. The high time synchronization accuracy requirement placed by acoustic localization WSNs require the use of a protocol such as FTSP. FTSP is also used in Vanderbilt's sniper localization systems. In this thesis work we make use of a wired synchronization for synchronization with accuracy well beyond that provided by wireless synchronization protocols. We make use of this wired protocol to reduce contribution of error due to a time-synchronization protocol to the system. Section 4.3 provides details of our wired time-synchronization protocol.

2.4 Wireless Sensor Network Applications

Over the past decade wireless sensor networks have started to see tremendous growth in the variety of applications they are used in. Though largely in the research stage, WSNs have seen uses in applications ranging from surveillance to human body sensor networks. As discussed before WSN allow for cheaply monitoring a large area unlike a wired system. Vanderbilt university has bone significant work in gunshot and sniper localization [4]. UC Berkeley has used sensor networks for tracking animals based on their calls [13]. In our previous work, we performed efficient signal processing for speaker recognition in a sensor network with our LAKON sensor node architecture [24]. Researchers at UT Dallas have done much work on body motion recognition which is one of the fast growing areas in WSNs relating to monitoring wearer's health. Even smart phones used for collecting and enhancing environmental data has gathered large interest for surveillance and user experience enhancement [16]. WSNs are bound to become a ubiquitous part of our daily life in our near future.

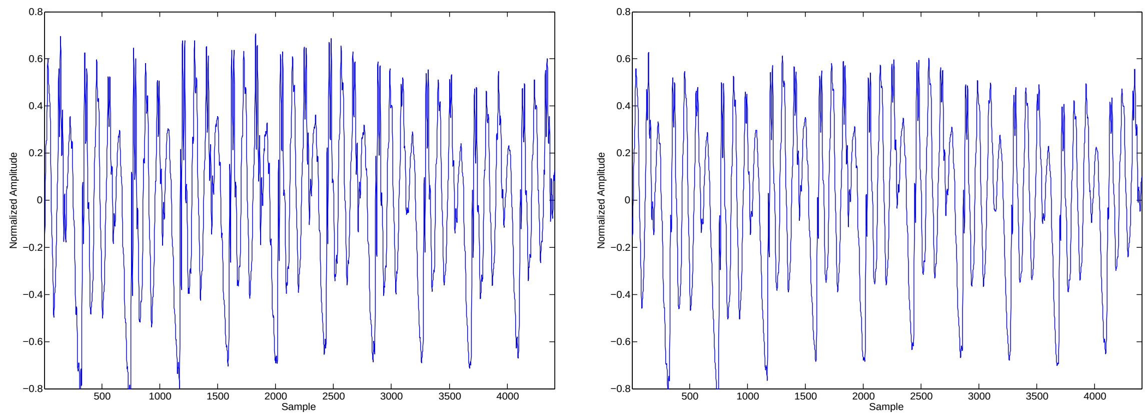
CHAPTER III

Theory

3.1 Theory Of Operation

3.1.1 Acoustic Signal Aquisiton

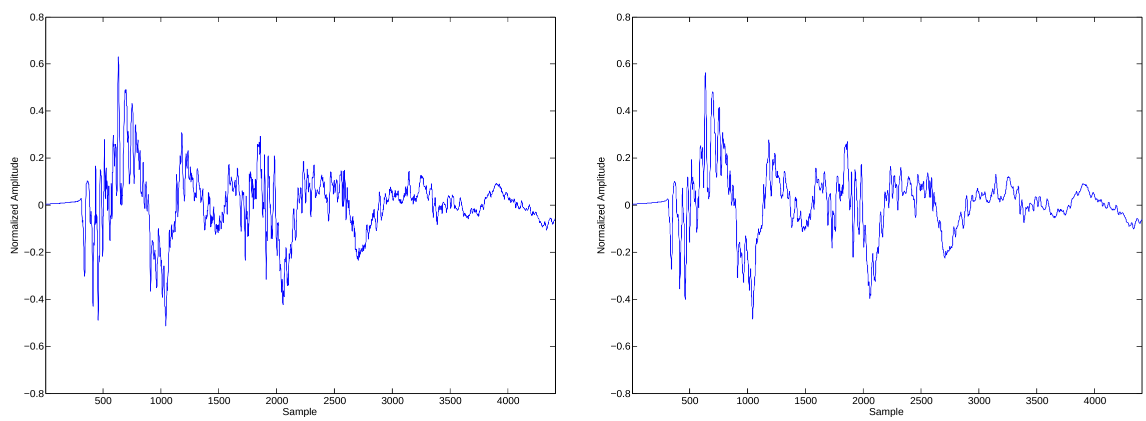
The Analog to Digital Converter (ADC) present on each sensor node's processor is responsible for sampling the incoming microphone signal. One of the most important parameter of concern to acoustic signal processing and in particular, to acoustic source localization is that of sampling rate. The sampling rate is the number of times the ADC reads the value of a signal in a period of time. Typically, acoustic sources of concern closely match that of the human hearing ranges of 20 Hz to about 15 KHz. Despite the average human's frequency response range being so large, the majority of sound energy is concentrated in a very short range, well below 5 KHz. Figure 1 - Figure 4 show some common sounds sampled at 44 KHz, and the same sound, frequency limited to 2 Khz. In most commonly occurring sounds it can be seen that there is minimal loss in the sound's major features. The presence of higher frequencies often provide fine details about a particular sound, especially human speech, which we recognize and distinguish readily. These show that despite the loss of a significant portion of higher frequencies, the acoustic signals are still unique and readily distinguishable from each other, at least with respect to the envelope of the sound. For the purpose of ASL, we are concerned more with the ample presence of the envelope of the sound, that is, the general outline of the sound.



(a) Speech sampled at 44 KHz

(b) Speech sampled at 4 KHz

Figure 1: Human speech sampled at 44Khz and 4Khz

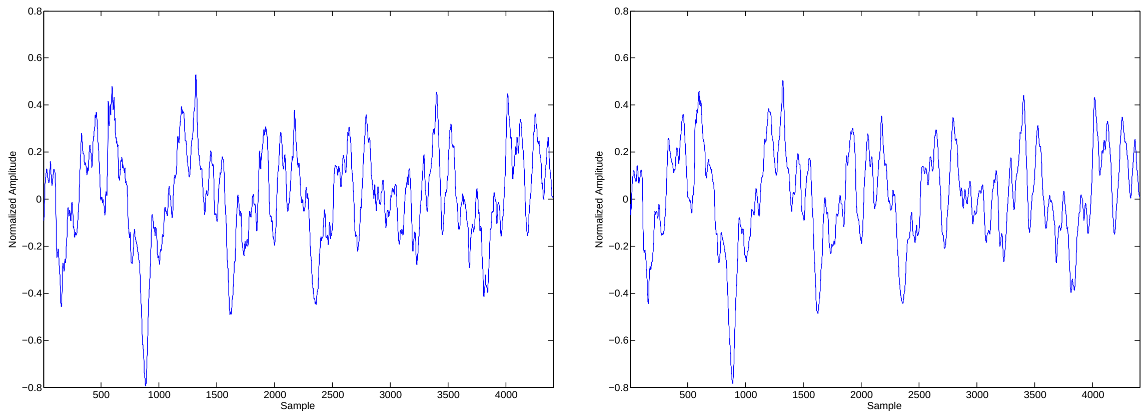


(a) "Thud" sampled at 44 KHz

(b) "Thud" sampled at 4 KHz

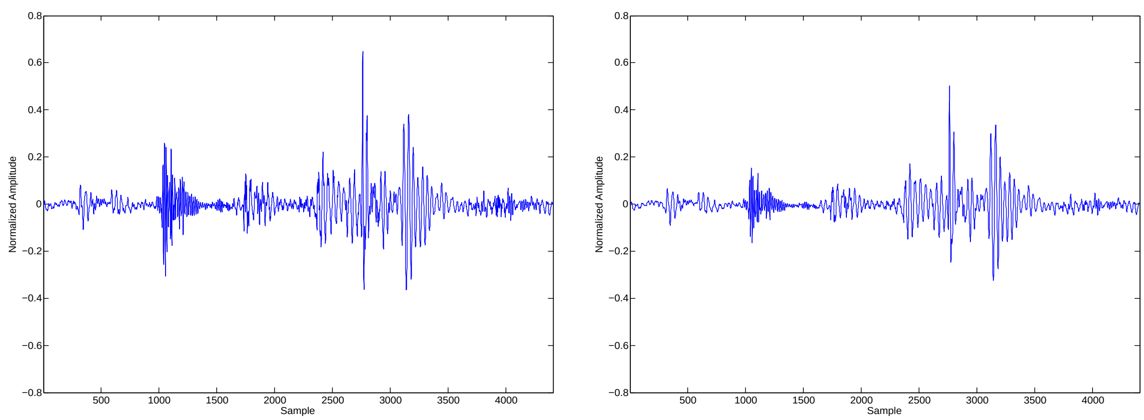
Figure 2: Desk "thud" sound sampled at 44Khz and 4Khz

Figure 5 - Figure 6 show the frequency range vs energy information of the four sounds from Figures 1 - Figure 4 that were sampled at 44 KHz. The frequency spectrum information shows that the largest energy is present below 2 KHz frequency. This can be seen as peaks above a -20 dB magnitude as a rough estimate, which almost entirely occur frequencies below 2 KHz. These examples show that high frequency



(a) Fan sound sampled at 44 KHz (b) Fan sound sampled at 4 KHz

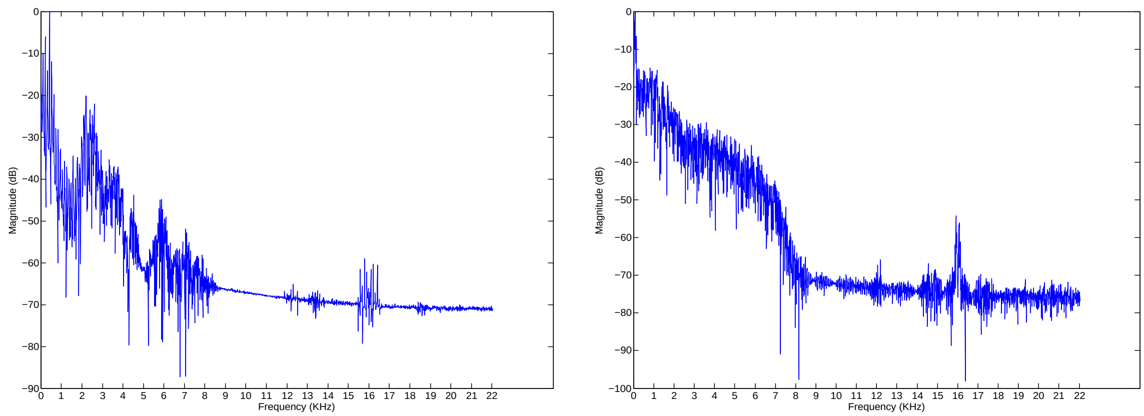
Figure 3: Table fan sound sampled at 44Khz and 4Khz



(a) Water flow sampled at 44 KHz (b) Water flow sampled at 4 KHz

Figure 4: Flowing water sound sampled at 44Khz and 4Khz

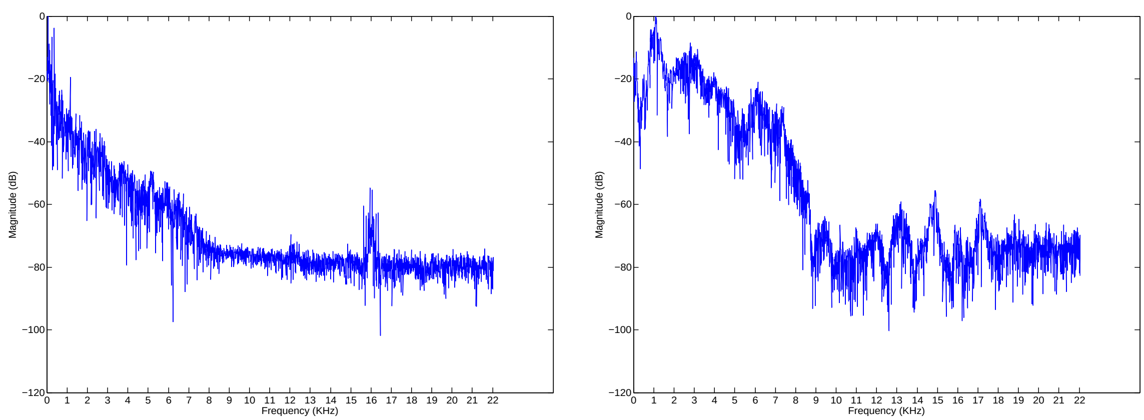
sampling is not necessary to obtain important features in a signal. One of the primary challenges faced by WSNs is the limited energy and processing capabilities available. Therefore, it is important to select an optimal ADC sampling rate such that only the significant information is retained. For most acoustic signals for the purpose of source localization, frequencies below 5 KHz should provide most of the information. For a given desired sound to be acquired, Nyquist criteria says that the sampling rate required must be at least twice that of the maximum frequency to be captured. Therefore for a maximum frequency bandwidth of 2 KHz, a 4 KHz sampling rate by the ADC would be minimum. Because of aliasing effects however, a slightly higher sampling rate is required assuming there is negligible energy at frequencies higher than the desired maximum. Ideally for a real-time ASL sensor network, continuous sampling and processing would also be required. A circular buffer containing the data that is constantly updated would be one approach. The length of stored acoustic samples would depend on the buffer size, which in turn is dependent on the available free RAM on the microcontroller.



(a) Frequency spectrum of speech

(b) Frequency spectrum of ”thud”

Figure 5: Frequency spectrum of speech and impulse sounds



(a) Frequency spectrum of fan

(b) Frequency spectrum of water flow

Figure 6: Frequency spectrum of fan and water sounds

3.1.2 Change Detection Processor

The sampled acoustic data must be processed in order to recognize an event of interest that occurred. This is the job of the change detection processor. Operating on a highly limited microcontroller restricts access to sophisticated signal processing algorithms such as those relating to frequency domain analysis in real-time. Therefore much simpler algorithms such as threshold crossing detection is often used for detecting if and when a significant change occurred. The idea behind threshold based detection is simple. The signal is checked sequentially and when the signal passes a certain threshold value, an event is said to have occurred at the time when the signal surpassed the threshold value. In reality however, thresholding methods are often more sophisticated in order to avoid false triggers [11]. Even with added complexity, thresholding methods are among the simplest and most efficient change detection methods applicable to signals.

