Robust Sound Localization in 0.18 μm CMOS

David Halupka, *Student Member, IEEE*, Nebu John Mathai, *Student Member, IEEE*, Parham Aarabi, *Member, IEEE*, and Ali Sheikholeslami, *Senior Member, IEEE*

***Abstract*—This paper presents a hardware implementation of a sound localization algorithm that localizes a single sound source by using the information gathered by two separated microphones. This is achieved through estimating the time delay of arrival (TDOA) of sound at the two microphones. We have used a TDOA algorithm known as the "phase transform" to minimize the effects of reverberations and noise from the environment. Simplifications to the chosen TDOA algorithm were made in order to replace complex operations, such as the cosine function, with less expensive ones, such as iterative additions. The custom digital signal processor implementing this algorithm was designed in a 0.18-μm CMOS process and tested successfully. The test chip is capable of localizing the direction of a sound source within 2.2° of accuracy, utilizing approximately 30 mW of power and 6.25 mm^2 of silicon area.**

***Index Terms*—Acoustic arrays, acoustic signal processing, adaptive arrays, application specific integrated circuits, array signal processing, circuit optimization, digital signal processors, microphones.**

I. Introduction

LOW-POWER electronics are widely utilized in hand-held organizers, cell phones, and digital cameras. Hand-held organizers in particular require a substantial amount of user input, often provided via a small keyboard or touch screen, neither of which is user friendly. Voice-activated portable electronics, on the other hand, provide a more suitable form of user input. To that end, sound localization in conjunction with speech separation can increase the accuracy of a speech recognition system in noisy environments [1]–[3]. Speech separation is the process of separating sound signals from each other using a multimicrophone array. Sound localization is the process of identifying the spatial coordinates of a sound source based on the sound signals received by the microphone array. This is in direct contrast with sound projection [4], which synthetically projects a monaural sound source in a simulated three-dimensional (3-D) environment. Sound localization is required for applications such as video conferencing [5], robot navigation [6], and speech recognition [7], [8]. This paper examines a sound localization algorithm based on the phase transform and examines a way of implementing this algorithm in hardware.

Manuscript received December 18, 2003; revised July 8, 2004. The associate editor coordinating the review of this manuscript and approvng it for publication was Prof. Karim Drouiche.

D. Halupka, P. Aarabi, and A. Sheikholeslami are with the Artificial Perception Laboratory, the Edward S. Rogers, Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada M5S 3G4.

N. J. Mathai is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843 USA.

Digital Object Identifier 10.1109/TSP.2005.847820

A simple method of multimicrophone sound localization is to estimate the time delay of arrival (TDOA) of a sound signal between two microphones. Many TDOA algorithms are widely available [7], [9]–[11], among which, the phase transform weighted version is most suitable for reverberant environments. There are two approaches to implementing this algorithm via software or hardware.

Software-based sound localization is not capable of providing real-time sound localization, due to the processing requirements of the algorithm. Our own experiments with software implementations have shown that a 2-GHz Intel® Pentium® 4 provides only half-rate real-time sound localization estimates. An off-the-shelf digital signal processor (DSP) solution is not a viable approach as it is not scalable and might require a multi-DSP communication network.

A custom hardware implementation can provide the processing power required by the algorithm while consuming a reasonable and controllable amount of power. The initial hardware solution by [12] implemented a sound localization algorithm on a field programmable gate array (FPGA), which operates at 10 MHz and consumes an estimated 100 mW per microphone. This hardware approach can be further enhanced by implementing a higher precision algorithm in an application-specific integrated circuit that is capable of lower power consumption.

Section II of this paper presents the basics of sound localization theory, including the development of the algorithm used in this work. Section III provides an overview of the architecture of the TDOA test chip, including some of the design decisions, challenges, and tradeoffs. Section III also provides an overview of the test board utilized in testing the TDOA chip. Finally, the experimental test results are presented in Section IV.

II. Sound Localization Basics

The most popular sound localization method utilizes a pair of microphones to estimate the TDOA of a single-source sound signal between two microphones [10], [13]. A single TDOA between two microphones will constrain the location of the sound source to a hyperbola in two dimensions or a hyperboloid in three dimensions. The intersection of multiple hyperbolas or hyperboloids, from different microphone pairs, will constrain the location of the sound source to a single point.

