Improved Sound-based Localization Through a Network of Reconfigurable Mixed-Signal Nodes

A Thesis Presented
by

Anurag Umbarkar

to

The Graduate School
in partial fulfillment of the
Requirements
for the degree of

Master of Science
in
Computer Engineering

Stony Brook University
August 2010

Stony Brook University
The Graduate School
Anurag Umbarkar
We, the thesis committee for the above candidate for the
Master of Science degree,
hereby recommend acceptance of this thesis.

Dr. Alex Doboli, Advisor of Thesis
Associate Professor, Department of Electrical and Computer Engineering

Dr. Sangjin Hong, Associate Professor,
Department of Electrical and Computer Engineering

This thesis is accepted by the Graduate School.

Lawrence Martin
Dean of the Graduate School

Abstract of the Thesis

Improved Sound-based Localization Through a Network of Reconfigurable Mixed-Signal Nodes

by

Anurag Umbarkar

Master of Science

in

Computer Engineering

Stony Brook University

2010

There have been extensive theoretical studies on sound-based localization using pairs of microphones as well as microphone arrays. In contrast, there has been much less work on implementing and experimenting sound-based localization realized as customized electronic designs. This thesis presents a low cost implementation of the phase-based sound localization method proposed in Halupka et al [2]. The implementation uses PSoC programmable mixed-signal embedded System on Chip, which incorporates microcontroller, on-chip SRAM and flash memory, programmable digital blocks and programmable analog blocks, all integrated on the same chip. The report presents a set of experiments to characterize the quality of localization using the proposed low-cost design.

In addition, the thesis suggests a modification in the digital signal processing part through which Maximum Likelihood is replaced by an alternative method. The results for both these methods are then compared on the basis of accuracy, memory requirement and execution time. In order to improve the localization accuracy, filter corner frequency reconfiguration and gain reconfiguration is implemented. A wireless sensor network implementation is also presented. An extensive set of experiments are provided to explore the advantages of dynamic reconfigurability as well as the network implementation.

To My Parents

Table of Contents

List of Figures

ACKNOWLEDGEMENTS

First of all, I would like to thank Dr. Alex Doboli for his continued support and encouragement throughout the course of this thesis project. He has also provided valuable inputs for this thesis report. His keen interest and enthusiasm towards research inspired me a great deal. From the numerous discussions we have had, he has taught me to think like an Engineer. I will always remember what he wrote in an email once: 'Research is not a burden, its for fun'. I consider it a privilege to have Prof. Doboli as my advisor.

I would also like to thank my colleagues from the VLSI Systems Designs Laboratory (VSD Lab): Varun Subramanian, Cristian Ferent, Mike Gilberti, Pengbo Sun and Meng Wang for the help they have provided. Varun has been a good friend and has provided me with valuable advice on technical as well as non-technical matters. He has helped me a lot not only during this project, but ever since I came to Stony Brook as a new student. I would like to extend special thanks to Cristian for taking time out from his busy schedule and spending hours discussing ideas to debug my implementation.

My parents have always supported me in all the career choices that I have made so far. I would like to thank them for all the love and support. My sincere thanks to my sister and brother-in-law, for their valuable guidance and for helping me at each step since my arrival in the US. I would also like to thank my brother and sister-in-law for helping me make important decisions and for encouraging me to do my best.

Chapter 1

Introduction

Sound localization is the process of identifying the spatial coordinates of a sound source based on the sound signals received by a microphone array [2]. Sound localization is required for various applications such as security systems, surveillance systems, video conferencing, robot navigation and speech recognition [1, 3, 4, 6]. Sound enhancement and speech separation techniques are usually employed to achieve high precision localization. This has helped in the evolution of voice-activated portable electronics. Sound localization can be achieved using one or more microphones in the microphone array. This paper presents an implementation using two microphones.

The experiments in [2] detail the fact that the localization accuracy decreases with increase in the ambient noise. Also, temperature dependency of the speed of sound in air, introduces additional abnormality in the results. The dynamic reconfigurability aspect of PSoC can be used to overcome these limitations faced by implementations which use FPGAs or ASICs. PSoC is a programmable, mixed-signal SoC that includes 8-bit microcontroller, on-chip

SRAM and flash memory, programmable digital blocks, and programmable analog blocks [7]. This makes PSoC a very attractive architecture for this application as it supports integrated implementation of the mixed-signal frontend for sound-based localization. The analog frontend of the design consists of signal conditioning, filtering, and analog to digital conversion (ADC). The digital processing includes Hanning windowing, Fast Fourier Transform (FFT), phase calculation and Maximum Likelihood (ML) algorithm [2].

The main advantage of the implementation in this paper is its low cost as compared to the much higher cost required to develop customized integrated circuits as in [2]. Moreover, after detecting a sound source, the sound characteristics, e.g. level, frequency bandwidth, and noise, can be used to dynamically customize the implementation by modifying the topology and parameters of the building blocks, like amplifiers and filters at run time. By dynamically adjusting the corner frequency of the filter and the total gain of the system, the Signal to Noise Ratio (SNR) can be improved. Also, the reconfigurable input MUXs can be used to integrate other designs on the same chip. For example, the analog front end used for sound localization is modified at runtime and reused for temperature sensing. Temperature sensing helps in improving the localization accuracy. Therefore, reconfiguration reduces the utilized resources, hardware blocks and energy, while satisfying the performance requirements of the application.

The algorithm proposed in [2] provides adequate localization accuracy only in a specific range of the Direction of Arrival (DOA) of the sound signal. Beyond the DOA value of 60 degrees, there is drastic rise in the percentage of incorrect readings. In order to mitigate this shortcoming, this paper proposes a wireless network implementation of localization nodes. By strategically deploying these nodes over the area under consideration, sufficient accuracy can be obtained with minimal redundancy.

The report is organized as follows: Chapter 1 offers an overview of the basics of sound localization as well as the maximum likelihood and alternative solution. Chapter 2 provides the details of the analog frontend and DSP core implemnted on PSoC1. Chapter 3 explains the design performance in terms of execution time and memory usage. Chapter 4 details the cutoff frequency and gain reconfiguration. Chapter 4 also explains modification of the existing design to accomodate and additional application for temperature sensing. Chapter 5 offers a wireless sensor network implementation for sound localization. Chapter 6 provides experimental results for maximum likelihood and the alternative method. It also provides results for reconfiguration and network implementation. Chapter 7 presents our conclusions and the future scope for this application.

1.1 Sound Localization Basics

A simple method of localization is to estimate the time delay of arrival (TDOA) of a sound signal between the two microphones. This TDOA estimate is then used to calculate the Angle of Arrival (AoA). Combining the data from two microphone pairs and by using the process of triangulation, we compute the distance of sound source from the microphone pairs as shown in Figure 1.

In Figure 1.1(a), TDOA= Y-X/v , where v is speed of sound and θ is the AoA. In Figure 1.1(b), 'dist' gives the perpendicular distance between the sound source and line joining the microphone pairs.

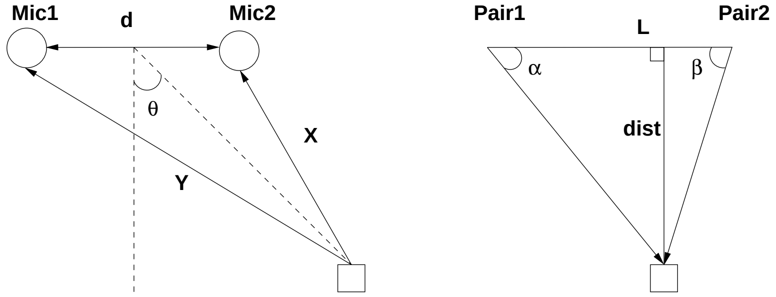


Figure 1.1: Sound Localization: (a) TDOA estimation (b) Process of triangulation

1.2 TDOA Estimation using GCC and PHAT

The most commonly used TDOA estimation method is generalized cross correlation (GCC). If we consider a single sound source placed in front of a microphone pair, the signals received will be of the format:

$$\begin{aligned} m1(t) &= s(t) + n1(t) \\ m2(t) &= s(t+\tau) + n2(t) \end{aligned} \tag{1.1}$$

where, $s(t)$ is the actual sound signal, $s(t+\tau)$ is the delayed version of the sound signal and $n1(t)$ and $n2(t)$ are the noises associated with the two microphones. These noise elements get mixed with the actual signal due to the ambient noise, reverberations and microphone-induced noise.