In much of the existing research, the time of the significant amplitude change is often used as the Time Of Arrival (TOA) necessary for localization. This method however is prone to significant errors in TOA. Primarily, there are two large sources of error when using thresholding methods to derive TOA. First, due to the physical separation of any two nodes, the time the same signal reaches the two nodes in most cases will be different. This in addition to low signal sampling rates will cause the threshold to be triggered slightly out of phase with respect to the correct phase. This is essentially phase error where increasing the sampling rate will reduce this error. The second significant problem with threshold detection methods is caused by the varying signal strength between two nodes. Assuming there is sufficient signal strength to trigger the threshold on both nodes, due to the difference in signal strength between the two nodes, one node will trigger with a greater phase difference then expected due to a faster rise in signal amplitude required to meet the threshold limit. This effect is however independent of sampling rate.

In less resource limited systems, signal cross-correlation is another frequently used technique for determining time differences in signal propagation between sensors. Cross-correlation is a method used to determine the phase difference between two identical signals separated by some phase shift in time domain between the two signals. Cross-correlation methods however have significant limitations in its simplest form. In an environment where reflections are present, significant distortions will occur for the signal of interest. To help mitigate this problem to a degree Generalized Cross Correlation with Phase transform (GCC-PHAT) is used [3]. GCC is a fairly accurate way to determine time differences assuming there is sufficient signal length. GCC requires a sufficiently long signal before it can be processed and also relies on Fast Fourier Transform (FFT) which is a compute intensive process best left for Digital Signal Processors (DSP). Therefore, in resource limited sensor networks cross-correlation methods are best avoided because of their significant computational requirement.

Our signal processing therefore relies on modified thresholding methods due to their compute friendly requirements. However, unlike traditional methods, we perform significant further analysis before deriving the TOAs. This analysis allows for much more precise computation of TOA. Another significant reason to use a threshold value as the trigger for change detection is so that it ensures sufficient signal strength is available to many nodes. Having a high signal-to-noise ratio is critical for performing accurate processing especially signal matching. The first stage of the change detection processor is on an individual node level which looks for significant changes in signal strength. This significant change would be indicative of a significant acoustic event. Once the change is the detected, the node records the time with respect to its local clock and proceeds to the next stage for signal analysis. Figure 7 shows the flow chart of our theory.

3.1.3 Primary Node and Arbitrator Node

Now that at least a few nodes have been triggered by an acoustic event near the nodes, it is time to determine the primary node. The primary node is of importance because it is the one node that contains the greatest information on the particular acoustic event. The important signal characteristic information provided by the pri-

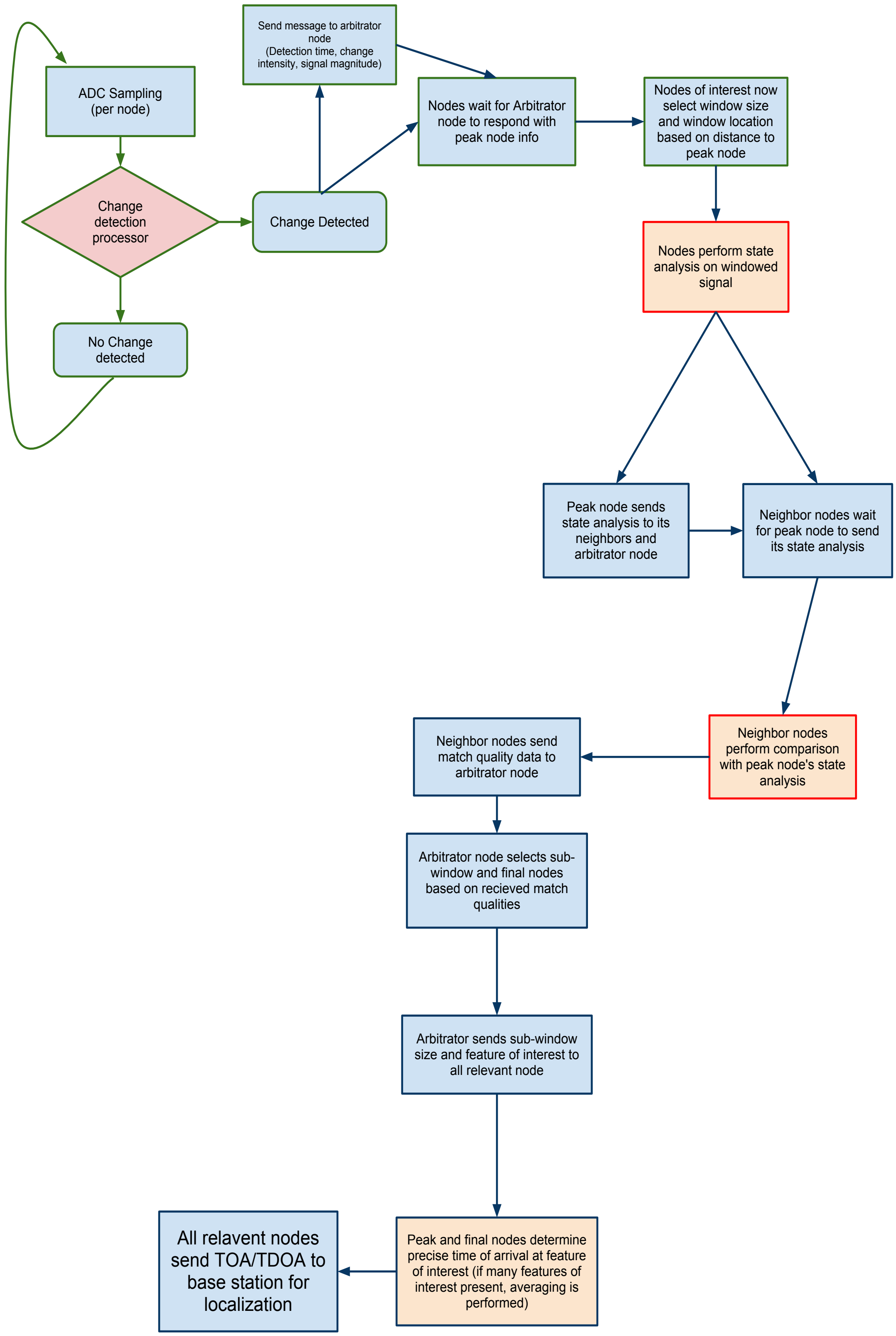


Figure 7: Theory Flow Chart

mary node will be used by other nodes for comparison. Sound is a mechanical wave that propagates in all directions in 3-Dimensional space given that it is unrestricted. Because of this, the sound wave reduces in intensity as radius r increases according to:

$$Intensity : I \sim \frac{1}{r^2} \tag{3.1}$$

In addition, sound travels at a velocity of 343 meters per second at standard temperature and pressure. As a result for any two nodes with different distances from the acoustic source, the node closest to the source would measure greater values in its signal along with arriving earlier. Accordingly its signal-to-noise ratio would also be the lowest assuming the magnitude and characteristics of the noise is similar in all nodes. Because of this, the primary node will be the node with the greatest information of the properties of the acoustic event and source. Therefore, the primary node is the node closest to the acoustic source.

Deciding which node is the primary node is the job of the arbitrator node. The arbitrator node is simply a dedicated node whose purpose is to make decisions for the sensing nodes and also acts as a gateway to the localization computer. Once the sensing nodes are triggered by an acoustic event, they immediately send the arbitrary node information of the triggered event. Particularly, the time the triggering occurred on a global time scale, and the peak amplitudes of the signal near the triggered event. Because the closest node to the source should see the greatest magnitude and earliest time of event, the amplitude of the signal along with the trigger time can be used to determine the closest node to the source, that is the primary node. The sensing nodes that were triggered in the meantime wait for the arbitrator node to respond with the primary node information. Once the sensing nodes, including the primary node receive the primary node information from the arbitrator node they determine an appropriate subset of the total sampled signal appropriate for further processing. This is because only a small portion of the acoustic event is of interest to us.

3.1.4 Signal Windowing

So far the sampled signal is continuous and unrestricted in size with exception to the buffer size limit. Processing the entire samples buffer is wasteful except if frequency-domain based methods like GCC are used. For thresholding methods, only a small subset of total buffer is sufficient assuming the start of the acoustic event is contained in the window. Because of the time delay in the acoustic signal reaching different nodes that vary in distance from the source, the time of trigger, which is the start of the event will also vary accordingly on a global time scale. Therefore, the window of samples selected must contain sufficient signal length surrounding the time of trigger. The size of window is of importance as unneeded processing can be avoided by having a sufficiently small window without discarding useful information. The window size is primarily determined by the distance between the primary node and the corresponding node. This inter-node distances are constant in most sensor networks and therefore the coordinates of each node can be programmed into the memory of each node beforehand or determined once after node installation. Distances from any two nodes can be calculated simply by Euclidean distance formula. Based on the distance and propagation of sound, a safe window size is calculated by empirical methods.

Another significant reason for an appropriate window size is to help avoid repeating patterns in the sampled signal. Many naturally occurring acoustic events exhibit stationary behavior at least to a few periods beyond the initial peak of the signal. Figure 1 which shows speech patterns clearly showing this stationary behavior. To a certain extent other naturally occurring sounds also exhibit this stationary behavior. The most interesting part of the signal is obviously in the initial few periods after the acoustic event peak. Therefore the ideal window size would save the need to process repeating patterns. In a sensor network where the inter-nodes distances are constant however, this window size will also remain constant and therefore this step can be eliminated. Once the window size is known either beforehand or by calculation, the placement of the window is such that it contains some insignificant signal before the trigger point and the most significant portion of the signal after the point of triggering. The node can therefore have a predetermined window offset determined adequate by empirical methods beforehand. Next the windowed signal is processed by the node using pattern recognition.

3.1.5 Signal Analysis

The windowing process ensures that signals similar enough will be compared by pattern matching techniques. Ideally the acoustic signal captured by all nodes would be identical copies, different in no way other than in amplitudes and time differences due to the basic properties of wave propagation. If the variations were only that of magnitude and phase difference, the pattern matching would be a simple process. However, in reality the signal will vary noticeably sometimes significantly in their pattern depending on the locations of each node for a given environmental setting. Many factors influence the captured signal's pattern. Among the most significant factors are those of sample timing, noise profiles, and reflections which are different at each node. Sample timing, which is a combination of time synchronization between nodes and distance of nodes from the source will almost always result in phase mismatches such that a captured local wave would be slightly different. Oversampling the signal can ensure the signal's fidelity, which is however not feasible for a low power WSN. The noise profiles are dependent on the physical location along with hardware dependent noise unique to each node. Different noise sources in the environment can effect the nodes in unpredictable ways due to reflections. Hardware unique to each node especially the analog components such as the microphone, amplifiers and ADC can cause noticeably different noise profiles in each node. Reflections, besides contributing to the noise profile acting as unwanted sources, if large enough can cause significant distortions at each node. Reflections, especially in an closed environment with highly reflective surfaces such as concrete office walls can create significant distortions in sounds based on the location. These factors combined, make analyzing the pattern for comparison a challenging task for low power WSNs.

There are two possible approaches that could be taken for the purpose of signal analysis. First, the windowed signal could be transmitted to a significantly more powerful base station node so that it could perform cross-correlation. Though this is much more efficient than transmitting the entire signal and cross-correlating amongst various pair combinations, it would be much less distributed and inefficient when compared to a more distributed approach. More distributed approaches involve performing local processing such that only key features of the windowed signal are identified and sent to the arbitrator node for comparison. This process of recognizing features would have to ensure those features are common among the captured signals in nodes being compared. In the case of no signal degradation besides amplitude, the features of interest will be identical on all nodes making the comparison trivial. But due to various factors described, the identification process will require pre-comparison communication to identify key features. The process can be approached by various pattern matching techniques. The most basic of which are the cross-correlation methods which we want to avoid due to their high resource requirements. Pattern recognition based on state analysis is a possible method that requires minimal resources compared with frequency domain based correlation [4]. State machine analysis such as string matching is one such technique. Fuzzy matching of patterns is important due to the variable nature of sampled signal. Among the pattern recognition methods, unsupervised recognition would be required as the patterns to be analyzed can follow a widely varying pattern and supervised matching would require a large training set.

3.1.6 Key Feature Selection

Selecting key features is critical for proper comparison of varying patterns on the nodes. Key features are assumed to be features significant enough to be well preserved across nodes. Two broad areas of analysis that should be analyzed for significant features are those of time and frequency domain. In time-domain analysis for example, we can look for the envelope of a signal measured by its amplitude. Patterns in the envelope of a signal tend to be more preserved across nodes in comparison to individual peaks or troughs. A derivative of the envelope measurement method is the basic threshold analysis. Threshold analysis made for a significantly longer time can be seen as envelope analysis. Signal magnitudes alone can vary significantly across nodes especially due to reflections. Frequency-domain analysis on the other hand when seeking frequency changes is significantly less affected by reflections, though the amplitudes of the signal can vary depending on the frequencies. The most common frequency domain analysis technique is the Fourier Transform. In practice Fourier Transform is performed using the Fast Fourier Transform (FFT). The forward version of FFT would take a signal in its time domain and output its frequency-domain spectrum. FFT however requires a significant calculation time on a microcontroller without DSP capabilities.

Zero-crossing rate is another frequency measurement technique suitable for highly repeating signals such as those seen in acoustics [2]. Zero-crossing rate also requires significantly less computational resources along with it not being restricted to DSPs for efficiency. FFTs require "O(n*log(n))" operations whereas zero-crossing rate analysis would require "O(n)" operations where n is the number of samples in the signal. For the purpose of finding significant changes in a signal's frequency changes in a highly limited system, especially when time information is important, zero-crossing rate filter would be of more use. Once the signal is traversed using amplitude analysis along with zero-crossing rates, possible key features can be identified for pattern recognition between nodes. Key features are only considered important once the pattern recognizer determines common features between signals in different nodes that could be matched together. Therefore the pattern recognizer is really a networked pattern analysis algorithm working with all relevant nodes simultaneously.