The most common TDOA estimation method is known as the generalized cross correlation (GCC) [11]. Consider two microphones separated by a distance d and a single sound source placed in the vicinity of the microphones. The two microphones will receive signals $m_1(t)$ and $m_2(t)$, respectively. The two signals are ideally scaled and time delayed versions of each other. However, in practical environments, the microphone signals also contain external noise, reverberation, and microphone-induced noise. The microphone signals can be modeled as

$$m_1(t) = s(t) + n_1(t) \tag{1a}$$
$$m_2(t) = s(t+\tau) + n_2(t) \tag{1b}$$

where $s(t)$ and $s(t+\tau)$ are the received signals from the sound source, and $n_1(t)$ and $n_2(t)$ are the total noises associated with the two microphones. The TDOA to be estimated is τ. Attenuation of the sound source has been neglected in order to simplify the derivation. This simplification, as shown later, does not affect the precision of the estimate as the magnitude of each signal is discarded by the chosen algorithm.

The Fourier transforms of the received signals are $M_1(\omega)$ and $M_2(\omega)$, respectively. The TDOA estimate $\tilde{\tau}$ can be calculated using frequency cross correlation, by finding the delay (phase shift) that maximizes the cross correlation between m_1 and m_2, as given by

$$\tilde{\tau} = \arg\max_{\beta} \int_{\omega=-\infty}^{\infty} M_1(\omega)\overline{M_2(\omega)}e^{-j\omega\beta}\,d\omega. \tag{2}$$

The TDOA estimate $\tilde{\tau}$ is simply the β that maximizes the cross correlation.

Generalized or unfiltered cross correlation (UCC), as shown in (2), does not perform well in noisy and reverberant environments [14]. A frequency weighting factor $W(\omega)$ is introduced into the cross correlation formula to allow the TDOA estimates to be less sensitive to noise or reverberations, resulting in

$$\tilde{\tau} = \arg\max_{\beta} \int_{\omega=-\infty}^{\infty} W(\omega)M_1(\omega)\overline{M_2(\omega)}e^{-j\omega\beta}\,d\omega \tag{3}$$

where possible weighing functions are given by

$$W_{\mathrm{UCC}}(\omega) = 1 \tag{4a}$$
$$W_{\mathrm{ML}}(\omega) = \frac{|M_1(\omega)||M_2(\omega)|}{|N_1(\omega)|^2|M_2(\omega)|^2 + |N_2(\omega)|^2|M_1(\omega)|^2} \tag{4b}$$
$$W_{\mathrm{PHAT}}(\omega) = \frac{1}{|M_1(\omega)||M_2(\omega)|} \tag{4c}$$

denoting, respectively, unfiltered cross correlation (UCC), maximum likelihood (ML), and phase transform (PHAT). Among these, the PHAT weighting function is most suitable for reverberant environments [14].

In discrete time, TDOA estimates are calculated based on groups of samples, and these groups of samples are termed segments or windows. Larger segments offer more data on which to base TDOA estimates; however, this results in a longer calculation latency. A single-segment TDOA estimate using PHAT in discrete time is given as

$$\tilde{\tau} = \arg\max_{\beta} \sum_{n=-N/2}^{N/2} \frac{M_1(n)\overline{M_2(n)}}{|M_1(n)||M_2(n)|}e^{-j2\pi F_S n\beta/N} \tag{5}$$

where F_S is the sampling frequency, n is the index of the discrete time Fourier transform, and N is the number of samples

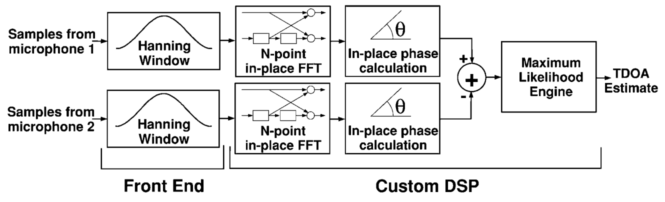

Samples from microphone 1
Samples from microphone 2
Hanning Window
Hanning Window
N-point in-place FFT
N-point in-place FFT
θ
In-place phase calculation
θ
In-place phase calculation
+
-
Maximum Likelihood Engine
TDOA Estimate
Front End
Custom DSP

Fig. 1. Data flow through the whole chip, showing division between front end and custom DSP.

in the segment being analyzed. Equation (5) can be further simplified as

$$\tilde{\tau} = \arg\max_{\beta} \sum_{n=0}^{N/2} \cos(\angle M_1(n) - \angle M_2(n) - 2\pi F_S n\beta/N) \tag{6}$$

where "$\angle$" denotes the phase angle of its respective argument.

The PHAT weight treats each frequency's contribution to finding the overall TDOA estimate equally by normalizing the magnitude of each frequency component to be equal to one. This normalization of magnitude allows PHAT to perform better in reverberant environments since the signal to reverberant noise ratio is similar over all frequencies. For these reasons, this technique was employed in our work.