The Fourier transforms of the received signals are $M1(\omega)$ and $M2(\omega)$. The TDOA estimate $\tilde{\tau}$ can be calculated by applying the cross correlation equation as follows:

$$\tilde{\tau} = \arg\max_{\beta} \int_{\omega=-\infty}^{\infty} M1(\omega)\overline{M2(\omega)}e^{-j\omega\beta}d\omega \tag{1.2}$$

In order to ensure that the application works well in noisy environments, we apply the PHAT weighting function to the above equation. After simplification as explained in [2], we get the following equation:

$$\tilde{\tau} = \arg\max_{\beta} \sum_{n=0}^{N/2} cos(\angle M1(n) - \angle M2(n) - 2\pi F_s n\beta/N) \tag{1.3}$$

where, "$\angle$" denotes the phase of its respective argument, n is the index of the discrete time Fourier transform, N is the total number of samples and F_s is the sampling frequency. Thus, the TDOA estimate will be that value of β which results in the maximum sigma and hence maximum cross correlation. The Angle of Arrival is then calculated using the equation:

$$AoA = asin(\tau * v/d) \tag{1.4}$$

where, d is the inter-microphone spacing. The overall process can be summarized as shown in Figure 1.2.

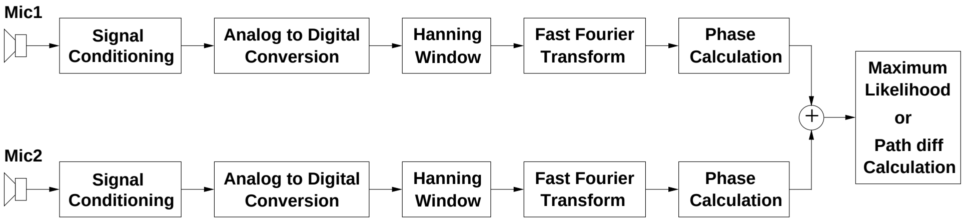


Figure 1.2: Sound Localization Data Flow

1.3 Alternative Solution

As proved in [2], the above mentioned maximum likelihood equation provides satisfactory results for low noise conditions. However, at high noise conditions, the percentage of abnormal readings goes beyond acceptable values. This is mainly because maximum likelihood is not a precise estimator. Hence, in this report we have proposed an alternative solution to improve accuracy of results. We identify the most significant component of the signal by observing the FFT output. Phase difference is then calculated only for this point. Then path difference and hence, TDOA is calculated using the formula:

$$\begin{aligned} Phasediff &= M1(peak) - M2(peak) \\ Pathdiff &= phasediff * \lambda/2\pi \\ TDOA &= pathdiff/v \end{aligned} \tag{1.5}$$

where, λ depends on the frequency of the most significant component. This frequency can be derived from the FFT.

Chapter 2

Implementation Overview

The entire implementation is first coded in software using matlab. This software implementation is run for matlab-generated ideal inputs and results are tabulated. The hardware implementation is done on PSoC1. The microphone circuitry is mounted on the breadboard of the PSoC Eval board. The analog modules are implemented using reconfigurable analog blocks inside the PSoC while the digital modules are implemented in c code.

2.1 Microphone Circuitry

The dc source used to power the microphones should not contain any analog components. Hence, we use a RC filter to make sure we get pure dc from the source. The microphone pair absorbs the sound signals and convert them to electrical signals. These signals are then passed through a dc-blocking capacitor and resistor divider network. These analog electrical signals need to be given a dc shift so that they lie in the operating range of PSoC. This is done by using a Programmable Gain Amplifier with gain=1 and input of 2.5 volts

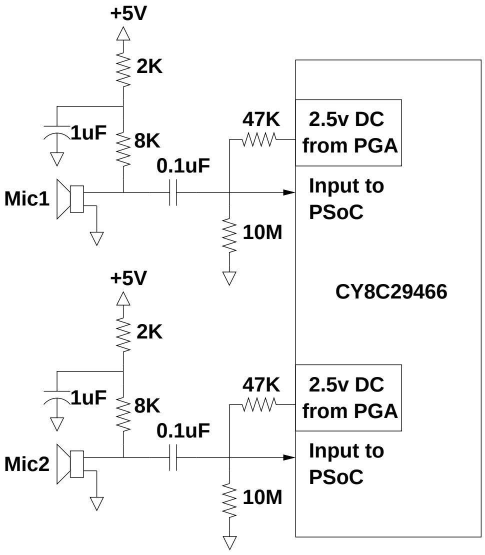


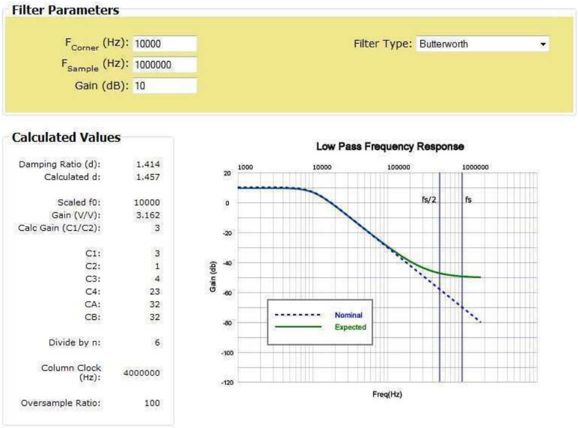
Figure 2.1: Microphone Circuitry

dc. The signal, before entering the PSoC, is mixed with this PGA output. This circuit is replicated on the bread board for each microphone. The circuit is as shown in Figure 2.1.

2.2 Analog Frontend

The output of the microphone circuitry is given to the analog amplifier. The gain of this amplifier is set to 24. After amplification, the signal is passed through a low pass filter. The filter characteristics are as shown in Fig 2.2.

The filter removes the high frequency noise and we get an improved clarity signal. The filter output is then passed through an Analog to Digital Converter (ADC), to convert the signal into discrete form. The ADC sampling rate is set at 13.33 KHz and the resolution is set as 7 bits/sample. PSoC provides the option of using a DUAL ADC which contains two ADCs which sample inputs



Filter Parameters

F_{Corner} (Hz): 10000

F_{Sample} (Hz): 1000000

Gain (dB): 10

Filter Type: Butterworth

Calculated Values

Damping Ratio (d):	1.414
Calculated d:	1.457
Scaled f0:	10000
Gain (V/V):	3.162
Calc Gain (C1/C2):	3
C1:	3
C2:	1
C3:	4
C4:	23
CA:	32
CB:	32
Divide by n:	6
Column Clock (Hz):	4000000
Oversample Ratio:	100

Figure 2.2: Filter Characteristics

simultaneously. This is perfect for sound localization application, since the signal at both microphones needs to be sampled simultaneously. The Analog and Digital block usage map inside the PSoC is as represented in Fig 2.3.

2.3 Digital Section

After the Analog Frontend, the signal is passed through the Digital section, which consists of Hanning window, Fast Fourier Transform and CORDIC phase calculation. This is followed by the TDOA estimation and AoA calculation.

2.3.1 DSP Core

We collect 128 samples from each channel of the DUALADC. This segment size is chosen as a power of two to facilitate the working of Fast Fourier Transform (FFT). Another constraint which affects the segment size, is the availability of on-chip memory. Since PSoC has only 2KB of internal memory, spread over 8

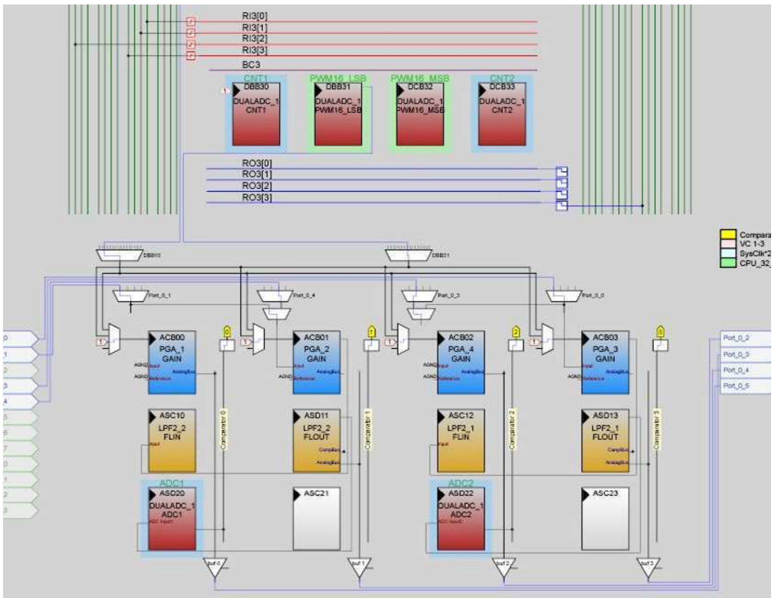


Figure 2.3: PSoC Digital and Analog Blocks

pages of 256 bytes each, we select a modest segment size of N=128. Increasing this size would increase density of FFT output and hence improve localization accuracy.