3.1.7 Signal Matching and TOA Calculation

Once the networked pattern matcher finds significant features in common with the nodes of interest, TOAs need to be calculated for each node. Each feature in the signal is separated in time relative to a global time scale. Assuming the nodes are synchronized, the matched features should be separated uniformly depending on a node's distance from the source. If we make the assumption that the features do not have any distortions which result in skewing the TOAs, we would only need one common feature for comparison. In reality however these features even though significant in the nodes, will see at least some distortions in their TOAs. Most real world noisy measurement are processes that have a zero mean distribution. Regardless of the distribution the mean error with significant number of measurement should reduce to a system's minimum. By using this fact we can take the TOAs of as many features as possible to reduce the error in TDOAs which is needed for localization. In addition, this process of feature comparison and TDOAs extraction can be done simultaneously. Only two nodes with common features can be compared. It is very possible in the features found on nodes, nodes would have some features not present in other nodes, therefore the pattern matching and TOA calculation must be robust enough to not consider features unique to a node. Once all features are exhausted across nodes of interest the TDOA error between pairs is considered to be minimized and therefore sent to the base station for localization.

3.2 Acoustic Source Localization

Source localization is the process of using time differences of signal received at various receivers to determine the position of the source relative to the positions of the receivers. There are several methods of localization available, the most common being TOA based and TDOA based. TOA method is used in GPS where the time of arrivals are known from all transmitters. GPS localization is the inverse form of localizing a source, such that the receiver is localized with respect to a global coordinate system. In GPS localization the receiver, such as a hand held GPS unit receives signals from various geo-synchronous satellites with a globally synchronized time stamp. In effect, the GPS unit knows the TOA of the signal from each satellite. These TOA are input into an algorithm which represent the intersection of three or more spheres. The intersections represent the possible location of the receiver. Usually more than one intersection is present but because of improbable locations of the other intersections only one location is chosen, the real location of the hand-held GPS receiver. Time difference of arrival, TDOA methods on the other hand do not require the TOA of the sources signal to the receivers in a global time frame. While similar to TOA based localization TDOA method are used when the the actual time of the signal's transmission is not known. The is the case in acoustic source localization where the acoustic signal information contains no information on the time it was sent by the source. However, the TOA of the signal at each nodes is know because the receivers are time synchronized. This is the case with our work.

The received TOAs are relative to a synchronized time frame between nodes starting from the first sample of the signal in each node. The TOAs represent the separation of the number of samples between pairs of nodes. TOAs are easier to compute directly in our research as direct TDOAs computation is done by cross-correlation methods. Whereas TDOAs can also be computed indirectly by taking the differences of TOAs. Specifically if t_i, t_j, t_k represent the TOA of a common feature in three different nodes based on a synchronized time scale then the TDOA can be calculated as:

$$t_{ij} = t_j - t_i \tag{3.2}$$

$$t_{ik} = t_k - t_i \tag{3.3}$$

$$t_{jk} = t_k - t_j \tag{3.4}$$

In theory t_{jk} contains no relevant information that t_{ij} and t_{ik} does not already carry. In practice however, due to various sources of error t_{jk} can be used as a basic test of the validity of t_{ji} and t_{ik} using the triangle inequality rule.

$$t_{jk} \leq t_{ij} + t_{ik} \tag{3.5}$$

This is a commonly used method in both TOA and TDOA localization techniques as a basic validity test. Once t_{ji} and t_{ki} are deemed valid by the triangle inequality test they are sent to a 2D localizer. For 2D localization based on two TDOA, two non-linear equations can be generated.

$$t_{ij} = \frac{1}{c} \times (\sqrt{(x - x_j)^2 + (y - y_j)^2} - \sqrt{(x^2 + y^2)}) \tag{3.6}$$

$$t_{ik} = \frac{1}{c} \times (\sqrt{(x - x_k)^2 + (y - y_k)^2} - \sqrt{(x^2 + y^2)}) \tag{3.7}$$

Where $[x_j, y_j]$, $[x_k, y_k]$ and $[0, 0]$ are the locations of nodes at know location in the global coordinate system and c is the propagation velocity of sound, which in our case is 343 m/s at STP. The equations generated represent a system of two non-linear equations with two unknowns $[x, y]$ which is the location of the sound source. While in theory the system always converges to a valid point, in reality due to numerous sources of errors they are often multiple solutions and sometimes no valid solutions exist. This then becomes an optimization problem. Further, most localization systems rely on more than three sensors in order to over-determined the system for optimization. Once overdetermined, optimization techniques such as least-square method could be used to find the most likely solution. This is the case with our work as well and will be discussed further in Section 4.5.

CHAPTER IV

Implementation

4.1 Core Components

While it would be of the greatest benefit to have the whole system built equivalent to that described in the theory section, due to the scale of the problem we limit our implementation to the core ideas proposed for low power acoustic source localization. A real-time low power WSN for acoustic TOA calculation proposed by us would require substantial work possibly beyond the scope of one masters thesis. Instead in the theory section we describe the full possible idea behind such a system once fully realized. However, what has been implemented is the most critical component of the whole system in order to evaluate the core ideas presented in Chapter 3. First, in order to test even these core ideas behind our work, many components commonly present in other related previous works needed to be implemented. Figure 8 shows the important components of the overall theory that was implemented and tested.

4.1.1 Acoustic Event Detection

Similar to what was described in Section 3.1.2 for acoustic event detection, we implemented a basic triggering mechanism to look for interesting events worth local-

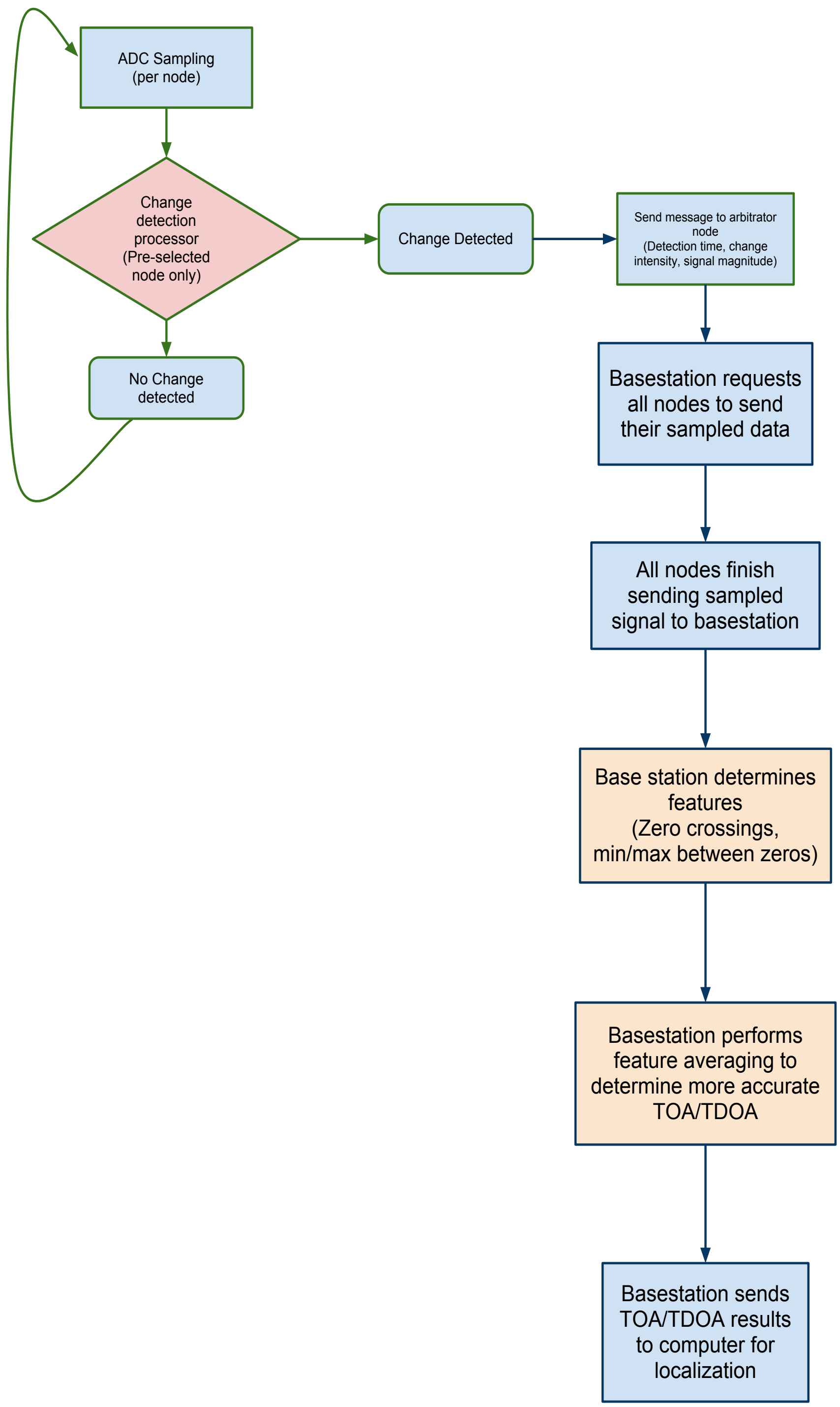


Figure 8: Implemented Theory Flow Chart

izing. However, several key changes have been made, which while still allowing for the demonstration of the principle, simplify the implementation. The trigger processor module is where the nodes continuously monitor the acoustic medium and triggers the localization process when sufficient activity is detected. Unlike in Section 3.1.3, where all sensing nodes perform this trigger processing on the signal they sample, in order to reduce complexity we only require one node to perform this processing. While in a real implementation acoustic events can occur at any location, for testing purposes we ensured the acoustic events occurred closer to the primary node. One of the problems we encountered with allowing all nodes to perform trigger processing is that when a significant acoustic event occurs, all nodes that triggered were required to notify others or just the base station of this event. However because acoustic events require a short time to travel between nodes, the notification sent by each triggered node cause radio message contention frequently resulting in dropped messages, or worse, no messages being sent properly. This was the primary reason for only allowing one node to trigger and notify. However by using one pre-determined node to trigger, we simply make that node the primary node for all testing.

A key requirement for a real-time system is continuous sampling so that no events are lost. Due to the significant challenge in building a real-time system with low-power motes we opted to sample selectively. We currently sample continuously for up to one thousand samples and then immediately process them. During the event detection processing of the sampled data or if an event is detected, we do not perform any sampling. Only after the sampled data is fully processed and evaluated the sampling on the nodes is resumed. This is done from the base station which sends a resume sampling signal to all sampling nodes. This simplification though removing the real-time criteria, does not interfere with the evaluation of the underlying principles behind our idea. Because the real-time criteria is removed we can no longer be concerned with missed event. However, we ensure that at least some event exists for processing by continuously playing significant acoustic events from a speaker. This way we can test the core idea while maintaining consistency.

One of the problems we encountered while sampling between nodes is that of clock drift, which will be explained in Section 5.2.1. Due to clock drift, only about the first 10 ms of collected data is properly synchronized. Because of this we largely limit our processing to this first 10 ms of collected data. The frequency of sampling however changes the collection time. Therefore to further simplify this we simply look for triggers only in the first 100 samples as long as the sampling rate is below 15 Khz, which we found empirically. This number is varied as required for the cases where sampling rate is high. Once a trigger occurs in this restricted sampling window, samples that occur at a time higher than the end of the window are used sparingly in order to alleviate the effect of phase shifting cause by different clock drifts of the nodes. However this is true only for lower sampling frequencies. We made use of as many as 500 samples when sampling at 15 KHz. We however also employ a clock drift compensation mechanism to help alleviate some of the negative effects due to this.

4.1.2 Sense Nodes and Base-Station

The sense nodes perform the basic operation of sampling the acoustic signal. The ADC unit on each sensor nodes use Direct Memory Access (DMA) in order to achieve low power sampling. The sampling rates are varied for various testing purposes from a minimum of 5 KHz to up to 15 KHz. One of the sense nodes is assigned as the primary node. The primary node is responsible for performing trigger processing on the initial 25 - 100 samples of the sampled signal and sending a message to the base-station should a significant event occur. The sense nodes also send raw data to the base-station when requested by the base-station. Currently, no processing is performed on the sense nodes as our current implementation is not appropriate for real-time processing, which is one of our main purpose for making use of distributed computing. While we could easily perform the basic calculation on the sense nodes, our current implementation is designed more for testing the underlying theory than to build a fully operational system. For the sake of simplicity we therefore download all sampled data from the sense nodes to the base-station. Just before sampling, the sense nodes also perform General Purpose Input Output (GPIO) based wired time-synchronization which is a critical component in distributed sampling required by our system. Using GPIO-based time-synchronization we are able to achieve sub μS synchronization. GPIO-based time-synchronization will be described in Section 4.3. The base-station is responsible for downloading the sampled data from the sense nodes. Once downloaded, the base-station analyzes the signals from the sense nodes and determines the TOAs. The processing done by the base station currently implemented perform zero-crossing and minima and maximua calculations. These are key features we use as comparison points between the sense node signals. These are explained in detain the following section. Once the TOAs are calculated, they are sent to the connected computer for localization.