III. Hardware Implementation

The hardware modules discussed in this section were described in the Verilog hardware description language using register transfer logic (RTL). The RTL description of the chip was synthesized by Synopsys® Design Compiler™ into a gate level net list. The resulting net list was floor planned and routed using Cadence® Silicon Ensemble®. All simulations and verification was performed using Cadence® Verilog-XL® logic simulator. The RTL logic is technology independent and can be easily ported onto an FPGA or a more aggressive CMOS technology.

The implemented PHAT TDOA estimation technique is employed for two microphone pairs. Each microphone signal undergoes amplification, antialiasing filtering, and sampling at 20 kHz with 8 bits of resolution before being processed by the TDOA chip. The TDOA chip interfaces directly to two analog-to digital converters (ADCs).

The overall chip architecture can be broken down into two major components: a preprocessing front-end and a custom digital signal processor (DSP) core, as shown in Fig. 1.

A. Preprocessing Front-End

In order to obtain robust TDOA estimates in real-time, all incoming microphone samples must be processed. In order to apply (6), a full segment needs to be buffered before the data can be processed. On-chip memory buffers a segment while another segment is being processed.

To allow for a tradeoff between TDOA estimate latency and accuracy, the size of the segment window can be selected at power up; available window sizes are 256, 512, and 1024 samples, corresponding to 12.8-, 25.6-, and 51.2-ms segment sizes, respectively. Segment sizes were chosen to be powers of two in order to simplify calculation of the fast Fourier transform (FFT).

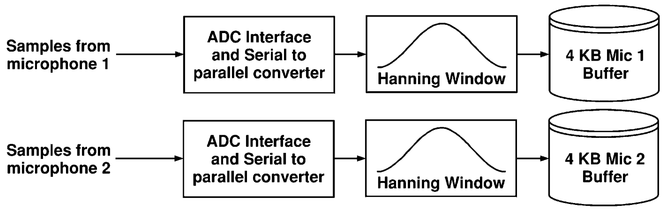

Samples from microphone 1
ADC Interface and Serial to parallel converter
Hanning Window
4 KB Mic 1 Buffer
Samples from microphone 2
ADC Interface and Serial to parallel converter
Hanning Window
4 KB Mic 2 Buffer

Fig. 2. TDOA chip input front end: Serial data is read from each ADC, multiplied point-by-point by a Hanning window, and stored into an input buffer.

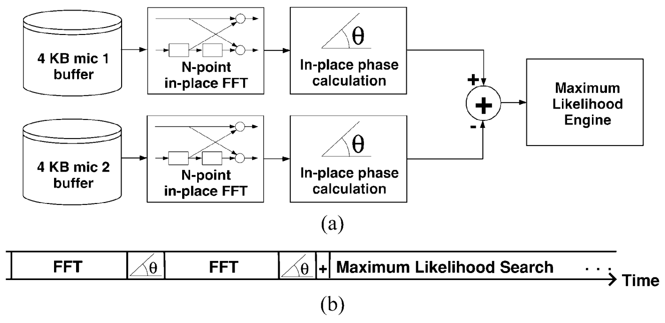

4 KB mic 1 buffer
N-point in-place FFT
In-place phase calculation
4 KB mic 2 buffer
N-point in-place FFT
In-place phase calculation
+
-
Maximum Likelihood Engine
(a)
FFT
FFT
Maximum Likelihood Search
Time
(b)

Fig. 3. (a) Data flow through different operation blocks in DSP core. (b) Representative timeline showing the order of and relative time required for the operations.

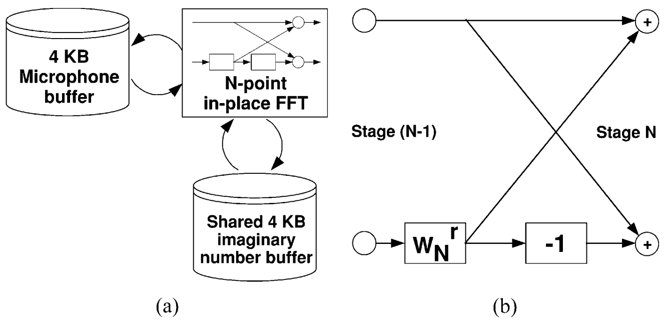


Fig. 4. (a) FFT block utilizes 4 kB of computational storage and the 4 kB shared memory block to perform an in-place FFT. (b) Two-point FFT butterfly architecture.

The maximum window size is mainly limited by the largest possible memory block that can be accommodated on the allocated silicon die size.