The samples from the ADC are passed through a Hanning window. The windowing function can be mathematically represented as follows:

$$w(n) = 0.5(1 - cos(2\pi n/(N-1))) \tag{2.1}$$

Diagrammatic representation of the window is as shown in Fig 2.4.

The window function is used to get a clear spectral response at the output of the FFT. This output is then provided as input to the FFT. The FFT is an N-point in-place radix-2 decimation-in-time (DIT) FFT. An in-place FFT is much faster than the Discrete Fourier Transform (DFT). DFT calculation

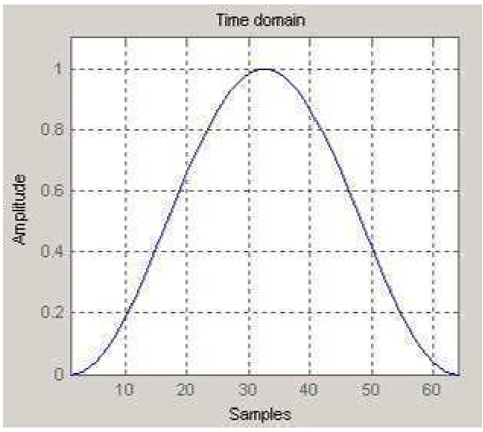


Figure 2.4: Hanning Window

requires $O(N^2)$ calculations, whereas the FFT algorithm requires $O(Nlog_2N)$ calculations. Typical butterfly structure for DIT FFT is as shown in Fig 2.5.

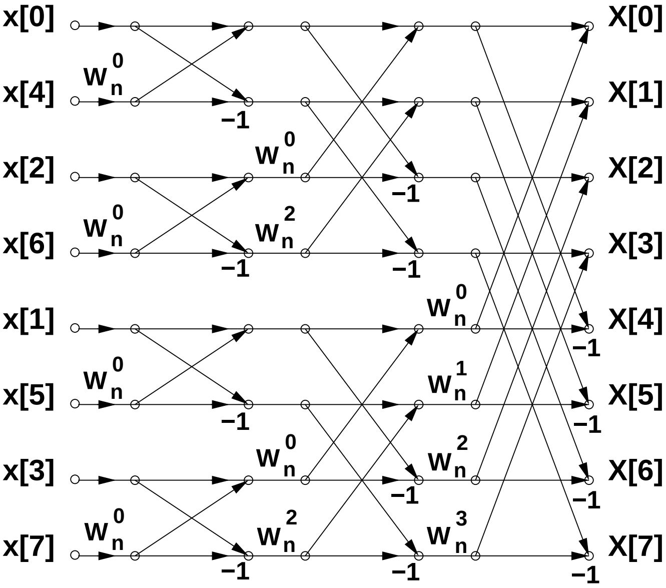


Figure 2.5: Radix-2 DIT FFT

2.3.2 Phase Calculation using CORDIC

CORDIC stands for Coordinate Rotating Digital Computer. CORDIC algorithm is used to convert from Cartesian to polar coordinates. It performs rotations of the Cartesian vector until the resulting vector is sufficiently close to x axis. By keeping track of the rotations, the original phase of the vector is approximated.

The CORDIC equations for phase calculation are as follows:

$$\begin{aligned} x_{i+1} &= x_i - y_i.d_i.2^{-i} \\ y_{i+1} &= y_i + x_i.d_i.2^{-i} \\ z_{i+1} &= z_i - d_i.tan^{-1}.2^{-i} \\ d_i &= {}^{+1, if y_i < 0}_{-1, otherwise} \end{aligned} \tag{2.2}$$

The variables x and y are the Cartesian coordinates of the vector undergoing rotation, and z is the sum of the rotations performed. The index i is the iteration counter used for the CORDIC algorithm. The pair $(x0, y0)$ are initialized to be the Cartesian coordinates of a vector, and $z0$ is initially 0. At each iteration, the algorithm makes a decision based on the current value of y_i as to which direction to rotate the vector in order to make it lie on the x-axis. After a fixed number of rotations, the phase of the vector is given by z, whereas the dilated magnitude given by x; $y$should be approximately zero by virtue of rotating the original vector onto the x-axis but has no defined meaning.

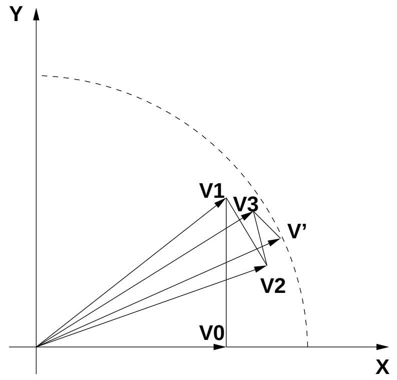


Figure 2.6: CORDIC phase estimation

CORDIC can also be used to calculate sine and cosine of an angle, by modifying the equations in the following manner:

$$
\begin{aligned}
x_{i+1} &= x_i - y_i.d_i.2^{-i} \\
y_{i+1} &= y_i + x_i.d_i.2^{-i} \\
z_{i+1} &= z_i - d_i.tan^{-1}.2^{-i} \\
d_i &= {}^{-1, if z_i<0}_{+1, otherwise}
\end{aligned}
\tag{2.3}
$$

Thus, CORDIC provides phase estimation, sine and cosine estimation in an iterative manner. This eliminates use of math.h library and hence reduces the code length and program memory required.

2.3.3 Angle of Arrival Calculation

Using maximum likelihood, TDOA estimate can be obtained as explained in the previous section. Angle of Arrival can then be derived by using the formula:

$$AoA = asin(\tau * v/Fs * d) \tag{2.4}$$

For the alternative method, we identify the most significant component of the signal by observing the FFT output. Phase difference is then calculated only for this point using wave-counting algorithm. Then path difference and hence, TDOA is calculated using the formula:

$$\begin{aligned} Phasediff &= M1(peak) - M2(peak) \\ Pathdiff &= phasediff * \lambda/2\pi \\ TDOA &= pathdiff/v \\ AoA &= asin(\tau * v/d) \end{aligned} \tag{2.5}$$

where, λ depends on the frequency of the most significant component. This frequency can be derived from the FFT.

2.4 Waveforms for PSoC Implementation

For the example in Figure 2.7, the frequency of the input audio tone is 1 KHz. There is a phase shift between the signals received at the two microphones. The Hanning window output for these signals is shown in Figure 2.8. The FFT output is as shown in Figure 2.9. Since the input is a single frequency tone, there is just one peak on the FFT at 1000 Hz. The peak at around 0 Hz corresponds to the DC component with which the input is offset. The FFT is symmetrical around the Nyquist frequency (Fs/2). This is because the inputs to the FFT are real values.

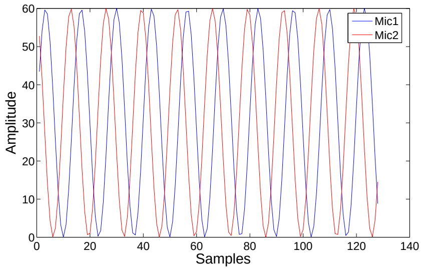


Figure 2.7: Input audio tone at mic1 and its phase-shifted version at mic2

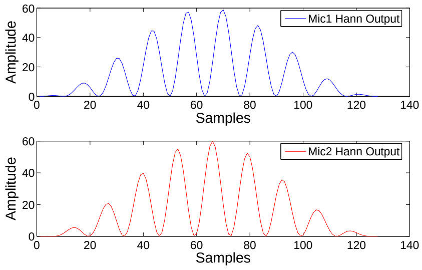


Figure 2.8: Hanning window output for mic1 and mic2 signals

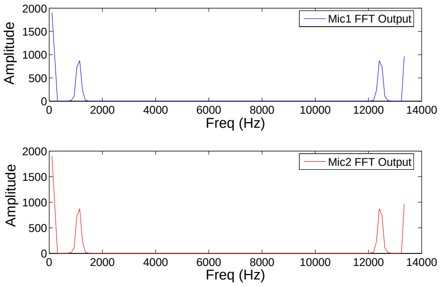


Figure 2.9: FFT output for mic1 and mic2 signals

Chapter 3

Resource Management

3.1 Design Performance for single PSoC

3.1.1 Execution Time and Memory Usage

The implementation of the sound-based localization system uses PSoC Family1 chip CY8C29466-24PXI. The following programmable resources of PSoC are used: four programmable digital blocks and two programmable analog blocks for the dual ADC, four programmable analog blocks for the two low pass filters, and four programmable analog blocks for the four programmable gain amplifiers. In addition, the implementation used two microphones, two 2K resistors, two 8k resistors, two 47k resistors, two 10Mresistors, and two disc ceramic capacitors of 0.1uF.