4.1.3 Base-Station Processing and TOA Calculation

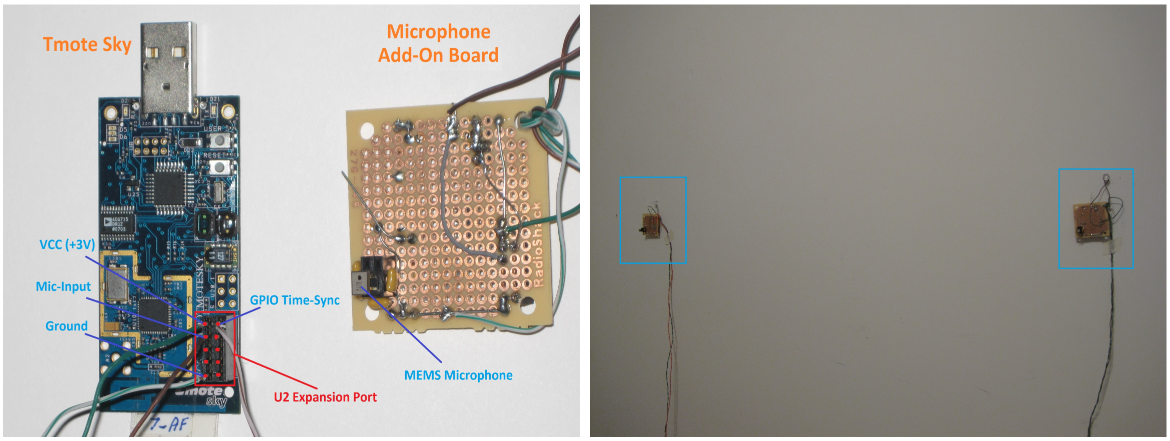
The base-station performs two different signal analysis for extracting features. Zero-crossings are one of the most easily detected features of a sinusoidal signal, therefore our TOA calculations relies heavily upon them. Another features are local peaks between zeros including local maximums and local minimums. Zero-crossing detection is in general a simple process requiring "O(n)" steps. We take this one step further and determine the precise location of the zero-crossing by interpolation. Like standard zero-crossing detectors we travel each signal until there is a change in sign indicating a zero. The zeros location however is not precise enough as it lies between two samples, one taking a positive value and the other taking a negative value. We convert the locations to floating point values and calculate the slope from this. Using the slope, and y-intercept we generate a line function between these two points. Finally we solve for Y=0 in order to get the x-intercept which is the zero-crossing with sub-sample accuracy. Depending on the sampling rate, all zero-crossing up to about 500 samples from the start are recorded for averaging later on. Only zeros that are sufficiently spaced apart are valuable as they represent a significant change. We ensure that this is the case by testing with a sound source such that the ADC sampling rate even at 5 KHz oversamples the signal for consistent data.

Once the zeros have been determined we proceed to get the local minima or maxima between the zeros. Only the minimum or maximum is recorded between two consecutive zeros. This is because the largest absolute value represents the most significant part of the sampled signal between the zeros. This point is most likely to be captured across the sensed signals between nodes. Small local peaks are often lost or combined into larger peaks, so we avoid them. Again, we step through the signal between zeros and keep track of the location of the maxima and minima along with their amplitudes. Interpolating maxima and minima values for better accuracy will require substantially more processing than interpolating values between zeros. Therefore we keep the location of the local peaks as they are for crude correlation purpose only. Finally, with the collected locations of zeros, local peaks are also averaged together however separate from the zeros. That is, one average indicates the TOAs of the zeros and another for the TOAs of the peaks. These average values are the TOAs for that signal relative to the start of the signal sampling which is common to all sampling nodes due to time-synchronization. The purpose of averaging the features is that though each feature can contain significant error for various reasons, by averaging many features the result will converge to the most likely TOA. We assume the errors in sampling is a zero-mean process.

4.2 Hardware

Acquiring acoustic data in precise metrics for localization purposes is a challenging task for several reasons. Limited available memory for sample data, low-power high-noise ADC modules and networked synchronization are some other big challenges encountered. Given the limited resources available in a sensor network node, not much room is available for sophisticated acquisition qualities like those available on a desktop computer or other floating-point digital signal processors. In our hardware design we start with the basic Tmote Sky sensor mote designed for low power WSNs [18]. The Tmote Sky has an ultra-low-power consuming Texas Instruments MSP430 micro-controller along with several onboard peripherals. Some of the key features of the MSP430 are that of ultra low-power consumption [29], wide voltage range of operation and sufficient resources for many WSN applications, making it one of the most popular microcontrollers for WSNs. The MSP430 is equipped with a 12-bit ADC which can sample at a theoretical maximum of 430 KSPS. While this number is significantly higher than what is used in this work, having that available headroom means a lower sampling rate can be achieved with minimal resource usage. The MSP430 ADC is wired externally to several expansion pins available on the Tmote. In particular we use the U2-10 pin expansion port which has at least one available ADC input, at least one General Purpose Input-Output pins (GPIO) necessary for time-synchronization used in this work and power supply connections (+3V and GND). For our work, these are the only connections needed to be accessed and using port U2 is sufficient for all our needs.

In order to access and use the T-motes resources through the U2 port, we built a custom circuit board with necessary analog electronic components along with a microphone. There are several kinds of low power and small microphones available on the market. The most popular are electret microphones which have excellent qualities ideal for low power acoustic applications. They have been used in most acoustic sensor network research before. However for our research we opted to use the relatively new MEMS based microphones. MEMS microphones provide several key advantages over the common electret microphones. First they often have precise built-in circuitry to significantly reduce the number of on-board components which might also require fine-tuning. Second, MEMS microphones are more tolerant to variations in temperature while soldering them by hand, something we found to be a problem in electret microphones [1]. Otherwise, most other qualities we seek from MEMS microphones are comparable to that of electret microphones. The MEMS microphone we use in particular is the Knowles acoustic SPM0408HE5H. This microphone has a relatively flat frequency response between 20 Hz and 10 KHz in addition to built-in amplification and lower consumption [12]. In our testing we found however that the 20 dB maximum amplification provided by the SPM0408HE5H is insufficient for capturing sound near the sensor nodes as the SPM0408HE5H was designed primarily for cellphones with human speakers right next to the microphones. To solve this problem, we added further amplification using a LMV324 op-amp [28] and necessary analog components for an additional 20 dB gain. In total, the weak signal from the internal microphone is amplified by 40 dB which we found to be sufficient. Figure 9 shows the picture of our add-on board attached to a Tmote Sky Mote along with an example of node placement in the testing environment.



(a) Tmote Sky with microphone add-on board

(b) Two microphone nodes placed 1 meter apart on wall

Figure 9: Hardware setup

4.3 Wired Time-Synchronization Protocol

Time-synchronization as discussed before is a critical piece for distributed acoustic sensor networks relying on TOA methods. There are several available software based time-synchronization protocols available for WSNs that provide sufficient accuracy for acoustic WSNs. RBS and Vanderbilt's FTSP are two protocols that can provide micro-second level time synchronization suitable for such applications. We however choose not to use any software based protocol for two reasons. First, wired synchronization can provide significantly more accurate, sub-microsecond time synchronization good for testing purposes. Secondly, wired time-synchronization was easier to implement and quantify even though necessary code is available for the mentioned software protocols. However, in a real wireless sensor network, wired synchronization would defeat the purpose.

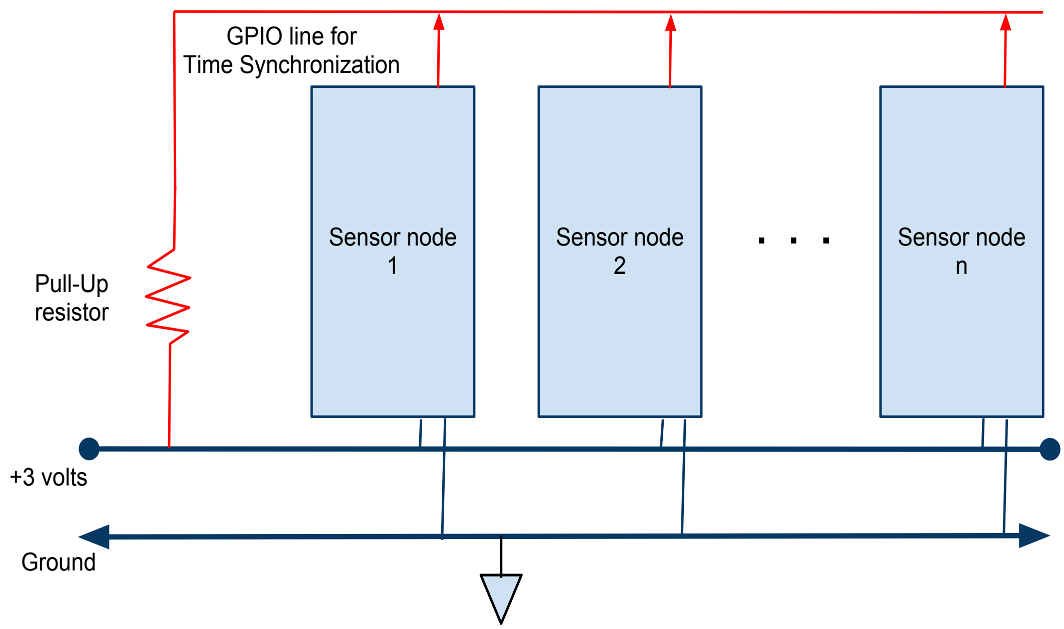


Figure 10: GPIO synchronization hardware setup

Wired time-synchronization though simple to implement, requires careful design in order to have both accuracy and precision. We use one of the available GPIOs on-board the Tmote, wired in common to each node on the network. Also connected to the GPIO is a resistor which on the other end is connected to the VCC (+3V) side of the power supply on board. The resistor acts like a pull-up resistor and causes the GPIO to remains high by default unless forced down by one or more nodes. In software all nodes first pull down the GPIO (logic 0). As each node is finished with some task ready for the next cycle of sampling the node makes its GPIO as a input, which causes that particular node to release its hold on the GPIOs low state. However the GPIO does not see a high (logic 1) state unless all nodes have released their hold on the GPIO. Once a node releases its hold on the GPIO it continuously reads the pin to see if it is high. The moment the high state is activated, all nodes resume sampling almost simultaneously. This method is extremely simple and provides a consistent accuracy under 1 uS when evaluated empirically. Ensuring such accuracy will eliminate considerations of time-synchronization error from our TOA calculations and thus simplifying them. To the best of our knowledge this is the first time this method is used for wired time-synchronization. Figure 10 shows a diagram of the time-synchronization hardware setup.

4.4 Software

Significant software services are needed for a real-time acoustic sensor network. A large part of this software directly deals with controlling the hardware. In order for us to program the nodes we make use of TinyOS [30], a nesC based software framework designed for sensor networks. TinyOS is an event driven operating system framework developed for managing embedded sensor networks [22]. TinyOS event driven framework makes it suitable for real-time sensor networks where actions and outcomes are asynchronous. TinyOS remains the most popular choice among sensor network researchers for a wide array of applications. In addition, TinyOS is perfectly suited for the Tmote sky due to extensive software implementation for Tmotes hardware. Over the course of software development we have made use of several key software components provided by TinyOS. We have two separate programs in our implementation one for the sensing nodes and another for the base-station.

The Sense Node program is identical across all sensing nodes. The sense node program is responsible for acquiring audio data, then sending the acquired data by radio to the base-station. Sense nodes also receive instructions from the base-station regarding sampling metrics and communicate with the base-station when samples are ready for transfer. Sensing on the sense nodes is done using Direct Memory Access (DMA) for accurate sampling and minimizing resource usage. The `Msp430Adc12ClientAutoDMA_RVGC()` component in TinyOS provides access to the ADC and DMA module in one convenient package. Using this component, continuous sampling can be performed with minimal CPU interference although we are currently sampling sequentially, that is, we sample first then transmit while sampling is suspended. This is not sufficient for a real-time acoustic sensor network, however it is sufficient to test our methodology. Radio communication is performed using the `ActiveMessageC` component while making use of the `AMSenderC()` and `AMReceiverC()` components. These provide access to the `message_t` buffer necessary for all radio communication. Time-synchronization by GPIO is performed by accessing the `HplMsp430GeneralIOC` component. These components make up the critical modules necessary for testing. There are several other components used as standard requirements and debugging such as the `MainC` and `LedsC` components. The `LedsC` component which provides control to three LEDs on the Tmote Skys, is used extensively for visual debugging and verification. A different GPIO via the `HplMsp430GeneralIOC` were used extensively for debugging the sampling and time-synchronization between nodes using an oscilloscope.

The Base-station node does not need to perform any sampling and therefore the ADC/DMA components are not used. Similarly, the GPIO interface is also not used as no timing or time-synchronization is necessary with the sensing nodes. Standard components such as `MainC` and `LedsC`, for debugging purposes, are used similar to the sense nodes. The most used component for the base-station is the `ActiveMessageC` component again along with `AMSenderC()` and `AMReceiverC()`. These form the radio communication module which is a prime function of the base-station. Once a message has been received by the base-station due to successful event detection, the base-station sequentially requests data and reads until completion before moving on the next node. Once the base-station receives sampled data from the nodes, the base-station performs signal processing on the acquired signals. Although our final aim is to move most of the processing to the sensing nodes themselves, for simplicity and ease of testing we are currently performing all feature matching and TOA caluclations on the base-station. Currently we process the signal to acquire the zero-crossing with sub-sample accuracy and local minimas and maximas between zeros. These values are averaged and sent to screen via the `printf()` library provided by TinyOS. The `printf()` library internally uses the UART serial communication port through the Tmote's USB port to send the results to the computer.

4.5 Localization

Once TOA data is sent to the computer, it is used to perform two-dimensional localization. Performing localization is in general a compute heavy task that is almost always performed on desktop class computers. One of the more expensive tasks in general for acoustic localization systems is the cross-correlation methods which we avoid. Most existing systems send their TOA or TDOA information to the base computer for localization. We have employed MATLAB to perform the final localization step, though the TOAs which have been determined by the base-station is entered manually into MATLAB. MATLAB is a convenient and powerful tool to perform complex calculations. The ability to visualize data conveniently is one of MATLABs strengths and has been made use of throughout this work.

The localization technique we implemented in MATLAB is hyperbolic positioning. As described in the Section 3.2, hyperbolic positioning is the placement of hyperboloids on a 3-Dimentional region representing the localization environment. Intersection of three or more hyperboloids at a point represents the acoustic source. We are limiting our localization to two dimensions only. Therefore we need only a minimum of three nodes or two time differences. However we make use of six nodes which greatly increases accuracy while allowing for optimizations to be made. The reason we cannot use spherical positioning such as that of GPS is that the global time is not know which includes the acoustic source. We only have a common time frame amongst the nodes.

We calculate two TDOAs from three TOAs as follows: $t_{ij} = t_j - t_i$, $t_{ik} = t_k - t_i$; where t_i, t_j and t_k are TOAs and t_{ij}, t_{ik} are TDOAs. In addition to this we also rely on the basic triangle inequality principle for the first level of error rejection. Triangle inequality says that two sides of a triangle must add to be greater than the remaining side. In our case along with t_{ij} and t_{ik}, $t_{jk} = t_k - t_j$ is also calculated. While in ideal measurements t_{jk} contains no new information, in practice t_{jk} can be used to verify the quality of t_{ij} along with t_{ik} as follows: $t_{jk} <= t_{ij} + t_{ik}$ must be valid in order to proceed with t_{ij} and t_{ik}. Next t_{ij} and t_{ik} are used to construct two hyperbolas on a

2D surface using the 2D hyperbolic equations.