In order to make best use of silicon area, calculations are spread out sequentially in time so that the time required to calculate a TDOA estimate is slightly less than the time required to acquire a new segment to be processed. A segment size less than 256 samples proved to be impractical and too sensitive to errors [15].

As samples are acquired from the ADC, they are multiplied by a Hanning window and stored into an input buffer, as shown in Fig. 2. The input buffer has a fixed depth of 1024 samples. Each sample is 32 bits wide: 1 sign bit, a 25-bit mantissa, and a 6-bit exponent. The total memory block size required to store 1024 samples is 4 kB.

To reduce the total silicon area, a 4-kB memory block is shared between the two microphone channels; each microphone channel also has two dedicated 4-kB memory blocks. These two dedicated memory blocks switch between acting as an input buffer and as storage for computations. Initially, memory block A acts as an input buffer, whereas the previously buffered segment in memory block B is being used to compute a TDOA estimate. When memory block A—the input buffer—is full, memory blocks A and B switch tasks. Now, memory block A is being processed, while memory block B is buffering samples for a new segment. The memory block shared between the two microphone channels is used as temporary storage for imaginary Fourier transform coefficients.

B. Custom DSP Core

The chip is designed to accommodate a maximum microphone separation distance of approximately 50 cm with full range (-90° to 90°) sound localization capability. TDOA estimates can be obtained with a resolution of 0.1 samples or 5 μs through the processing gain of the transform domain. These design constraints result in the need to search though 601 possible TDOA estimates (βs) in order to find the one that results in the maximum cross-correlation.

The computations carried out in the DSP core can be broken down into several operations: FFT calculation, phase calculation, phase difference calculation, and maximal likelihood search. The data path for these operations is shown in Fig. 3(a). Fig. 3(b) shows the sequence of the operations. The most time-consuming operation is the search for the TDOA estimate, as each possible delay must be checked.

1) FFT Calculation: An N-point in-place radix-2 decimation-in-time FFT is calculated using each microphone's 4-kB computational storage and the shared 4-kB memory block. N represents the window size being used or the number of samples in the segment being processed. The computational memory block is used to store the real Fourier transform terms, whereas the shared memory block is used to store the imaginary terms.

The hardware implementation utilizes the two terminal input/output butterfly structure shown in Fig. 4(b), which consists of one complex multiplier and two complex addition/subtraction units. A finite-state machine controls the timing of this module; the function of this state machine is to read two datums from memory, process them using the aforementioned butterfly, and then write the resulting data back to the same memory location. Memory address generation and timing is also controlled by this module's finite state machine.

Although an extra memory block is required to store the intermediate imaginary results, an in-place FFT algorithm is much faster than direct calculation of the Discrete Fourier Transform (DFT). Direct DFT calculation requires $O(N^2)$ calculations, whereas the FFT algorithm requires $O(N \log_2 N)$ calculations. On the other hand, direct DFT calculation requires only one register to store the intermediate imaginary results for each term. The result can then be converted to polar coordinates and only the phase (angle), which is the only term required as per (6), would be stored.

The FFT for each microphone is performed sequentially to accommodate the sharing of a memory block for storing imaginary terms. Since only the phase of each FFT point is needed for further processing, the FFT terms are converted to polar representation. The magnitude is then discarded to free up the shared memory block.

2) Phase Calculation: Cartesian to polar conversion is performed using the Coordinate Rotating Digital Computer (CORDIC) algorithm [16]–[19]. CORDIC performs rotations of the Cartesian vector until the resulting vector is sufficiently close to the x-axis. By keeping track of the rotations required, the original phase of the vector can be sufficiently approximated along with the magnitude of the vector. Although the CORDIC algorithm introduces dilation of the vector magnitude, the phase is not affected.

The equations for the CORDIC rotations are

$$\begin{aligned} x_{i+1} &= x_i - y_i \cdot d_i \cdot 2^{-i} \\ y_{i+1} &= y_i + x_i \cdot d_i \cdot 2^{-i} \\ z_{i+1} &= z_i - d_i \cdot \tan^{-1} 2^{-i} \\ d_i &= \begin{cases} +1, & \text{if } y_i < 0 \\ -1, & \text{otherwise.} \end{cases} \end{aligned} \tag{7}$$

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 (x_0, y_0) are initialized to be the Cartesian coordinates of a vector, and z_0 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.

The CORDIC algorithm offers a hardware efficient method for calculating the phase of a Fourier transform element, since it utilizes only additions and bit shifts. However, a look-up table is required to store the values of the $\arctan(2^{-i})$ terms. The length of this look-up table is determined by the number of iterations required to reach the required precision: Approximately 1 bit of precision is gained for each rotation.