Table 3.1 and Table 3.2 summarize the memory requirement and the execution time, respectively. Both the implementations, maximum likelihood and alternative method, were run on a Cypress PSoC CY8C29466. The system clock was 24MHz and the ADC sampling frequency was 13.3 KHz. 128 samples are taken simultaneously on both the microphone channels. Hence, the

Routine	**Max Likelihood**	**Alt. Method**
Sampling	256 Bytes	256 Bytes
Hanning w.	512 Bytes	512 Bytes
FFT	1024 Bytes	1024 Bytes
Cordic	512 Bytes	30 Bytes
Max Likelihood	648 Bytes	-
DOA equations	-	14 Bytes
Variables	64 Bytes	64 Bytes
TOTAL	3016 Bytes	1900 Bytes

Table 3.1: Memory Requirement

sampling duration is 9.69 msec. The Hanning window and FFT procedures are common to both the methods. For the implementation using Maximum Likelihood, the memory requirement is very high. The major contributors are CORDIC and Maximum Likelihood. CORDIC is run 128 times and the Maximum Likelihood equation is run more than 5000 times. This results in a total execution time slightly more than 45 seconds. This means that for 45 seconds the node is busy processing the data and will not be able to detect incoming signals in that duration. For the alternative method, CORDIC is used only twice and the DoA equations are executed once. Therefore, the memory requirement and the execution time is very low as compared to Maximum Likelihood. The entire algorithm executes in just 3.67 sec.

Table 3.3 gives comparison between execution time for PSoC1 and PSoC3. The drastic reduction in the total execution time is because the PSoC3 can operate at 66 MHz clock, has a 8051 core and uses the optimized Keil compiler. Therefore, PSoC3 is used as central node for the network implementation.

Routine	**Max Likelihood**	**Alt. Method**
Sampling	9 ms	9 ms
Hanning w.	801 ms	801 ms
FFT	2834 ms	2834 ms
Cordic	881 ms	13 ms
Max Likelihood	41340 ms	-
DOA equations	-	9 ms
TOTAL	45865 ms	3666 ms

Table 3.2: Execution Time

Routine	**PSoC1**	**PSoC3**
Sampling + Hann w. + FFT + Cordic	4525	204 ms
Max Likelihood + DOA equations	41340 ms	776 ms
TOTAL	45865 ms	980 ms

Table 3.3: Execution Time

3.1.2 Memory Management

PSoC CY8C29466 has an internal SRAM of 2KB and flash memory 32KB. The SRAM is split over 8 pages each of size 256 Bytes. Hence, each page can hold at the most 64 float or double variables. Thus, memory management is crucial when the entire algorithm is to be implemented on a single PSoC1. Due to this memory constraint, the number of samples is decided as 128 for each microphone. Since, the resolution of ADC is 7 bits, each sample is of size 1 byte. Hence, entire page 1 is required for storing the samples. First half of page zero is left vacant for interrupt routines. Second half of page 1 is used for local variables used during the course of execution of the C code. After the samples have been received and stored in page 1, we first process the samples collected by Mic 1. At the output of FFT, we get 128 values corresponding to the real part and 128 values corresponding to the imaginary part. Each of these values is a double precision floating point number and requires 4 bytes of memory. Hence, the output of FFT for Mic 1 occupies pages 2, 3, 4 and 5.

For implementing Maximum Likelihood algorithm, we need to compute phase difference at each point on the FFT. Phase values are computed for current FFT points and are overwritten onto page 2 and page 3. These two pages are then pushed onto the flash memory using a virtual EEPROM created using a reconfigurable block in the PSoC. This virtual EEPROM helps the users to store data onto flash memory which is traditionally reserved as program memory. However, the user has to ensure that memory occupied by the code is not overwritten. The samples from second microphone are then processed. The FFT and phase calculation is completed. Then, the phase values pushed onto flash are read back into the SRAM and phase difference is calculated. This is then used to calculate TDOA using Maximum Likelihood. A flowchart representation of this memory usage process is as shown in Figure 3.1.

By using the alternative wave counting solution, we no longer need to compute the phase difference for all the values. We identify the peak from the FFT and calculate phase only for that point. This reduces the memory requirement by a huge amount. Also, we no longer need to use the virtual EEPROM. This reduces code length and hence, flash memory usage also goes

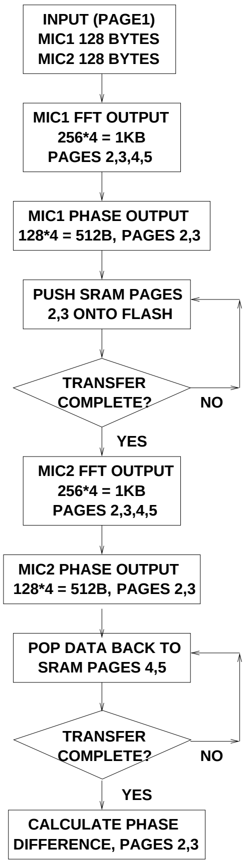


Figure 3.1: Memory Management

down.

While designing an embedded application, it is extremely important that we use minimum amount of memory and the time of execution should be as low as possible. There are two ways to minimize the execution time: optimize the code and optimize the logic. Optimization of the code depends on the coding skills of the user. There are however, some optimizations, which can be used to bring down the code size. These methods include: use of functions or macros, looping and recursive calls. In order to optimize the logic, we can study the concepts of the algorithm and think of ways to bypass certain modules. In the case of this report, it was observed that maximum likelihood is the part of the algorithm which utilizes the most time for execution. Hence, we have proposed an alternative solution which will bring down the time required by a significant amount.

3.2 Hardware Resources for Network Implementation

The sensor network implementation uses the maximum resources. Microphone circuitry captures the input signal which is then conditioned using reconfigurable amplifier and filter modules.The ADC used for sampling has a resolution of 7 bits and sample rate 13.33 KHz. This analog frontend is implemented on Cypress PSoC family 1 chip CY8C29466 which has a system clock of 24 MHz. The hardware modules for this chip were designed using PSoC Designer 5.0. UART communication between Sensing node and Network node operates at 4.8 Kbps. The data transfer between PSoC and the Radio Module CYWM 6935 on the Network node is implemented using SPI protocol. The data rate for this transfer is 19.2 Kbps. The Radio modules operate in the 2.4 GHz ISM band. The transceivers use Direct Sequence Spread Spectrum (DSSS) technique to achieve a data rate of 31.25 Kbps. The Cypress PSoC family 3 chip CY8C3866 is operated at a system clock of 24 MHz. The hardware and software modules for PSoC 3 are designed using PSoC Creator 1.0. Prior to implementation, simulations and verification was performed on MATLAB 7.0.

Chapter 4

Dynamic Reconfiguration

It has been observed that the localization accuracy increases with increase in the Signal-to-Noise ratio [2]. One method to improve the SNR is dynamic reconfiguration of the analog frontend. Using the Cypress PSoC Family 1 chip and some decision making mechanism, the Programmable Gain Amplifier as well as the Low Pass Filter modules can be reprogrammed on-the-fly. Gain reconfiguration increases signal strength while cutoff reconfiguration decreases rms noise. This results in improved SNR and consequently, better localization accuracy.

4.1 Filter Cutoff Reconfiguration

The cutoff frequency (Fc) for the low pass filter module can be adjusted to a desired value, so that the noise associated with higher frequencies is attenuated. Consequently, the SNR improves and the localization accuracy increases.

For example, the input is a 1 KHz single frequency tone with 0 dB SNR. The FFT plot for such an input is shown in Figure 4.1.

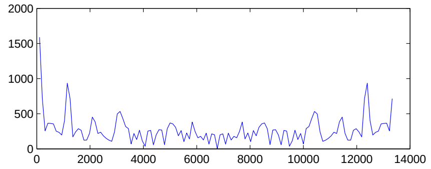


Figure 4.1: FFT plot for Input signal

FFT plot for the filter output with a cutoff frequency of 10 KHz is shown in Figure 4.2.

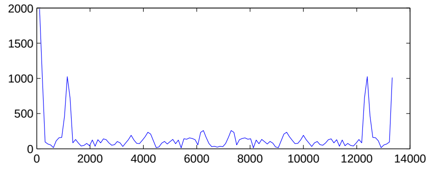


Figure 4.2: FFT plot for output of filter with 10 KHz cutoff

FFT plot for the filter output with a cutoff frequency of 2 KHz is shown in Figure 4.3.

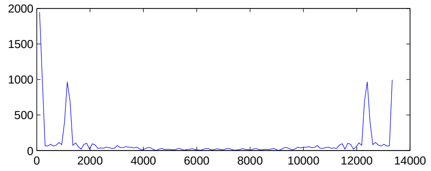


Figure 4.3: FFT plot for output of filter with 2 KHz cutoff

As seen in Figure 4.3, the noise associated with higher frequencies is diminished.