Two hyperbolas are drawn on the surface with their intersection being the acoustic source. However in many cases even though the TOAs seem reasonably good, no intersection points exist due to small variations at certain configurations. In order to solve this we simply discard measurements that don't approach a proper intersection at the least. If we use four or more sensors for 2D localization, we could apply an optimization method such as least-square method [31] or gradient-descent in the event of no intersection points or multiple intersections. We currently use six sensing nodes. Therefore we make use of least-square method in MATLAB. In total, we have a system of five non-linear equations while only solving for two variables, X and Y, the location of the sound source.

CHAPTER V

Results

5.1 Test methodology

In order to test the implemented theory we perform the testing in predictable setting to minimize errors and to avoid any unusual behavior. We choose a sensor array of six nodes placed on the walls of a small room. The room is 10 feet in width and 13 feet in length. However only a portion of this room was used as our testing environment due to computer equipment in the room. Figure 11 shows a top view diagram of the room with the locations of the nodes. Six nodes are placed 1 meter apart from each other in the formation shown. While the location of the nodes are normally very important before installation, because we only consider the initial parts of an acoustic event sound, we are able to place nodes by convenience without being odd.

Node 1, which we pre-define as the primary node is positioned with a global coordinate as X,Y = [0,0]. Node 2 - node 6 are also sensing nodes but do not search for acoustic events unless instructed to do so by the base-station or primary node. The base-station is another node that is connected to a desktop computer and sends commands to the sense nodes and receives data from the sense nodes. The data is then processed at the base-station and displayed on the computer screen which

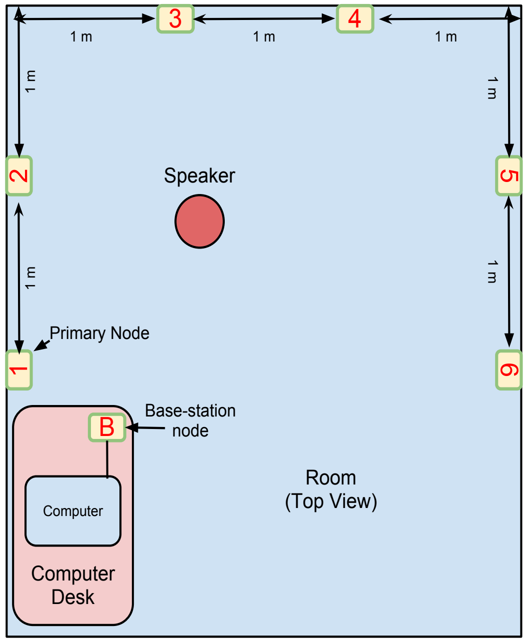


Figure 11: Top view of experiment setup. Nodes are placed 1 meter apart.

is finally saved to a file. The acoustic source we use is a speaker connected to the computer continuously playing a 1 KHz sine wave in pulses. Each pulse is composed of ten waves and there is a two second silence after each pulse. The silence ensures that the room is free of reflection effects before a new pulse starts. While this is not a realistic test environment, this is adequate to test the core ideas of the theory. We leave the testing in realistic setting with more natural sounds for future work.

Our test methodology involves localizing the source using a sampling rate of 5 KHz, 10 KHz, and 15 KHz. While we wanted to test higher sampling rates such as 40 KHz, we were unable to do so due to software limitations on our nodes. While the node hardware is in theory capable of much higher sampling rates, possibly because of software limitations imposed by TinyOS and/or our own software implementation, we limited our testing to a maximum of 15 KHz. As we play the sinusoidal wave pulses from our source, the primary node triggers, assuming the pulse begins in the node's first 100 samples. This is to reduce the effect of clock drift. Immediately the primary node halts further synchronization and sampling requests until the sampled event is localized. Once the capturing is finished, the data is uploaded to the base-station.

The base-station computes the features from each of the sampled node data and sends the first ten matched feature location to file on the desktop computer, which are the TOAs. On our computer we use a MATLAB script to perform the localization based on the extracted TOAs.

5.1.1 Localization with Independent TOA, Zero-Crossing Interpolation Disabled

The results of our testing are as follows. Figure 12 - Figure 14 shows the localization error at 5 KHZ, 10 KHz and 15 KHz at each matched TOA. Here localization at each matched TOA feature are evaluate independently, that is irrespective of the features before or after the particular TOA in question.

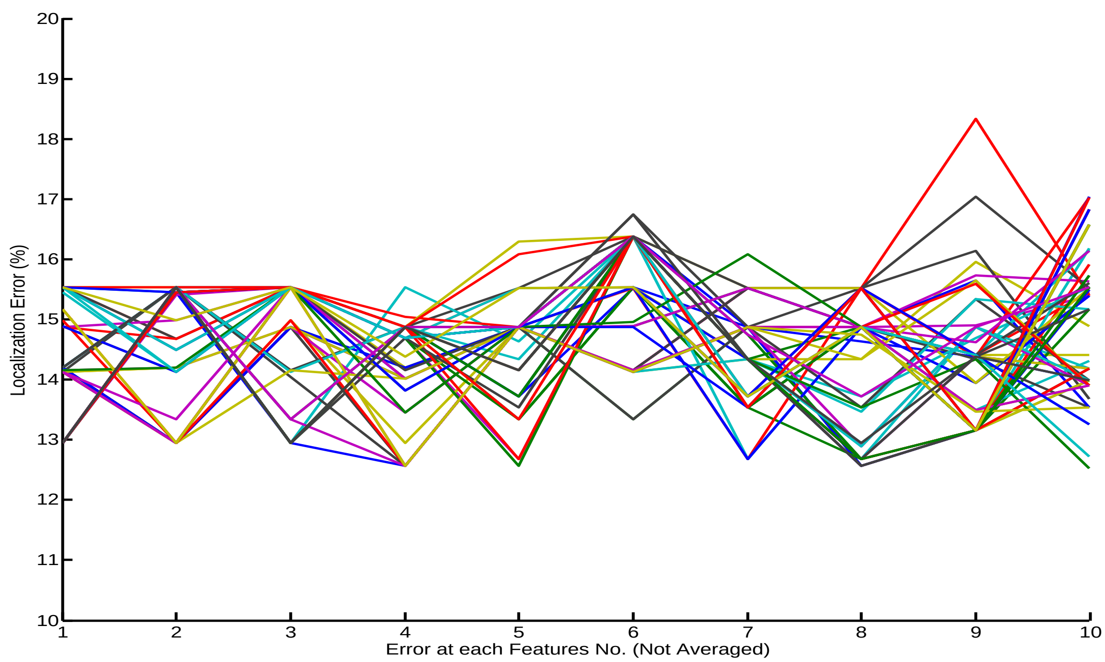


Figure 12: Localization error evaluated independently at each feature, Zero-Crossing interpolation disabled (5 KHz sampling rate)

The results show somewhat unpredictable localization error at any given feature. Each individual line represents a separate event caused by the sound source. It is clear that not all events have predictable behaviors. This is likely due to various

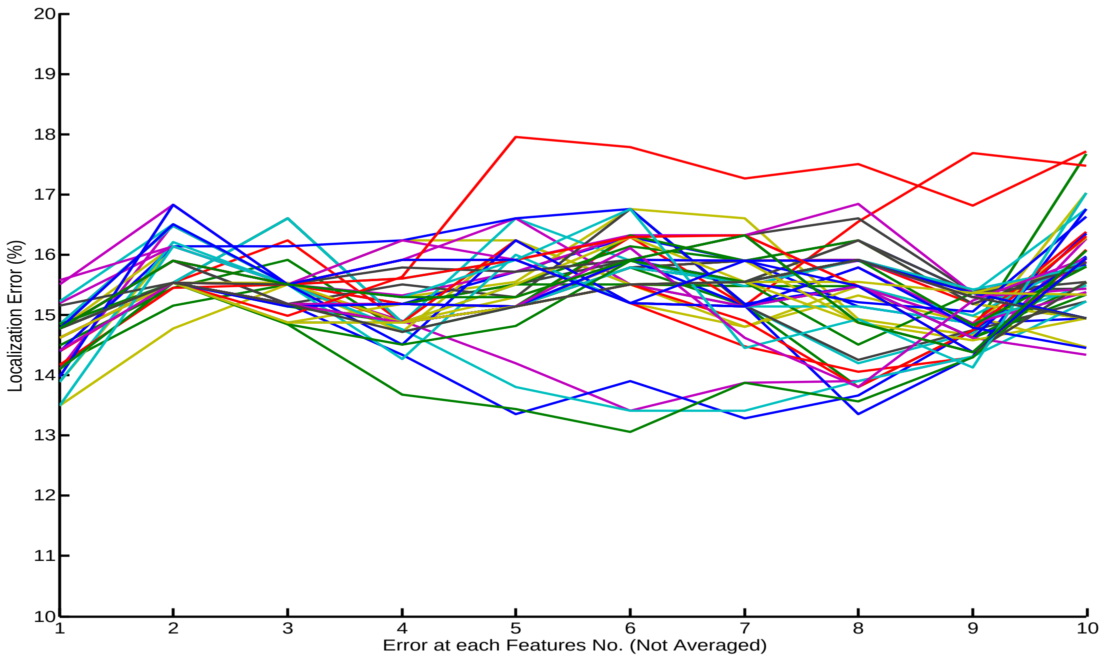


Figure 13: Localization error evaluated independently at each feature, Zero-Crossing interpolation disabled (10 KHz sampling rate)

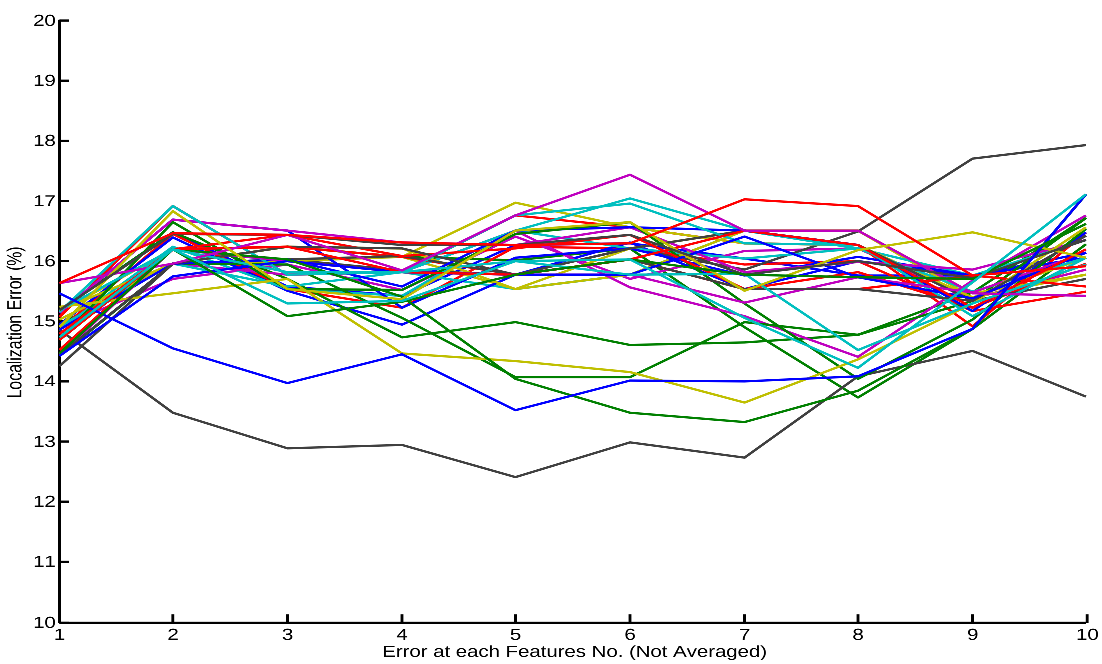


Figure 14: Localization error evaluated independently at each feature, Zero-Crossing interpolation disabled (15 KHz sampling rate)

noise sources especially if the environment is not sufficiently quiet before each event, a requirement for our current implementation. Fortunately only a few events exhibit this behavior. The majority of events remain predictable. The error rates also remained relatively unaffected with different source locations. This was true as long as the sound source was not too close to a wall or corners and within clear audible range of all nodes. Because of this we limited our testing to one source location as this allowed use to perform large number of tests repeatedly. The variations in error are less noticeable with increasing sampling rates. This we believe is due to reducing phase error with increasing sampling rate. The expected result in this is that increasing sampling rates increases stability due to increasing TOA precision. The accuracy however, is likely system limited, therefore no significant changes are visible.