The hardware implementation of the CORDIC algorithm is quite simple; only six addition/subtraction units and four arithmetic shifters are required for this algorithm. A finite state machine controls the movement of data to and from memory as well as the iterative nature of the algorithm.

The space required for the digital logic that implements the CORDIC algorithm is not a function of the final precision required. In order to achieve a precision commensurate with the precision of the floating-point representation, approximately 20 CORDIC rotations are required; two rotations are performed per clock cycle in this work.

3) Phase Difference: A direct implementation of (6) can be achieved by a repetitive computation of the phase difference between the two microphones. Phase differences are calculated before the maximum likelihood search in order to reduce the number of overall calculations and memory traffic.

Phases are read from the two computational memory blocks and subtracted from each other with the result being written back to one of the computational memory blocks. Flow control and address generation is provided by a finite state machine.

4) Maximum Likelihood Engine: In order to mitigate estimation error introduced by the high-frequency components, only components in the lower frequency range of 0 to 5 kHz are utilized in the maximum likelihood search. Thus, (6) can be rewritten as

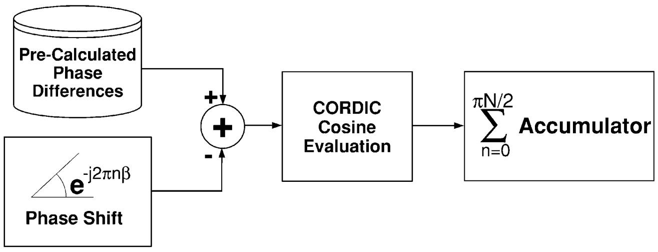

Pre-Calculated Phase Differences
+
-
$e^{-j2\pi n\beta}$
Phase Shift
CORDIC Cosine Evaluation
$\sum_{n=0}^{\pi N/2}$ Accumulator

Fig. 5. Data flow for maximum likelihood engine. For each β, the accumulator sums up the cosine of the phase error over all n. Four such operations occur in parallel. Each result is compared against the current maximum in order to find the β that results in the maximum likelihood.

$$\tilde{\tau} = \arg\max_{\beta} \sum_{n=0}^{\frac{N}{4}} \cos[(\angle M_1(n) - \angle M_2(n)) - 2\pi n\beta/N] \tag{8}$$

where β is now in samples instead of in seconds.

The term $2\pi/N$ is constant in all calculations. Moreover, by searching linearly through all possible β's, from β_{MIN} to β_{MAX}, one can restrict the starting angle offset to $-2\pi\beta_{\text{MIN}}/N$. For each step $\Delta\beta$, the term $-2\pi\beta_{\text{MIN}}/N$ increases by $2\pi\Delta\beta/N$. This approach eliminates the need for multiplication in calculating the phase angle for different β's. A similar approach is utilized, in order to avoid multiplications, for the evaluation of the changing phase term, in (8), for each different frequency point.

As shown in the data flow diagram of Fig. 5, another critical functional block for calculating the PHAT weighted correlation is the cosine evaluator. By modifying the CORDIC iteration formulas, the same algorithm utilized for phase evaluation can be used to calculate the cosine of an angle. The resulting formulas used are

$$\begin{aligned} x_{i+1} &= x_i - y_i \cdot d_i \cdot 2^{-i} \\ y_{i+1} &= y_i + x_i \cdot d_i \cdot 2^{-i} \\ z_{i+1} &= z_i - d_i \cdot \tan^{-1} 2^{-i} \\ d_i &= \begin{cases} -1, & \text{if } z_i < 0 \\ +1, & \text{otherwise.} \end{cases} \end{aligned} \tag{9}$$

Note the change in the rotation decision d_i. The algorithm rotates a vector from the x-axis by z_0 until z_i is sufficiently close to zero.

Initially, $x_0 = 1$ and $z_0 = \theta$, where θ is the argument of the cosine. By successively rotating $x_0 = 1$, with the intent to reduce z to 0, the cosine of θ can be obtained. In fact, $x = \cos(\theta)$ and $y = \sin(\theta)$ after the desired number of rotations. A total of 20 rotations are performed in order to achieve sufficient accuracy: Similar to phase evaluations, two rotations are performed every clock cycle. Dilation introduced by the CORDIC algorithm into the terms x and y can be compensated by adjusting the initial value of x_0; the dilation factor is a constant proportional to the number of rotations performed.