The decision-making mechanism for filter reconfiguration is as follows: During the first iteration, the filter cutoff frequency is set to 10 KHz. The frequency components of the input signal are then identified by analyzing the

FFT. The cutoff frequency for the next iteration is then decided by observing the most significant components in the FFT. The filter parameters can be modified by using the LPF2 module APIs. However, to operate the filter at high performance the oversampling ratio has to be maintained above 100. Therefore, the filter cutoff is adjusted by changing the column clock. This can be done by loading the appropriate values in the Oscillator Control Register.

4.2 Gain Reconfiguration

The input to the analog frontend is amplified by the Programmable Gain Amplifier (PGA) as well as the Low Pass Filter (LPF) module. We can monitor the signal captured by the ADC and adjust the amplifier and filter gain to maximize the signal strength for the next iterations. The amplifier gain can be dynamically modified using the PGA module APIs. Also, the filter gain can be varied by changing the C1/C2 capacitor ratio. The total gain can be estimated as shown in Table 4.1.

PGA Gain	**C1/C2 Ratio**	**LPF Gain(dB)**	**LPF Voltage Gain**	**Total Gain**
24	3	8	3.16	75.84
24	4	12	3.98	95.52
24	5	14	5.01	120.24
24	6	15	5.62	134.95

Table 4.1: Gain Reconfiguration

During the first iteration, the voltage gain for the amplifier is 24 and for the filter is 3.16 (8 dB). This results in a total gain of 75.84. The only gain value available for the PGA module beyond 24 is 48. Hence, the amplifier can be used when a drastic improvement in signal strength is required. However, during normal operation, we can fine tune the implementation by varying the filter gain in small steps. However, excessive gain might amplify the signal such that it exceeds the operating range for PSoC, resulting in clipping of the signal. Hence, we use a decision making mechanism to select the optimum value for amplification.

The decision-making mechanism for gain reconfiguration is as follows: During the first iteration, the filter gain is 3.16 (8 dB). The samples from the ADC are then analyzed to identify the peak value. This value is converted to its equivalent voltage using the ADC conversion formula. The voltage is then divided by the total gain of the system which gives us the peak value of the actual input signal. This peak value is multiplied with a series of available gain options such that the output should not exceed 5V. The optimum value for gain is selected and the C1/C2 ratio in the filter is modified accordingly.

4.3 Reconfiguration for Temperature Sensing

The Direction of Arrival (DOA) is calculated by using the following equation:

$$DoA = asin(\tau * v/d) \tag{4.1}$$

where, τ is the Time Difference of Arrival (TDOA) estimate, v is the speed of sound in air and d is the inter-microphone spacing.

However, in real environments, the speed of sound increases by 0.58 ms for every degree Celsius rise in temperature. Hence, in order to observe the effect of temperature variation on DoA estimation, a valid reading at 23 degree Celsius is considered for each of the DoA values. Then the temperature is varied from 15 to 30 degree Celsius and the expected value is compared with the result. Figure 4.4 depicts the absolute error, for variable temperature conditions. The error is more for higher values of DoA. This is due to the non linear nature of the asin function in the DOA equation. However, the curves are linear because the speed of sound varies in a definite step-size for every degree variation in temperature. A temperature sensing module is added to the design to monitor the temperature in the room. The value of speed of sound is then changed accordingly. This will eliminate fluctuations in readings, especially when measurements are taken over multiple days.

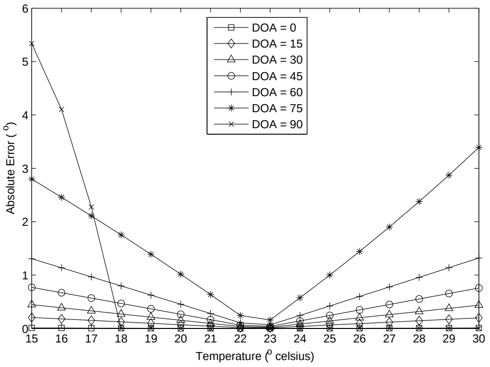


Figure 4.4: Impact of temperature on localization accuracy

The external circuitry required for temperature sensing consists of a thermistor and a resistor. An ADC is required to sample the voltage level at the thermistor circuitry. By exploiting the reconfigurability of PSoC, we can reuse the hardware which is used for sound localization. The reconfiguration is done as follows: The Analog Column Input MUX is used to switch input from microphone circuitry to the thermistor circuitry. The gain of the PGA and LPF blocks is set to 1. The ADC is then used to sample the input voltage. The temperature can be calculated using the Steinhart-Hart method. The MUX input, PGA and LPF parameters are then reset so that the system can be used for sound localization.

Chapter 5

Wireless Network Implementation

The network of PSoCs used for localization is as shown in Figure 5.1.

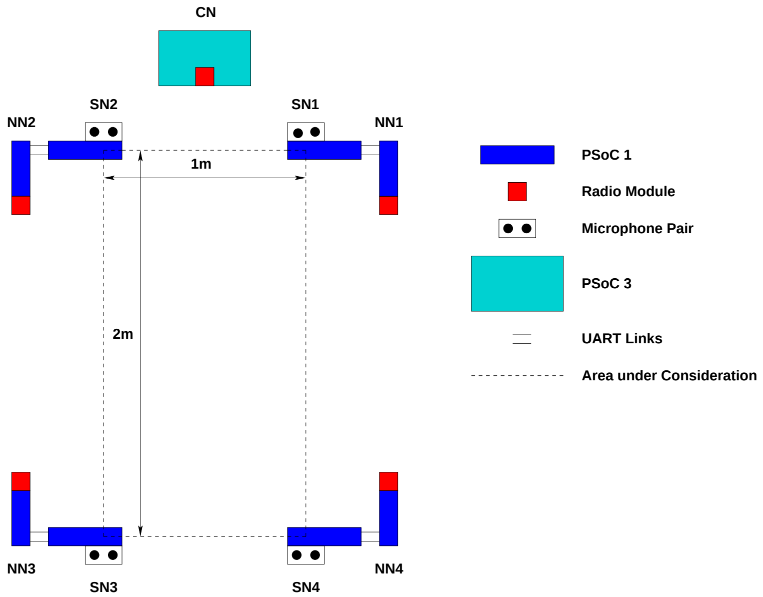


Figure 5.1: Wireless Sensor Network

Each of the microphone pairs is connected to a Sensing Node (SN). Each of the SNs is connected to a Network Node (NN) via UART links. The NNs are interfaced with Radio modules. The communication between the PSoC chip and Radio Module uses SPI protocol. The NNs communicate with the Central Node (CN) via wireless links. A star topology is thus implemented with a PSoC family 3 chip as the CN and PSoC family 1 chips as the NNs and SNs. It is important to observe that only two sensing nodes (SN1 and SN2) are sufficient for localizing the sound source. However, nodes SN3 and SN4 are added as redundancy such that when the DOA for SN1 and SN2 goes beyond 60 degrees, the DOA estimates from SN3 and SN4 are given priority and vice versa.

The localization process is executed on the network as follows: The sound signal emitted by the source is captured by the SN and converted to digital samples. 128 samples are recorded on each of the microphone channels. Therefore, each sensing node collects a total of 256 samples. These are then transferred to the NN using UART wired links. Each NN contains a unique ID number. The CN transmits a packet ordering the node with a particular ID number to start sending the samples. In order to compensate for the data loss, each NN is given 5 chances to transmit the 256 bytes. As soon as these samples are received, the CN processes the data and displays the DOA result for that SN. The CN then communicates with the next NN and the process continues in a round robin fashion.

The flowcharts for Sensing Node and Network Node are as shown in Figure 5.2. The flowchart for Central Node is as shown in Figure 5.3.