5.1.2 Localization with Independent TOA, Zero-Crossing Interpolation Enabled

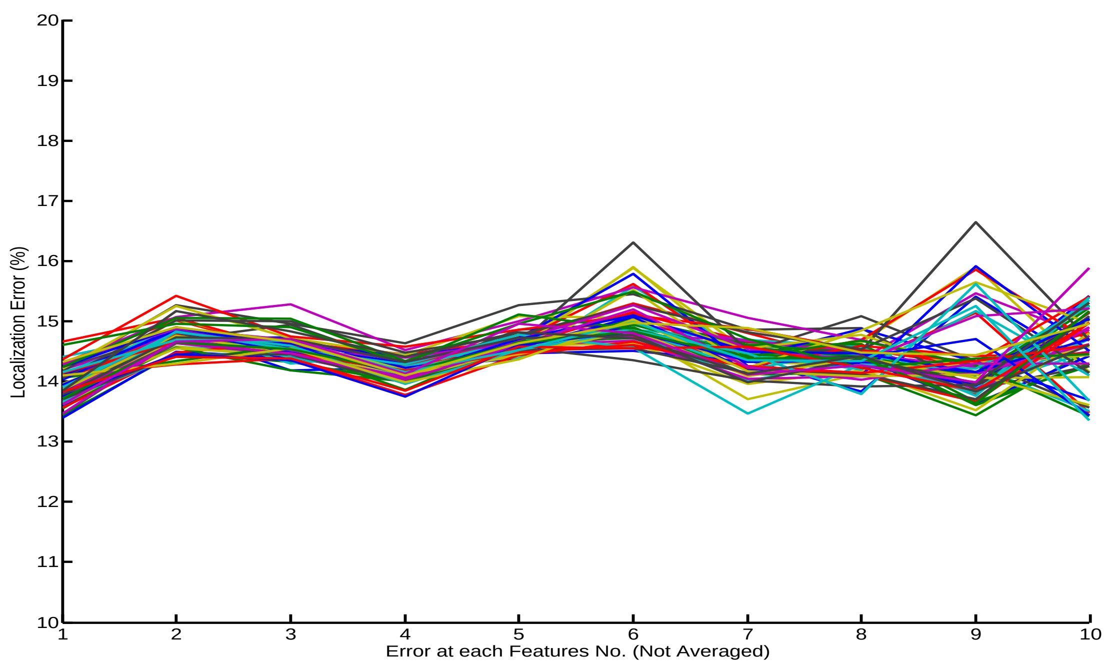


Figure 15: Localization error evaluated independently at each feature, Zero-Crossing interpolation enabled (5 KHz sampling rate)

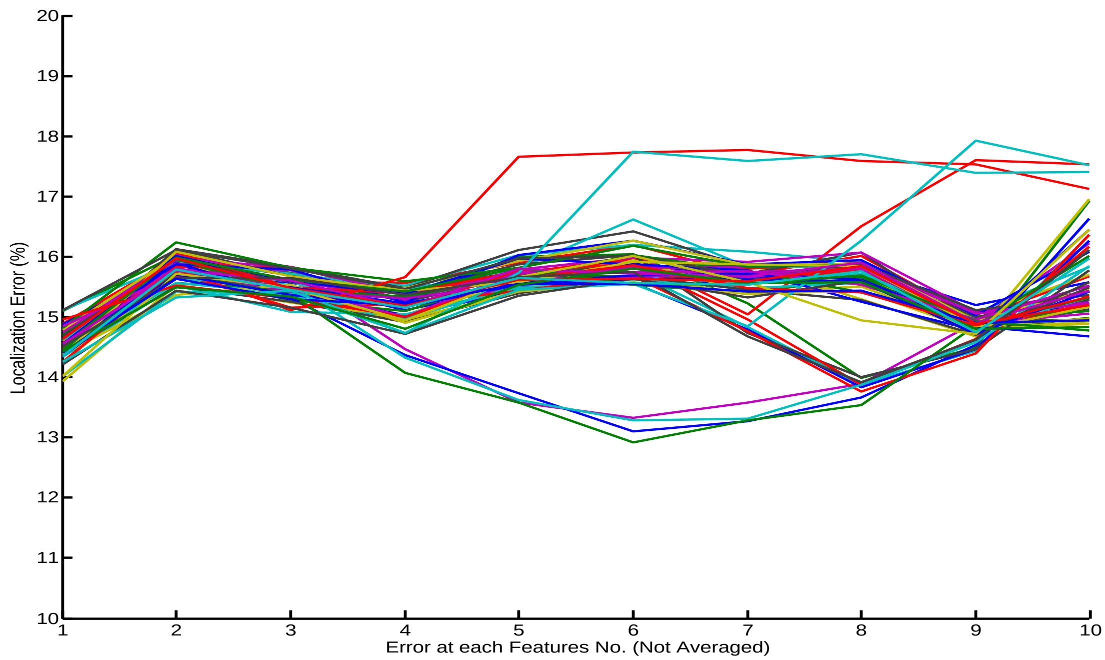


Figure 16: Localization error evaluated independently at each feature, Zero-Crossing interpolation enabled (10 KHz sampling rate)

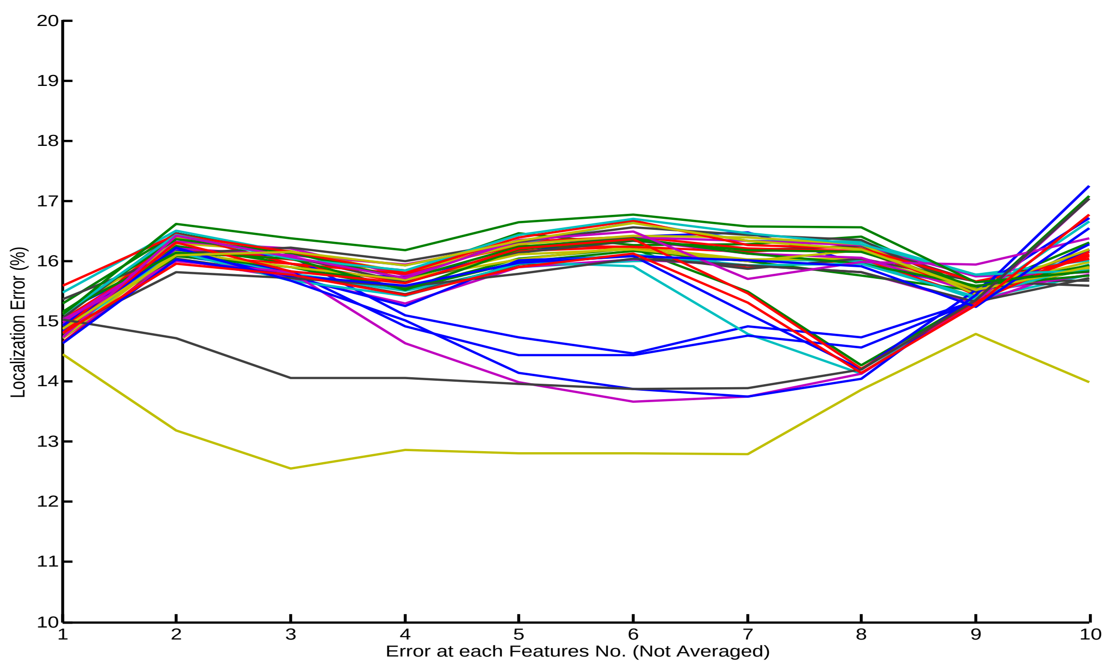


Figure 17: Localization error evaluated independently at each feature, Zero-Crossing interpolation enabled (15 KHz sampling rate)

Similarly, Figure 15 - Figure 17 shows the localization error at 5 KHZ, 10 KHz and 15 KHz at each matched TOA, this time with zero-crossing interpolation enabled. It is clearly visible here that events at all sampling rates now exhibit higher precision similar to the 15 KHz case without zero-crossing interpolation. The greatest difference can be seen with the 5 KHz case as it had the lowest precision when not using zero-crossing interpolation. The accuracy however does not improve noticeably in all three cases. This again leads us to believe that the accuracy is limited by the overall system. One of the simplicity offer by zero-crossing interpolation is that sub-sample accuracy can be obtained with minimal resource. To the best of our knowledge, ours is the first work to make use of zero-crossing interpolation for acoustic source localization.

5.1.3 Localization with Cumulatively Averaged TOAs, Zero-Crossing Interpolation Disabled

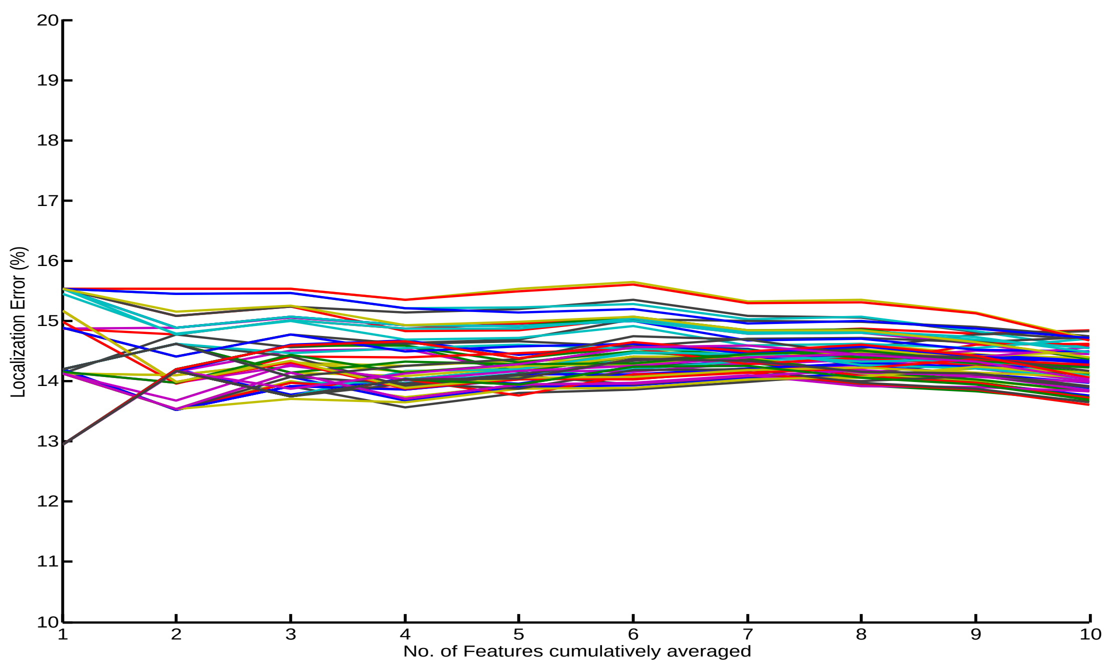


Figure 18: Localization error cumulatively averaged at each feature, Zero-Crossing interpolation disabled (5 KHz sampling rate)

Figure 18 - Figure 20 shows the localization error at 5 KHZ, 10 KHz and

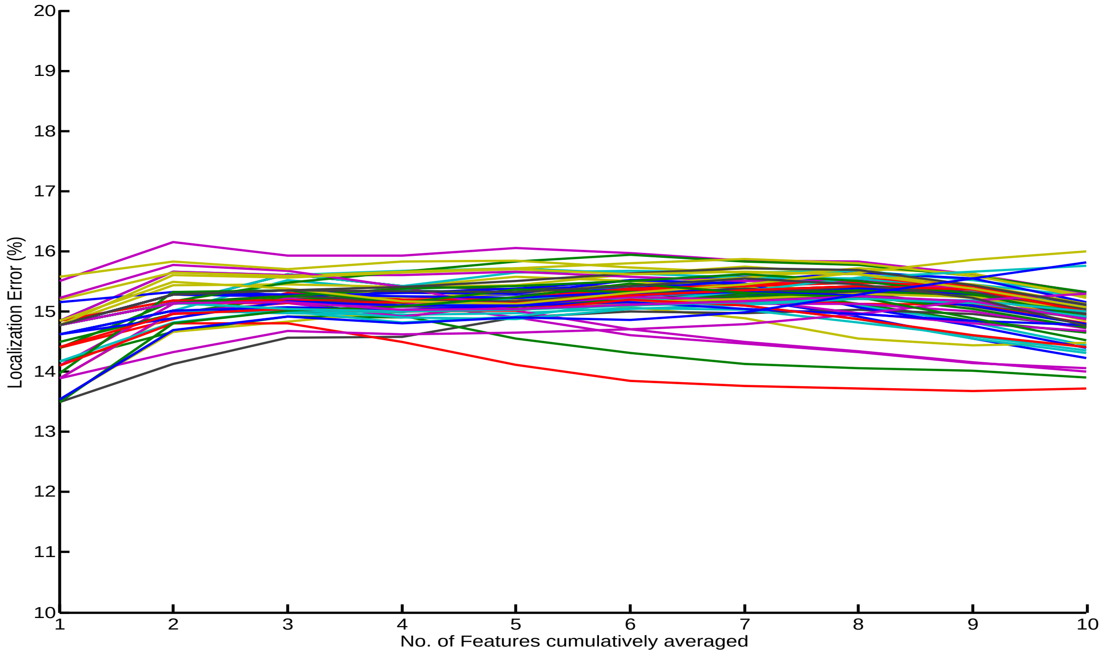


Figure 19: Localization error cumulatively averaged at each feature, Zero-Crossing interpolation disabled (10 KHz sampling rate)

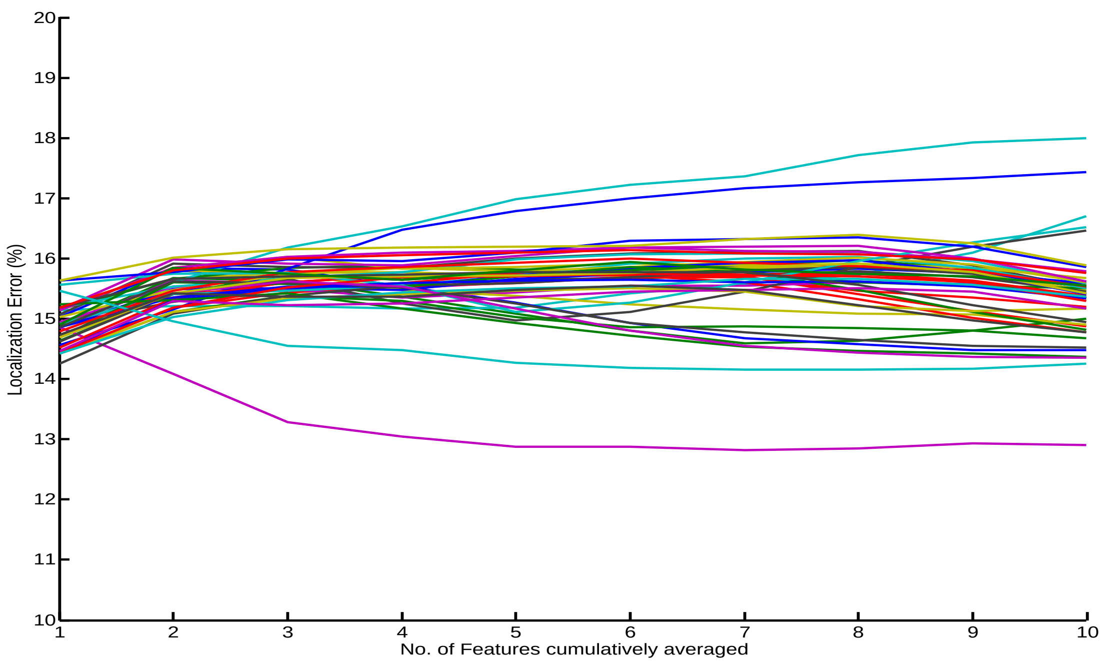


Figure 20: Localization error cumulatively averaged at each feature, Zero-Crossing interpolation disabled (15 KHz sampling rate)

15 KHz at each matched TOA, this time with zero-crossing interpolation disabled. However this time, we perform a cumulative average of the TOA with each additional feature known at that point. That is, for example, feature two is now an average of feature one and two. Here we can see an increase in stability after the fist few TOAs are averaged. The final TOA obtained although there is no significant change in accuracy, has a highly predictable error in most cases.