Once the cosine is evaluated, the result is added to previous cosine evaluation via an accumulator. The cosine of phase errors is summed over all frequency points, yielding a likelihood for a particular β. This likelihood is compared against a running

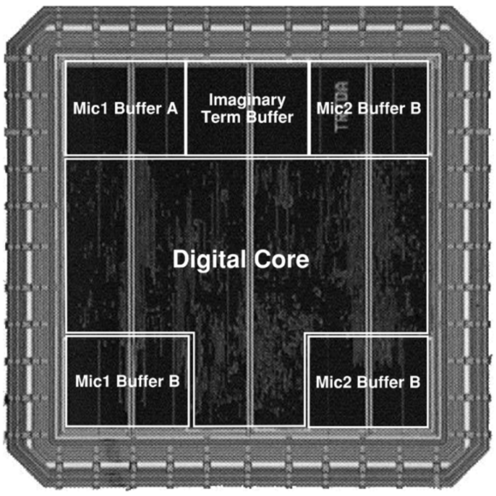


Fig. 6. Photograph of TDOA chip die. Locations of memory blocks and space occupied by digital logic is indicated on the die photograph.

maximum; if the calculated likelihood is the new maximum, then it is retained along with the β that corresponds to it.

The previous algorithmic modifications described in this section reduced the number of on-chip floating point multipliers to 4, which are exclusively used by the FFT block and front-end. By reducing the number of expensive operations and using small iterative operations, the area required for the digital logic is effectively reduced.

The iterative nature of cosine evaluation increases the time required for calculations, exceeding the time budget allocated for all calculations, which is only $\sim$50 ms for a 1024 sample window. Increasing the clock frequency in order to speed up the operations would not be a viable option since one would lose the ability to perform a complex floating-point operation in a single clock cycle. Therefore, four likelihood evaluations are performed in parallel.

C. Test Board and Prototype Chip

A photograph of the chip die is shown in Fig. 6. The die measures 2.5 by 2.5 mm. Approximately 42% of the silicon area is occupied by memory blocks. The 1.8-V chip core consumes an average power of 28.98 mW.

The test board housing the test chip also includes amplifiers, antialiasing filters, and analog-to-digital converters, which are all required to process the microphone analog signal before sound localization can be performed.

Two microphones are located on the board, approximately 19.8 cm apart. Headers are provided for the ability to connect external microphones.

A simple bipolar junction transistor (BJT) provides the first stage of amplification for each microphone signal, followed by two low-noise operational amplifiers. Antialiasing filtering is provided by a tenth-order switched-capacitor lowpass filter.

IV. Measurement Results

Three test scenarios are developed in order to measure the TDOA chip's performance. The first test scenario utilizes white

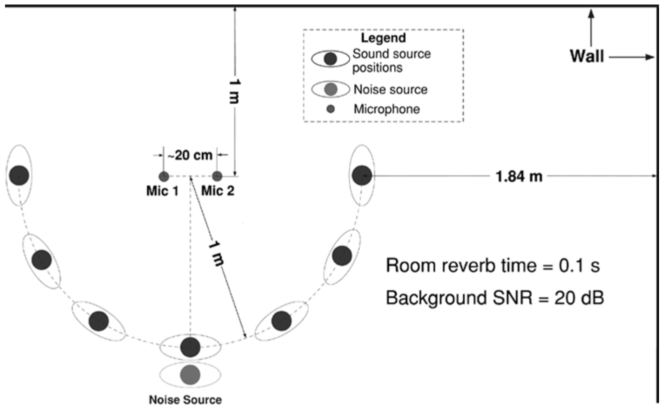


Fig. 7. Test setup used for testing the TDOA chip. The noise source was kept in front of the microphone pair, while the speaker of interest was placed in different positions in an arc about the microphones.

Gaussian noise as the source to be localized and as a noise source. The second test scenario utilizes various speech samples for both the source to be localized and for the noise source. These first two tests are carried out inside a simulated environment; MATLAB is used to delay and mix the signals and play them to the test board. The third test scenario places the sources in a lab environment in order to add reverberations and external noise sources to the test environment. All of these tests are performed using a PC running MATLAB. The PC is connected to the test board to receive TDOA estimates via the RS232 port. MATLAB is used to control the relative intensities of the sound source to be localized and the noise source. The tests, as well as their results, are described in more detail in the following sections.

A. White Gaussian Noise Tests

To verify the basic functionality of the algorithm, white noise sources are used for both the source to be localized and the noise source for the first pass/fail test. These tests are also used to make sure that the peripheral circuitry implemented on board are not causing errors in TDOA estimation. These tests are performed in a synthetic environment; MATLAB is used to delay and mix two separate sound signals and play the resulting audio signal. The analog audio signal is fed directly into the antialiasing filters tenth-order lowpass filter), bypassing the test board's front-end amplifiers.