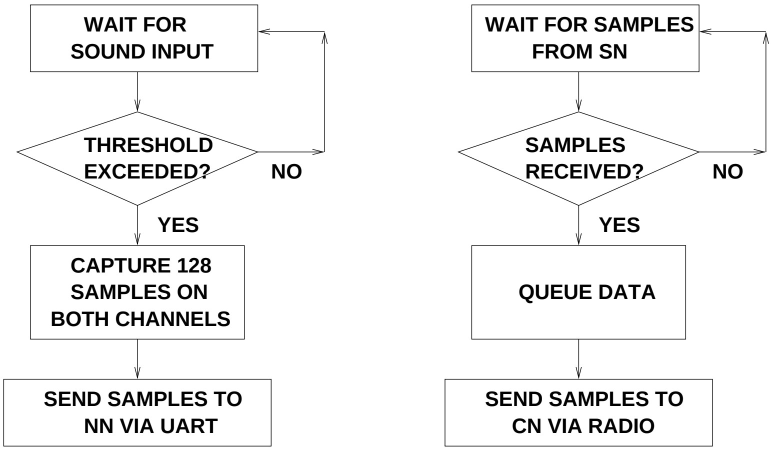


Figure 5.2: (a) SN Flowchart, (b) NN Flowchart

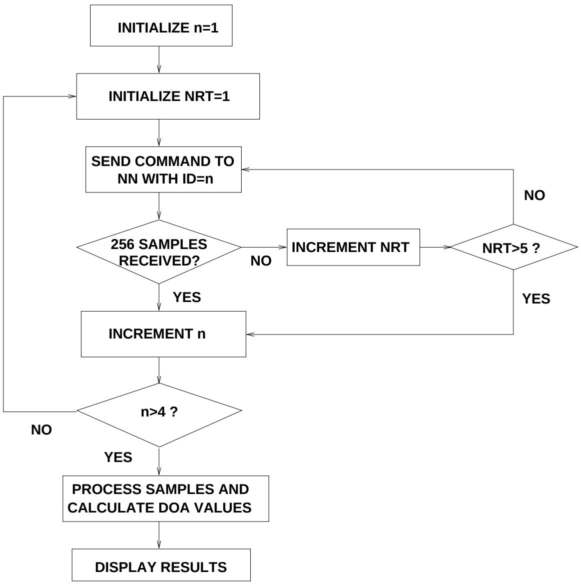


Figure 5.3: CN Flowchart

Chapter 6

Experiments

6.1 Localization Accuracy

The test setup for sound localization on PSoC is as shown in Figure 6.1. The microphone pair is situated at least 1 m away from either wall and at least 1 m above the floor. This is done to reduce effect of reverberations.

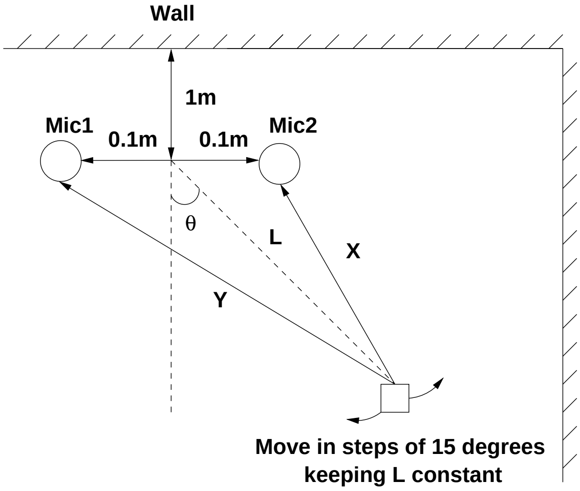


Figure 6.1: Test Setup

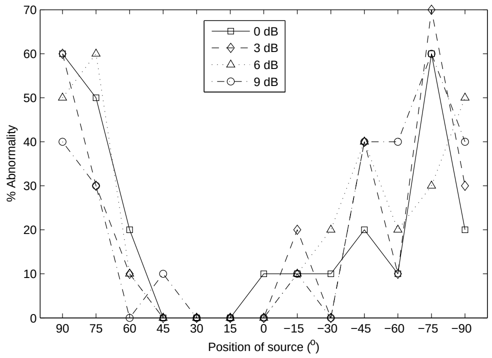


Figure 6.2: ML Results for High Noise conditions

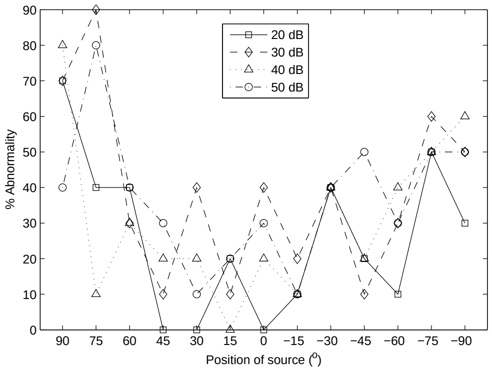


Figure 6.3: ML Results for Low Noise conditions

6.1.1 Maximum Likelihood Results

Figure 6.2 and Figure 6.3, plot the percentage of abnormalities for localization including Maximum Likelihood algorithm for high noise and low noise conditions, respectively. Pre-recorded speech was mixed with white noise using Matlab. The signal was then played in a laboratory environment using speaker placed at a distance of 1 m from the microphone pair. The different SNR values are 3dB, 6dB, 9dB, 20dB, 30dB, 40dB and 50dB. Readings were taken for values of DOA ranging from -90 to +90 in steps of 15 degrees. Ten readings were taken for each angle for a particular SNR. A reading was considered abnormal if its value deviates from the expected value by more than 5 degrees.

6.1.2 Wave Counting Results

Figure 6.4 plots the average errors for localizing sources at different angles using alternative method. Figure 6.5 displays the percentage abnormality plot for the alternative method at SNR of approximatley 50dB. Single frequency tones of 1KHz, 2KHz, 3KHz and 4 KHz were used as inputs. The signal was then played in a laboratory environment using speaker placed at a distance of 1 m from the microphone pair. Readings were taken for values of DOA ranging from -90 to +90 in steps of 15 degrees. Ten readings were taken for each angle for a particular frequency. A reading was considered abnormal if its value deviates from the expected value by more than 5 degrees.

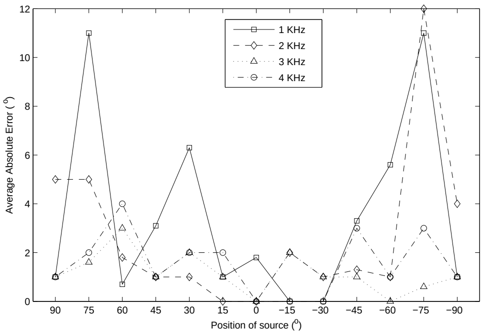


Figure 6.4: Alternative Method : Average absolute error

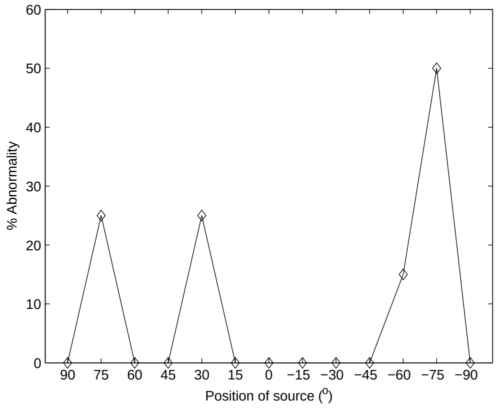


Figure 6.5: Alternative Method results for low noise condition

6.2 Reconfiguration Results

6.2.1 Cutoff Frequency Reconfiguration Results

A 1 KHz single frequency tone is mixed with white noise to obtain 0 dB SNR. This signal is used to perform experiments to quantify the percentage improvement in the rms value of noise due to Fc reconfiguration. The results are as shown in Table 6.1.

The percentage improvement in rms noise ranges from 1.54% to 41.62% with an average improvement of 17.98%.

A 4 KHz single frequency tone is mixed with white noise to obtain 0 dB SNR. This signal is used to perform experiments to quantify the percentage improvement in the rms value of noise due to Fc reconfiguration. The results are as shown in Table 6.2.

The percentage improvement in rms noise ranges from 4.22% to 23.15% with an average improvement of 12.90%.

RMS noise (Fc = 10KHz)	RMS noise (Fc = 2KHz)	Percent Improvement
49.57	39.42	20.47
52.93	43.49	17.83
53.97	46.01	14.74
40.86	40.23	1.54
55.15	42.13	23.60
51.72	49.60	4.09
52.72	38.99	26.04
49.18	46.25	5.95
53.64	31.31	41.62
55.56	42.26	23.93

Table 6.1: Fc Reconfiguration Results

A 6 KHz single frequency tone is mixed with white noise to obtain 0 dB SNR. This signal is used to perform experiments to quantify the percentage improvement in the rms value of noise due to Fc reconfiguration. The results are as shown in Table 6.3.

The percentage improvement in rms noise ranges from 2.11% to 16.15% with an average improvement of 10.40%.

6.2.2 Gain Reconfiguration Results

A 1 KHz single frequency tone is used as input signal for the analog frontend. The total gain for the system is initially 75.84. The plot for the ADC samples is shown in Figure 6.6.