5.1.4 Localization with Cumulatively Averaged TOA, Zero-Crossing Interpolation Enabled

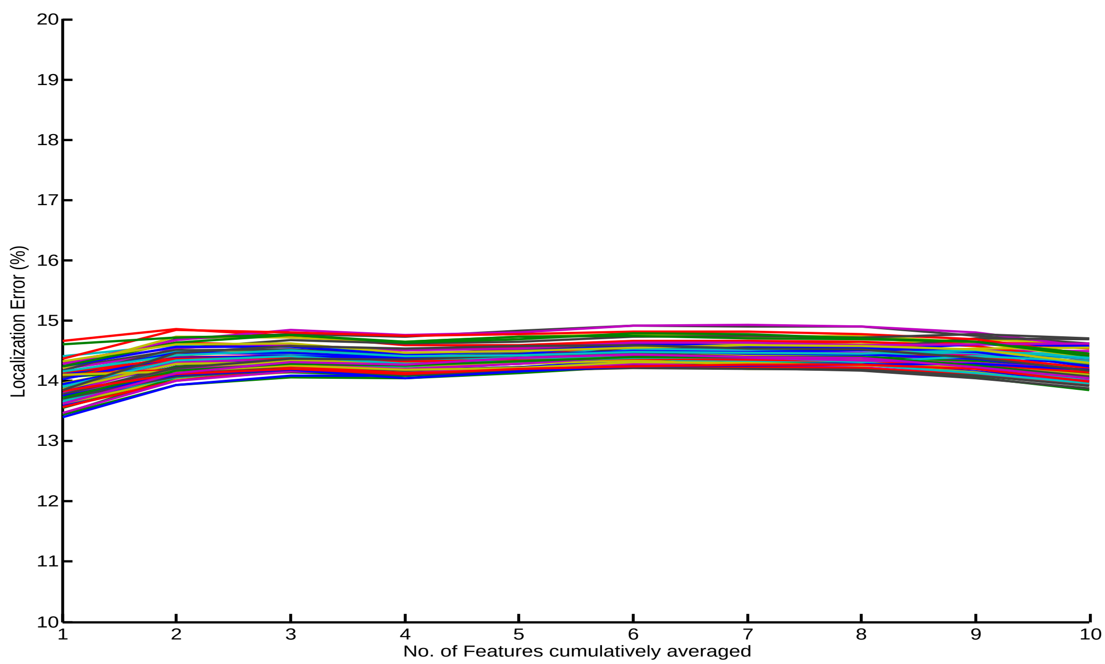


Figure 21: Localization error at each feature independently, Zero-Crossing interpolation enabled (5 KHz sampling rate)

Finally we combine cumulative averaging of TOA and zero-crossing interpolation for the best possible results. Figure 21 - Figure 23 shows the localization error at 5 KHZ, 10 KHz and 15 KHz. Here we see significant smoothing in all cases indicative of stability. The error rates are slightly higher for the case of 10 KHz and 15 KHz compared to the 5 KHz case. This is counterintuitive. However, we believe this is

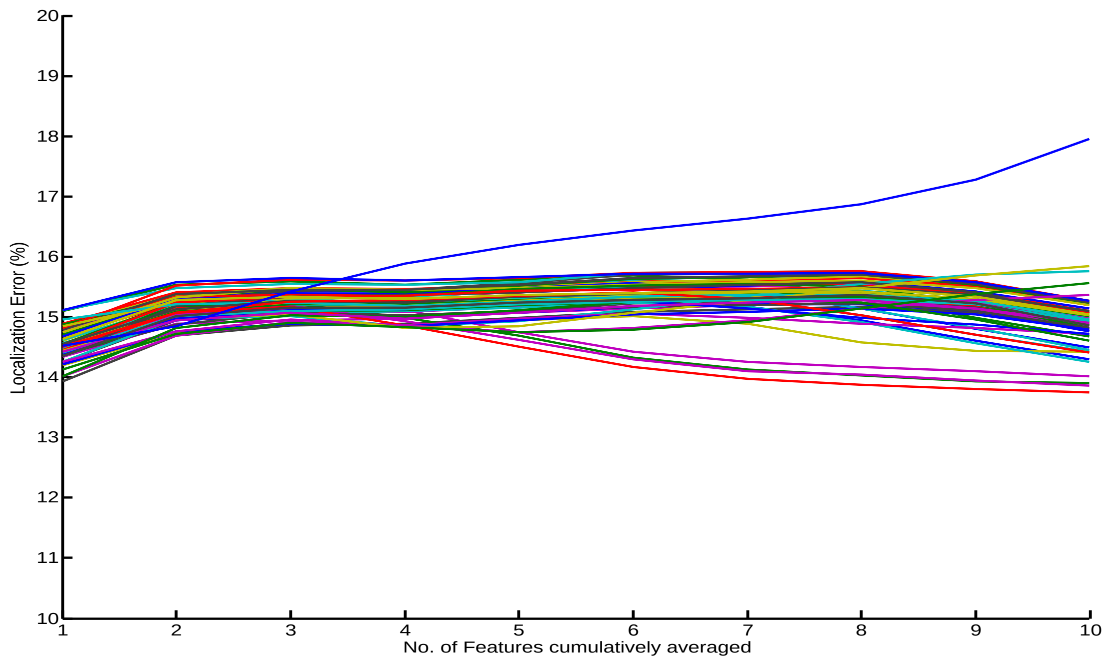


Figure 22: Localization error at each feature independently, Zero-Crossing interpolation enabled (10 KHz sampling rate)

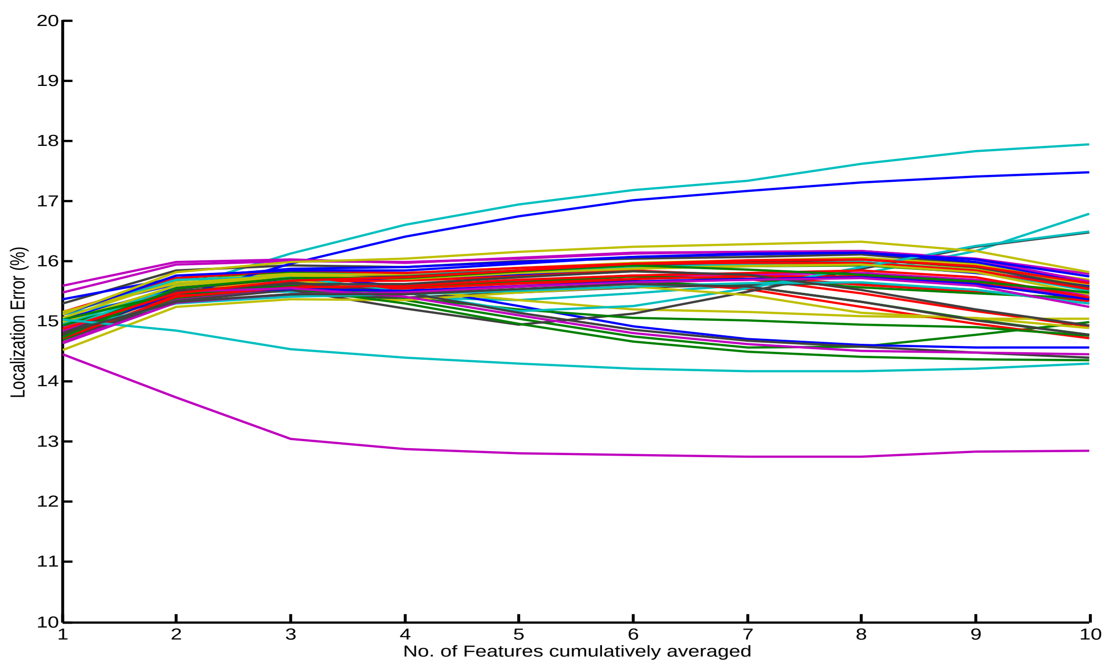


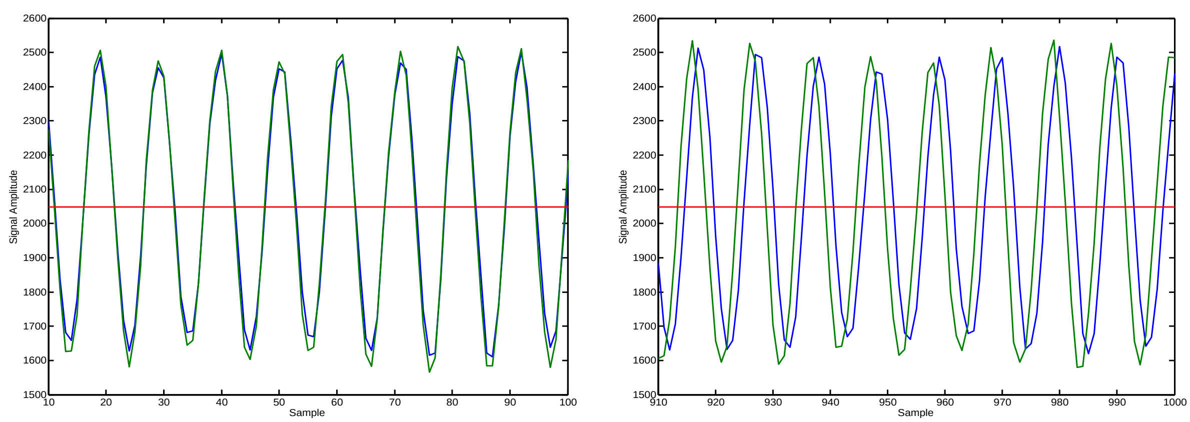
Figure 23: Localization error at each feature independently, Zero-Crossing interpolation enabled (15 KHz sampling rate)

due to increasing noise at higher frequencies which are see only at higher sampling rates. There are many sources of noise, more specifically the analog hardware and persistent sounds in the environment.

5.2 Problems Encountered

5.2.1 Clock drift

One of the large problems in wireless sensor networks is performing time-synchronization efficiently. Clock drifts at each node will occur and re-synchronization must be performed on a regular basis. Because of our use of wired synchronization, at least at the beginning of time-synchronization, we were able to alleviate the problem. However because we are sampling the acoustic signal for a significant length of time in the order of 100s of milliseconds, we face clock drifts. Even though the sampling process is started almost simultaneously as it should be on all nodes, the phase shifts are significant at the end of the sampling cycle. This problem is illustrated in Figure 24.



(a) Phase shift at start of sampling cycle

(b) Phase shift at end of sampling cycle

Figure 24: Clock drift resulting in sampling phase shift between nodes

In order to solve this problem we count the number of synchronization wait cycles just after sampling ends. This is done by restarting the time synchronization process for a second time, this time to get the time each node had to wait after sampling. This wait cycle count is used by the nodes for clock drift calculation.

5.2.2 Reflections

Perhaps the greatest challenges faced by acoustic localization systems is due to reflections. Reflections manifest in the form of reverberations and echoes. The reflection problems are largely minimized in a large outdoor setting where the source and receivers are in direct line of sight with not reflective object present. However this is not true in most cases and definitely not in indoor environments. Indoor settings or enclosed areas are particularly problematic for acoustic localization systems due to reflections. Some solutions exist such as the generalized cross-correlation using phase transformation (GCC-PHAT) for handling sounds in indoor settings. However this process as described before in section 3.1.2, is prohibitive for low power, large WSNs. Besides, GCC-PHAT is useful only for non-stationary sounds such as human speech. Our solution is to look for significant changes in amplitudes and frequencies as they represent new acoustic events. These new events at least at the beginning of the process, would be free of reflections. Figure 25 gives an example of our ten wave pulse where only the first 10 waves are new and rest being mostly due to reflections.

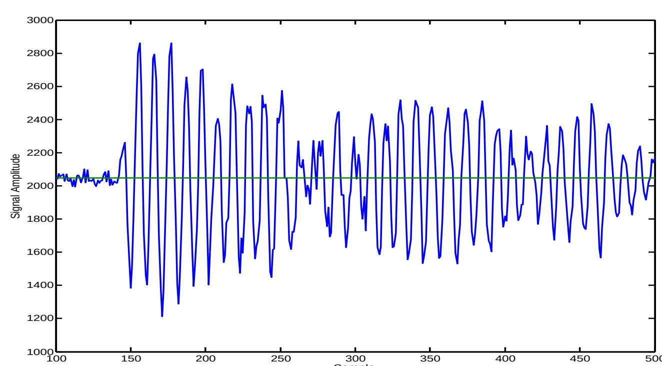


Figure 25: A sound pulse of 10 wave-lengths, beyond which reflections are significant.

CHAPTER VI

Related Work

Because of the severe restrictions placed by acoustic localization systems on wireless sensor network, there are only a limited number of ways the problem has been approached in. Acoustic localization as discussed before requires accurate time synchronization and can be computation and communication intensive. Most current acoustic localization systems based on WSNs can be separated based on four common approaches. Localization by time of arrival(TOA) of an impulsive signal and then taking the time differences, another way is to take the time differences of arrival (TDOA) directly, with the latter being much more resource intensive, by fusing multiple Direction Of Arrivals (DOA) and finally by signal strength. Signal strength methods will not be discussed as their limitation are greater than the other methods, in general.

6.1 Computing TDOA From Impulsive TOA

There are several works that have performed localization by using impulsive sound sources and then generating TOAs and TDOA from them. Vanderbilt's sniper localization systems are some such examples.