The positions of the sound and noise source for the synthetic test setup is shown in Fig. 7; this test setup does not include reverberation effects. The two microphones are located 20 cm apart. A white Gaussian noise source is placed directly in front of the microphone pair. The white Gaussian noise source to be localized is placed at different positions in a 1-m arc around the microphones in increments of 30°. Seven distinct angles are tested for, as well as eight signal-to-noise (SNR) ratios between the noise source and the signal of interest.

The accuracy of each TDOA estimate is measured using its equivalent angular direction of arrival (DOA). The relationship between the DOA and the time delay or arrival is given by

$$\mathrm{DOA} = \sin^{-1}\left(\frac{\tilde{\tau}v}{d}\right); \quad \text{where } v \text{ is the speed of sound.} \tag{10}$$

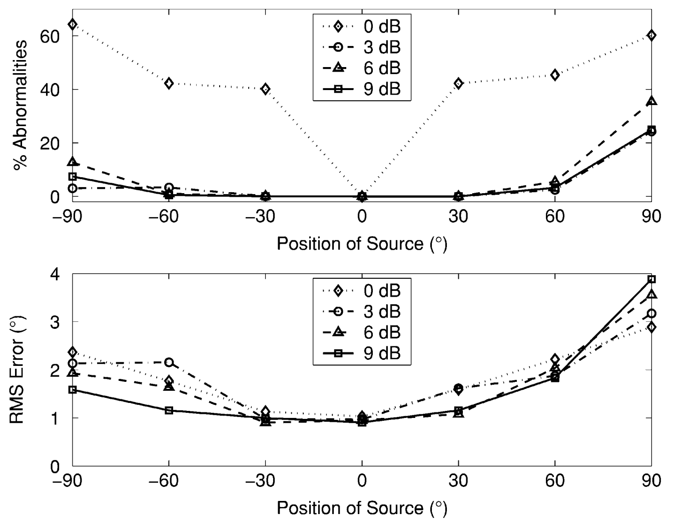


Fig. 8. TDOA estimate accuracy results obtained for a white Gaussian noise source placed at seven positions in the synthetic environment for different SNR ratios: 0, 3, 6, and 9 dB.

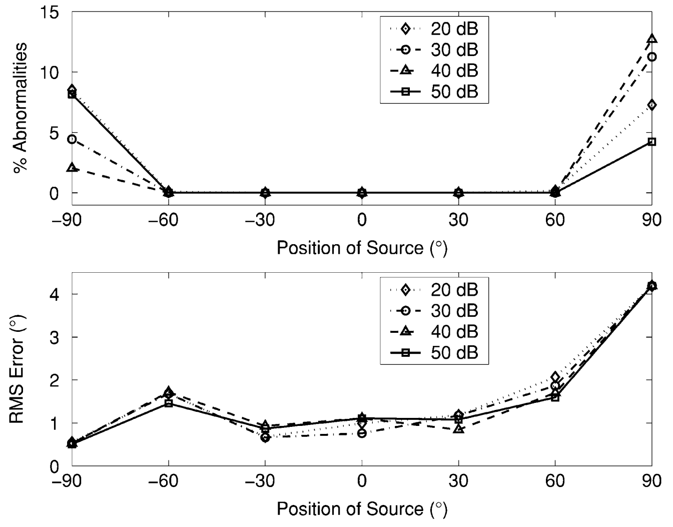


Fig. 9. TDOA estimate accuracy results obtained for a white Gaussian noise source placed at seven positions in the synthetic environment for different SNR ratios: 20, 30, 40, and 50 dB.

Results are collected using MATLAB via the RS-232 interface and compared against expected TDOA results. Measurements are collected in terms of DOAs, as defined by (10). The results shown in Figs. 8 and 9 are the corresponding percentage of abnormal measurements and the root-mean-squared error of the measurements over different angular orientations in the range $-90°$ to $90°$ for different SNR ratios.

Abnormal measurements are defined as estimates that have a DOA error (from nominal DOA) greater than 5°. Any measurements that fall within a DOA error of 5° are counted toward the RMS error. Since the microphone separation distance utilized is less than the initially assumed 50 cm, there exists a possibility that the TDOA might exceed the time required for sound to travel 20 cm. These TDOA estimates will cause the resulting DOA to be imaginary; these estimates are thus discarded and do not count toward the measurements collected.

To accumulate enough samples to make valid statistical measurements, 10 000 different TDOA estimates were taken for each source of interest position and SNR ratio.