The optimum voltage gain selected by the decision mechanism is 5.01 (14dB). This results in a total gain of 120.24. The plot for the ADC sam-

RMS noise (Fc = 10KHz)	**RMS noise (Fc = 5KHz)**	**Percent Improvement**
88.56	72.42	18.22
91.56	78.19	14.60
86.73	71.86	17.14
97.69	75.07	23.15
103.89	90.95	12.45
109.85	101.12	7.94
93.63	88.38	5.60
79.50	76.14	4.22
90.48	80.95	10.53
85.63	72.63	15.18

Table 6.2: Fc Reconfiguration Results

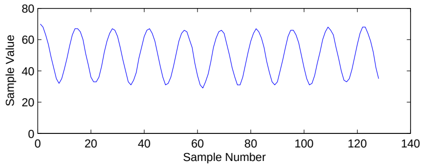


Figure 6.6: ADC Samples for filter gain = 8dB

ples for this iteration is shown in Figure 6.7.

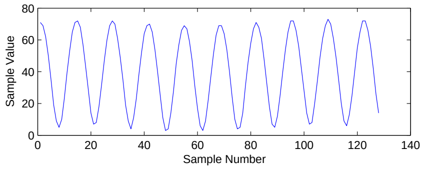


Figure 6.7: ADC Samples for filter gain = 14dB

6.3 Wireless Sensor Network Results

The network was first tested by placing the sound source in 3 different positions such that the DOA is less than 60 degrees for all the SNs. The results are as

RMS noise (Fc = 10KHz)	**RMS noise (Fc = 6.5KHz)**	**Percent Improvement**
53.68	47.05	12.35
44.78	41.47	7.39
52.94	45.67	13.73
54.37	48.61	10.59
47.22	39.59	16.15
55.80	54.62	2.11
41.72	37.50	10.11
62.45	55.40	11.28
60.71	53.38	12.07
56.67	52.03	8.18

Table 6.3: Fc Reconfiguration Results

shown in Table 6.4.

The experimental results for DOA greater than 60 degrees are as shown in Table 6.5.

The distance between SN1 and SN2 was doubled from 1m to 2m. The distance between SN2 and SN3 was also doubled from 2m to 4m. The area under consideration is therefore doubled from 2m x 1m to 4m x 2m.

The previous experiments were repeated for the new network dimensions. The results for DOA less than 60 degrees are as shown in Table 6.6.

The experimental results for DOA greater than 60 degrees are as shown in Table 6.7.

Note that in all situations the wireless sensor network improves the precision of the phase-based sound localization algorithm by selecting the readings of the two nodes with the lesser localization error.

	SN1	**SN2**	**SN3**	**SN4**
Expected DOA	-28.60	28.60	-28.60	28.60
Reading 1	-31.16	26.92	-21.13	32.91
Reading 2	-31.69	26.90	-24.45	30.30
Reading 3	-31.16	26.90	-26.92	27.75
Reading 4	-30.30	27.75	-25.27	26.92
Reading 5	-32.30	26.90	-29.14	22.83
Expected DOA	0.00	47.27	-47.27	0.00
Reading 1	0.74	44.31	-52.11	0.74
Reading 2	2.22	57.24	-43.19	-0.74
Reading 3	0.00	45.36	-54.19	0.74
Reading 4	1.48	46.42	-46.42	0.74
Reading 5	0.00	46.42	-54.59	2.22
Expected DOA	-47.27	0.00	0.00	47.27
Reading 1	-55.89	0.00	-1.48	49.75
Reading 2	-52.11	-0.74	-1.48	49.75
Reading 3	-49.75	0.74	-5.19	50.92
Reading 4	-40.30	-0.74	-2.22	52.11
Reading 5	-50.92	0.00	-1.48	45.36

Table 6.4: Network Results for DOA < 60 °

	SN1	**SN2**	**SN3**	**SN4**
Expected DOA	0.00	68.30	-35.60	0.00
Reading 1	-2.22	89.99	-35.10	0.00
Reading 2	0.74	52.11	-36.92	2.96
Reading 3	0.00	73.21	-37.10	-1.48
Reading 4	-0.74	61.61	-34.30	-1.48
Reading 5	0.74	61.61	-33.80	1.48
Expected DOA	-68.30	0.00	0.00	35.60
Reading 1	-73.21	-2.22	0.74	38.39
Reading 2	-68.67	-1.48	-2.22	37.45
Reading 3	-60.90	-2.22	-1.48	37.45
Reading 4	-70.81	-2.22	-0.74	33.8
Reading 5	-58.63	-0.74	-0.96	33.80
Expected DOA	-35.60	0.00	0.00	68.30
Reading 1	-36.52	-2.22	0.00	76.00
Reading 2	-31.16	-0.74	0.74	54.59
Reading 3	-38.39	-0.74	-0.74	60.90
Reading 4	-33.80	-2.22	0.00	58.63
Reading 5	-33.80	0.00	0.00	55.89
Expected DOA	0.00	35.60	-68.30	0.00
Reading 1	-0.74	37.45	-64.91	0.74
Reading 2	0.74	34.69	-61.61	0.74
Reading 3	0.74	35.60	-89.99	1.48
Reading 4	-0.74	37.45	-89.99	0.00
Reading 5	-0.74	36.52	-76.00	0.74

Table 6.5: Network Results for DOA > 60 °

	SN1	**SN2**	**SN3**	**SN4**
Expected DOA	-27.75	27.75	-27.75	27.75
Reading 1	-31.80	26.92	-25.27	28.60
Reading 2	-30.30	30.30	-26.90	29.44
Reading 3	-30.30	26.92	-30.30	29.44
Reading 4	-30.30	30.30	-31.16	29.44
Reading 5	-30.30	26.92	-27.75	28.60
Expected DOA	0.00	48.07	-48.07	0.00
Reading 1	-3.10	49.75	-50.92	0.19
Reading 2	-1.48	51.61	-49.57	0.68
Reading 3	-1.48	46.42	-45.36	4.45
Reading 4	-4.45	46.42	-50.92	2.96
Reading 5	-0.48	46.42	-45.36	0.19
Expected DOA	-48.07	0.00	0.00	48.07
Reading 1	-48.62	0.00	0.00	50.92
Reading 2	-48.62	0.19	-0.90	49.75
Reading 3	-45.59	1.48	0.00	53.33
Reading 4	-53.33	0.48	-0.96	50.11
Reading 5	-46.42	2.22	-2.96	50.92

Table 6.6: Network Results for DOA < 60 °

	SN1	**SN2**	**SN3**	**SN4**
Expected DOA	0.00	65.81	-34.60	0.00
Reading 1	-0.74	70.81	-33.80	0.00
Reading 2	-2.96	64.91	-36.52	1.48
Reading 3	-0.74	54.59	-36.52	0.00
Reading 4	0.74	52.11	-33.80	5.94
Reading 5	-1.48	66.72	-41.28	2.96
Expected DOA	-65.81	0.00	0.00	34.60
Reading 1	-63.21	0.00	0.00	37.45
Reading 2	-50.92	0.74	2.96	37.45
Reading 3	-49.75	0.00	-0.74	33.80
Reading 4	-57.24	0.00	-2.22	32.91
Reading 5	-89.99	-0.74	-3.70	37.45
Expected DOA	-34.60	0.00	0.00	65.81
Reading 1	-39.34	-5.19	-0.74	60.90
Reading 2	-34.69	-2.96	-1.48	66.72
Reading 3	-37.45	-1.48	-1.48	57.24
Reading 4	-40.30	-3.70	-0.74	76.00
Reading 5	-37.45	-0.74	-1.48	55.89
Expected DOA	0.00	34.60	-65.81	0.00
Reading 1	-0.43	39.34	-64.91	0.74
Reading 2	-0.74	31.16	-60.90	-0.74
Reading 3	-5.19	30.30	-50.92	1.48
Reading 4	-2.22	32.91	-64.91	0.74
Reading 5	1.18	35.60	-53.33	0.00

Table 6.7: Network Results for DOA > 60 °

Chapter 7
Summary

7.1 Related work

There have been extensive theoretical studies on sound-based localization using both pairs of microphones and microphone arrays [9, 10, 11, 12, 13]. However, there has been much less work on implementing and experimenting sound-based localization realized as customized electronic designs. This is important to achieve low cost requirement which in turn is important for large-scale deployment of localization in practical applications.

Azimi-Sadjadi et al. [4] describe an FPGA-based implementation of sound localization. The work focuses on detection of transient sound signals, including onset and duration. The PCB-based implementation uses a Xilinx Spartan-3L FPGA for the main sound localization algorithm, five analog channels, and a mote for wireless connection and node programming. The implementation presented in [6] adapts the principles of sound-based localization to compute the speed of air (e.g., wind). It exploits the fact that the sound velocity changes with the air speed. Digital processing is performed using ARM-SA1100 microprocessor and a Xilinx XC4000XLA FPGA. The analog fronend includes one 12-bit analog-to-digital converter. Halupka et al. [2] discuss a customized integrated chip in 0.18um process for low power execution of the digital signal processing routines needed for localization. The FPGA implementation in [14] uses the same algorithm as described in [2]. Xilinx Virtex II 2000 (2V2000) FPGA is used for implementation. The samples obtained from the analog frontend are fed digitally to the FPGA and are stored in user selectable buffers before being processed. Jin et. al.