Simon et al. developed PinPtr which is a sniper localization system using a specialized acoustic WSN [4]. Unlike previous sniper localization systems which were centralized, PinPtr is a distributed system by relying on relatively inexpensive wireless sensor nodes that greatly increase robustness localization system. PinPtr was designed on a Mica2 sensor node platform which has an Atmel Atmega 128L microcontroller. The microcontroller onboard the Mica2 though a very efficient processor for basic processing in sensor network applications, it is nowhere near capable enough for PinPtr's acoustic localization processing. Therefore for PinPtr the team developed an FPGA based add-on board to perform the signal processing. A Xilinx Sparta II was used with three 1 MHz ADCs to sample the acoustic medium, although only one was used. Although most audible sounds that travel significant distances occur at frequencies well within the human audible range, PinPtr samples the signal at 1 MHz. Significantly oversampling the signal results in highly accurate phase differences which directly translates to better localization accuracy. PinPtr localizes gunshots by using time of arrivals of the gunshot along with the bullet's shockwave if available and sends the time information to the base-station. The base-station then fuses the TOAs to perform the localization.

Volgyesi et al. similar to PinPtr developed a gunshot localization and classification system using a combination of soldier mounted sensors and networked fusion of sensor data [23]. This system begins with soldier mounted sensor nodes containing a Xilinx FPGA with four acoustic channels. Four microphones placed at specific locations on a soldier's helmet are connected to four 1 MHz ADCs which provide the data to the FPGA. Each soldier mounted node is capable of determining TOA and possibly angle of arrival(AOA) of the bullet. This data is then sent to the base-station for fusion. Since the nodes are mobile, they have an onboard 3-axis magnetometer which is used as a compass for orientation. The base-station combines the TOAs or if available AOA from multiple nodes to compute the shooter's location. However, what is more relevant to our work is that this mobile sniper localization system makes use of on-board pattern matching to determine the TOA of the bullet's shockwave. Both this work and PinPtr continuously perform state machine analysis to look for a "N-wave" which is characteristic of the acoustic signal caused by a bullet's shockwave as it passes by. This pattern matching although efficient, works only for a highly predictable pattern, in this case a bullet's shockwave. In our thesis we propose methods to take this further by dynamically recognizing and matching patterns for sounds in general making it much less application specific.

Because of the simplicity in computing TOAs by monitoring significant and rapid changes in signal, most low power acoustic WSNs rely on thresholding methods for TOA determination. Na et al. developed a parking lot surveillance system making use of low power WSN motes unlike Vanderbilt's sniper localization systems [11] which demands a more capable network. TelosB motes were used as the node in the WSN and for acquiring acoustic data. In order for such a low power microcontroller such as the onboard MSP430 microcontroller to acquire acoustic signals and process them simple time domain based thresholding methods are preferred due to their simplicity. The work makes use of a dynamic thresholding algorithm for acquiring the TOA of car alarms in a parking lot. These TOAs are sent to the base-station where the localization is performed using the TOAs.

Guo et al. also make use of dynamic thresholding to perform indoor sound localization although other algorithms are also used in their work [33]. The largest limitation posed by thresholding algorithms is that a significant change in the signal magnitude is required. In addition, only a few sounds or acoustic events are highly impulsive. These limitations mean impulse based TOA localization systems are limited in their application.

6.2 TDOA By Cross-Correlation

Only a few works in WSNs resort to cross-correlation methods. This is primarily because of the expensive nature of cross-correlation methods. First, a significant portion of a signal needs to be captured and transmitted by radio. In a large multi-hop network this can be prohibitive in both network lifetimes and radio communication restrictions. In addition a powerful base-station is needed just to extract the TDOAs from the signals before localization is performed. While this is insignificant for desktop class computers, in large sensor networks, base-stations or stargate nodes are just more powerful embedded computers. Despite this, some applications that involve localizing non-stationary sounds such as human speech are difficult to performed adequately by impulse seeking methods.

Guo et al. as mentioned before make use of dynamic thresholding for impulsive sounds [33]. In addition they also use generalized cross correlation when less impulsive sounds are detected. Their implementation consists of an additional classification state unlike most other works. In this classification stage a significant sound is classified by using various metrics to determine whether the sound is impulsive or repetitive. If the sound is more repetitive such as human speech, the sound is transmitted in whole to all nodes for TDOA computation. These TDOAs are transmitted to the base-station for localization. Should the sound be impulsive, the TOA is computed immediately on the node and then the resulting time transmitted. In this work they make use of six Intel Imote2 WSN motes which perform the classification on each node. Although their approach is applicable to a very broad class of sounds, the computation and communication expensive nature of their algorithm which directly resulting in more expensive nodes can restrict large deployments.

6.3 Other methods

While TOA methods are generally considered efficient, there are localization systems based on much more expensive techniques such as beam forming and spectral estimation. Beamforming for example can be used to determine the direction of arrival of on or more source simultaneously. One such work is the acoustic ENSBox.

Girod et al. developed the acoustic ENSBox which was designed for rapid deployment of acoustic sensing nodes for localizing various acoustic events [13]. ENSbox relies on beamforming for determining direction of arrivals (DOA) of the the acoustic source. Multiple DOAs from distinctly placed nodes when intersected, represent a source. Beamforming however is an expensive operation and as a result each ENSBox is a relatively expensive piece of hardware for a WSN node. The ENSBox has a 400

MHz Intel PXA255 processor for performing the beamforming to determine the DOA of the source. The hardware and power requirements of the ENSBox makes this type of WSN very costly to deploy on a large scale. Also the latest work making use of the ENSBox has an additional useful feature, self localization service for determining each node's location which interestingly is performed by TDOA.

6.4 Similarities to our Method

Simply put our work does not introduce any grandly new theory that existing work already does not. However to the best of our knowledge no work exists that make use of the simplifications we make for the purpose of localization. So far none of the current work for acoustic sensor networks exclusively seek the beginning of significant acoustic source signal events in addition to pattern matching for the purpose of avoiding reflections. We later discovered that animal hearing including human hearing is also based on the same principle of seeking sudden changes in the sound either in intensity of frequency of a new acoustic event. Vanderbilt's sniper localization systems introduced us to the possibility of using pattern matching and recognition of the acoustic signal in time domain. In this our work only adds to the idea by making the pattern recognition generalized unlike the work being specifically for gunshot localization. In this we are invariably trying to emulate biological hearing. Unlike biological hearing however which are limited to two detectors and as a result can only determine directions of sounds by time differences, we can make use of multiple detectors for localization.

CHAPTER VII

Conclusion

We present our work only as a extension to current work for acoustic source localization in wireless sensor networks. By combining several ideas for existing research work and biological systems we explore the possibility of deploying large scale surveillance networks that are both energy efficient and inexpensive to construct. While we only tested the core parts of our overall idea, from the data we acquired we believe such large scale acoustic wireless sensor networks are feasible.

Our results support two important ideas we developed in this work. First, multiple features for an acoustic event could be separated and used for localization with minimal resources, in addition to improving the precision and possibly accuracy when features are combined. Second, we demonstrate the applicability of zero-crossing interpolation for increased localization precision and principle of selectively avoiding reflections, something our ears perform on a daily basis. In particular, the stability of localization in all cases shows us that reflections can be avoided.

7.1 Future Work

Deploying a real-time generic-sound low-power wireless acoustic localization network would be the ultimate goal of this work. Despite the work presented in this thesis, a few additional components need to be in place before such a system can exists. A proper pattern identification and matching technique has to be implemented and tested. Our next step would be to expand the network to allow larger deployments. This would also mean the network should be able to localize a source with only a small group of nodes that detect a sound. Additional requirements that we place include real-time conformance and enabling a truly wireless sensor network. For this we would have to employ one of the existing time-synchronization protocols such as FTSP. Luckily, code for such existing algorithms are already present and freely available. Having these components in place would allow this network to be highly distributed and inexpensive for a large deployment.

7.2 Scale Of Applications

Our purpose for the work we presented is so far limited to testing important ideas for feasibility. In the future we could also build applications more specific to a particular class of applications. For example, gunshots could possibly be localized using our algorithm even though present WSNs for gunshot localizations make use of powerful hardware and extremely high sampling rates. By using feature recognition/-matching and zero crossing interpolation we can achieve to a certain extent similar benefits as high sampling rates provide without expensive hardware. In addition due to the basic requirements of our algorithm, miniature nodes can be deployed in large scales, opening the possibility for long term surveillance networks. Shopping complexes, cities, even battlefields could be monitored with a large network that perform most of the processing locally. Such networks would also be robust to failure of nodes in the network in addition to increased accuracy provided. Systems great scale are only possible if efficiency is a core ideal. The greater good that distributed, long lasting, surveillance networks for the propagation and maintenance of peace would be of great benefit to humanity.

BIBLIOGRAPHY

[1] BRUMBARCHRIS. Mems microphones to replace electret types. `http://dev.emcelettronica.com/mems-microphones-to-replace-electret-types`, May 2009.

[2] C. PANAGIOTAKIS, AND G. TZIRITAS. A speech/music discriminator based on rms and zero-crossings. In *IEEE Transactions on Multimedia, 2005* (February 2005), IEEE.

[3] C.H. KNAPP, AND G.C. CARTER. The generalized correlation method for estimation of time delay. In *IEEE Transactions on Acoustics, Speech and Signal Processing. ASSP* (Aug 1976), vol. 24, IEEE, pp. 320–327.

[4] G. SIMON, M. MARTI, AND A. LDECZI. Sensor network-based countersniper system. In *The Second ACM Conference on Embedded Networked Sensor Systems, SenSys* (November 2004), ACM.

[5] G. VALENZISE, L. GEROSA, M. TAGLIASACCHI, F. ANTONACCI, AND A. SARTI. Scream and gunshot detection and localization for audio-surveillance systems. In *IEEE Conference on Advanced Video and Signal Based Surveillance. AVSS* (January 2007), IEEE.

[6] H. SHAU, AND P.A. ROBINSON. Passive source localization employing intersecting spherical surfaces from time-of-arrival differences. In *IEEE Transactions on Acoustics, Speech and Signal Processing* (Aug 1987), IEEE.

[7] I.F. AKYILDIZ, W. SU, Y. SANKARASUBRAMANIAM, AND E. CAYIRCI. A survey on sensor networks. In *Communications Magazine* (August 2002), IEEE, pp. 102 – 114.

[8] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the 5th symposium on Operating systems design and implementation. OSDI* (2002), ACM.

[9] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems. SenSys* (2004), ACM.

[10] J.C. Chen, Y. Kung, and R.E. Hudson. Source localization and beamforming. *Signal Processing Magazine 19*, 2 (March 2002), 30–39.

[11] K. Na, Y. Kim, and H. Cha. Acoustic sensor network-based parking lot surveillance system. In *Proceedings of the 6th European Conference on Wireless Sensor Networks, EWSN* (2009), ACM.

[12] Knowles Acoustics. Knowles acoustics spm0408he5h mems microphone datasheet. `http://www.knowles.com/search/prods_pdf/SPM0408HE5H.pdf`, August 2009.

[13] L. Girod, M. Lukac, V. Trifa, and D. Estrin. The design and implementation of a self-calibrating distributed acoustic sensing platform. In *Proceedings of the 4th international conference on Embedded networked sensor systems. SenSys* (2006), ACM.

[14] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling. Imote2: Serious computation at the edge. In *International Conference on Wireless Communications and Mobile Computing. IWCMC* (August 2008), IEEE.

[15] L.A. Jeffress. A place theory of sound localization. *Journal of comparative and physiological psychology* (1948).

[16] M. Azizyan, I. Constandache, and R.R. Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking. MobiCom* (2009), ACM.

[17] M. Marti, B. Kusy, G. Simon, and A. Ldeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems. SenSys* (2004), ACM.

[18] Moteiv Corporation (Sentilla). tmote sky datasheet. `http://www.sentilla.com/files/pdf/eol/tmote-sky-datasheet.pdf`, November 2006.

[19] N. Patwari, J.N. Ash, S. Kyperountas, A.O. III Hero, and N.S. Correal. Locating the nodes: Cooperative localization in wireless sensor networks. *Signal Processing Magazine 2*, 4 (June 2005), 54–69.

[20] Nirupama Bulusu, and Sanjay Jha. *Wireless sensor networks.* Artech House, 2005.

[21] P. Dutta, J. Taneja, J. Jeong, X. Jiang, and D. Culler. A building block approach to sensornet systems. In *ACM Conference on Embedded Networked Sensor Systems. SenSys* (2008), ACM.

[22] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. Tinyos: An operating system for sensor networks. In *The Second ACM Conference on Embedded Networked Sensor Systems. SenSys* (November 2004), ACM.

[23] P. Volgyesi, G. Balogh, A. Nadas, C.B. Nash, and A. Ldeczi. Shooter localization and weapon classification with soldier-wearable networked sensors. In *Proceedings of the 5th international conference on Mobile systems, applications and services. MobiSys* (2007), ACM.

[24] P.G. Reddy, and N. Sridhar. Lakon: A middle-ground approach to high-frequency data acquisition and in-network processing in sensor networks. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks. IPSN SPOTS* (April 2010), ACM/IEEE.

[25] R. Bucher, and D. Misra. A synthesizable vhdl model of the exact solution for three-dimensional hyperbolic positioning system. *VLSI Design 15*, 2 (August 2002), 507–520.

[26] R.K. Rana, C.T. Chou, S.S Kanhere, N. Bulusu, and W. Hu. Ear-phone: an end-to-end participatory urban noise mapping system. In *Information Processing in Sensor Networks. IPSN* (November 2010), ACM.

[27] S. Ganeriwal, R. Kumar, and M.B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems. SenSys* (2003), ACM.

[28] ST Microelectronics. St microelectronics lmv324 datasheet. `http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00079372.pdf`, January 2010.

[29] Texas Instruments. Texas instruments msp430 microcontroller (msp430f1611). `http://www.ti.com/lit/ds/symlink/msp430f1611.pdf`, October 2002.

[30] TinyOS. Tinyos website. `http://www.tinyos.net/`, Dec 2011.

[31] W. Hanbiao, J. Elson, L. Girod, and D. Estrin. Target classification and localization in habitat monitoring. In *International Conference on Acoustics, Speech, and Signal Processing. ICASSP* (April 2003), IEEE.

[32] W. Hu, V.N. Tran, N. Bulusu, C.T. Chou, S. Jha, and A. Taylor. The design and evaluation of a hybrid sensor network for cane-toad monitoring. In *Information Processing in Sensor Networks. IPSN* (April 2005), IEEE.

[33] Y. Guo, and M. Hazas. Localising speech, footsteps and other sounds using resource-constrained devices. In *The 10th International Conference on Information Processing in Sensor Networks. IPSN* (April 2011), IEEE.