[15] describes a TDOA estimation algorithm using 3 microphones. Sampling is done using a lower clock and processing using a higher clock value. A sliding window mechanism is used to continuously sample and process incoming signals. For DOA estimation, correlation is used for all possible pairs of the 3 microphone elements. The system employs Analog Device AD1836A codec IC and Xilinx Virtex-4 FPGA for sound signal capturing and processing, respectively. Kugler et al. [16] discuss a hardware implementation for sound localization using neural networks. The FPGA used for this purpose is an Altera Stratix II and the audio sampling is done using AD7864. Apart from localization estimation, a sound classification estimator is also implemented on the same FPGA. An ASIC implementation for a sound localization system is described in [17]. Monaural and binaural localization cues are extracted from the signals incident on the left and right channels. Onset Detector and Envelope detectors are used to improve signal fidelity. Also, spectral cue frontend and time delay estimator are used to extract the spectral cues and time de-

lay information. Sakamoto et al. [18] have described a 3-D sound localization algorithm. Signal from the monaural sound source is split into 3 frequency subbands: low, intermediate and high frequency. These are analyzed separately and the delay estimates are extracted from each subband. These estimates are then mixed at the output stage. The implementation is accomplished using a Texas Instruments 16-bit fixed-point DSP, TMS320C54x to obtain real-time localization.

Localization using microphone arrays is discussed in [12]. Ledeczi et al. [1] discuss a sensor network implementation for sound-based detection of counter-sniper position. Sensor nodes are based on MICA 2 motes and a customized board for real-time detection, classification and correlation of acoustic information. There are two versions of the customized board, one using Xilinx XC2s100 FPGA with three analog channels, and second using ADSP-218x digital signal processor with two analog channels.

7.2 Future Scope

There are multiple factors which contribute towards errors in the sound-based localization implementation using PSoC. Future work will address reducing the impact of these factors. These can be identified as follows:

(i) According to [2], the number of samples should be at least 256 samples for both the channel. However, due to memory limitations, the number of samples used in the PSoC implementation is 128 samples. This problem can be resolved using the network implementation. (ii) Different materials exhibit different reflection and absorption coefficients. It has been observed that the material of the floor between the microphone pair and the sound source, affects the phase as well as amplitude of the signal received. (iii) As the distance between the microphone pair and the sound source decreases, the DoA estimates become coarser. (iv) The position of sources of ambient noise in the room is important. This will affect the nature of the percentage abnormality plot causing it to become non-symmetric. (v) Position of reflective surfaces around the experimental setup contributes towards the fluctuations in the Rms error curves. (vi) Physical parameters such as speaker width and sensitivity of the microphone contribute towards measurement errors. (vii) The frequency response of the microphone elements also affects the fidelity of the captured signal. (viii) Accuracy of experimental setup and error due to elevation of microphone and sound source are also factors which may cause errors.

7.3 Conclusions

This thesis presents an implementation of a sound-based localization technique using PSoC programmable mixed-signal System on Chip. The report summarizes the basics of sound-based localization as discussed in the literature. The process of time delay of arrival estimation is explained and an alternative solution, wave counting, is proposed. Then, the report explains the microphone circuitry, the analog front-end, and the related DSP methods. Waveforms are provided to support the theoretical analysis. The different optimization methods to decrease memory usage and reduce execution time are also explained in detail. Finally, a comprehensive set of experimental results are offered.

Even though there are memory constraints, PSoC can be used effectively to design low cost, sound localization applications. The execution time of wave counting based method is about 3.67 seconds, which is 12.5 times faster than the method using Maximum Likelihood. Also, its memory usage is about 1.5 times less. However, wave counting is limited to single tone inputs, which might be a limiting constraint for many applications. The wave counting method has lower percent of abnormal readings and Rms error than method using Maximum Likelihood, but the latter performs better in high noise conditions. The accuracy of the two low cost, PSoC-based implementations is comparable to that of the integrated chip implementation discussed in [2].

Reconfigurability of PSoC is exploited to achieve 17.98 % reduction in rms noise by dynamically varying the filter corner frequency. Also, reconfiguration is used to modify the system gain depending on the signal strength. The combined effect of these techniques causes an improvement in the signal to noise ratio which leads to better localization accuracy. Also, a temperature sensing application is appended to the design to further improve the localization estimates. Finally, it is experimentally proved that a wireless sensor network can be used to overcome the shortcomings of the phase-based sound localization algorithm.

Bibliography

[1] Umbarkar, A. and Subramanian, V. and Doboli, A. *Low-Cost Sound-based Localization using Programmable Mixed-Signal Systems-on-Chip*, Submitted to Microelectronics Journal, 2010.

[2] Umbarkar, A. and Subramanian, V. and Doboli, A. *Improved Sound-based Localization Through a Network of Reconfigurable Mixed-Signal Nodes*, Submitted to Proc. Int'l. Conference on Robotic and Sensors Environments, 2010.

[3] Ledeczi, A. and Nadas, A. and Volgyesi, P. and Balogh, G. and Kusy, B. and Sallai, J. and Pap, G. and Dora, S. and Simon, G. *Countersniper system for urban warfare*, ACM Transactions on Sensor Networks, 1:153-177, 2005.

[4] Halupka, D. and Mathai, J. and Aarabi, P. and Sheikholeslami, A. *Robust sound localization in 0.18 um cmos*, IEEE Transactions on Signal Processing, 53(6), 2005.

[5] Chen, Member, J. C. and Yip, L. and Elson, J. and Wang, H. and Maniezzo, D. and Hudson, R.E. and Yao, K. and Estrin, D. *Coherent*

acoustic array processing and localization on wireless sensor networks, Proceedings of IEEE, 91(8):1154-1162, 2003.

[6] Azimi-Sadjadi, M. and Kiss, G. and Feher B., and Srinivasan, S. and Ledeczi, A. *Acoustic source localization with high performance sensor nodes*, In Unattended Ground, Sea, and Air Sensor Technologies and Applications IX, Proc. 2008.

[7] A. Doboli, E. Currie *Introduction to Mixed-Signal Embedded Design*,

[8] Karalar, T. C. *An Acoustic Digital Anemometer*, MS Thesis, University of California at Berkeley, 2002.

[9] Cypress Semiconductor Corporation. *Psoc mixed signal array*, Document No. PSoC TRM, 1.21, 2005.

[10] Knapp, C. H. and Carter, G. *The generalized correlation method for estimation of time delay*, IEEE Transactions on ASSP, 24(4):320-327, 1976.

[11] Brandstein, M. and Silverman, H. *A robust method for speech signal time-delay estimation in reverberant rooms*, pages, 375-378, 1997.

[12] Brandstein, M. and Adcock, J. and Silverman, H. *A practical time-delay estimator for localizing speech sources with a microphone array*, pages, 153-169, 1995.

[13] DiBiase, J. and Silverman, H. and Brandstein, M. *Microphone arrays: Signal processing techniques and applications*, M. Brandstein and D. Ward (Eds.), Springer, 2001.

[14] Aarabi, P. *The fusion of distributed microphone arrays for sound localization*, pages, 338-347, 2003.

[15] Gannot, S. and Dvorkind, T. G. *Microphone array speaker localizers using spatial-temporal information*, pages, 1-17, 2006.

[16] Nguyen, D. and Aarabi, P. and Sheikholeslami, A. *Real-time sound localization using field-programmable gate arrays*, Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 2003.

[17] Jin, S. and Kim, D. and Kim, H.S. and Lee, C.H. and Choi, J.S. and Jeon, J.W. *Real-time sound source localization system based on fpga*, Proc. IEEE Intern'l Conference on Industrial Informatics, 2008.

[18] Kugler, M. and Iwasa, K. and Benso, V. and Kuroyanagi, S. and Iwata, A. *A complete hardware implementation of an integrated sound localization and classification system based on spiking neural networks*, pages, 577-587, 2008.

[19] Grech, I. and Micallef, J. and Vladimirova, T. *Analog cmos chipset for a 2-d sound localization system*, Analog Integrated Circuits and Signal Processing, Kluwer, 41:167-184, 2004.

[20] Sakamoto, N. and Kobayashi, W. and Onoyettt, T. and Shirakawa, I. *Dsp implementation of low computational 3d sound localization algorithm*, In IEEE Workshop on Signal Processing Systems, 2001.