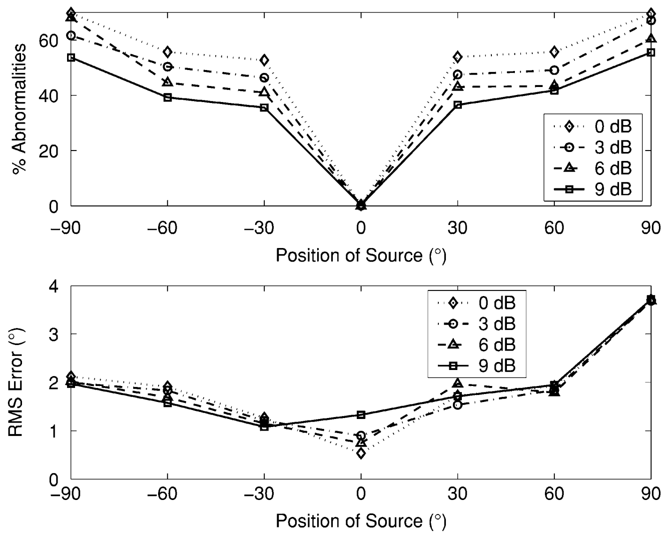


Fig. 10. TDOA estimate accuracy results obtained for prerecorded speech signals played at seven positions in the synthetic environment for different SNR ratios: 0, 3, 6, and 9 dB.

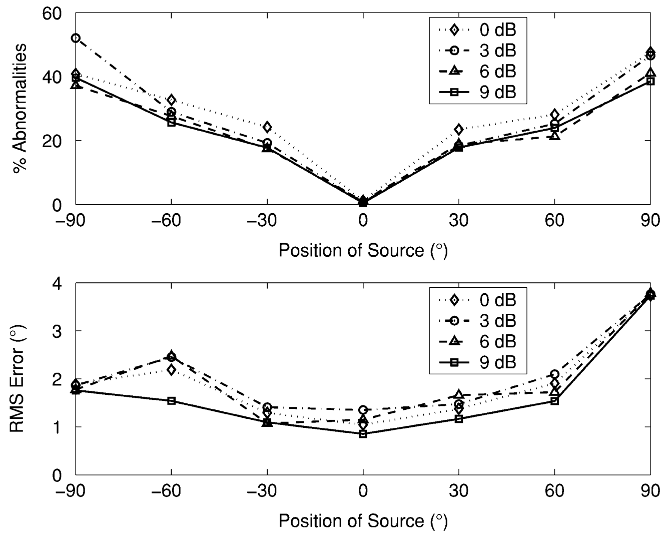


Fig. 11. TDOA estimate accuracy results obtained for prerecorded speech signals played at seven positions in the synthetic environment for different SNR ratios: 20, 30, 40, and 50 dB.

These tests indicate that the TDOA chip is functional and that it has an average estimation error of 1.25° RMS over various DOAs. Moreover, the number of abnormal measurements increases sharply below 3 dB SNR but otherwise remains relatively low, at less than 10%.

B. Voice Tests

The white Gaussian noise tests demonstrates the functionality of the sound localization algorithm and the precision of the localization engine. However, these tests do not provide insight into the performance of the intended application: sound localization. The test results presented here are obtained using recorded speech samples that are delayed and mixed, as for the previous tests, in order to measure the chip's localization accuracy using speech signals. Figs. 10 and 11 show the results obtained.

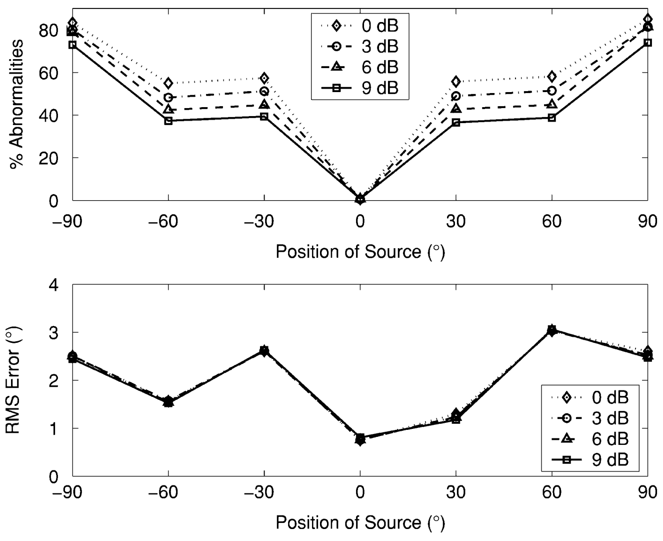


Fig. 12. TDOA estimate accuracy results obtained for prerecorded speech signals played at seven positions in a lab environment for different SNR ratios: 0, 3, 6, and 9 dB.

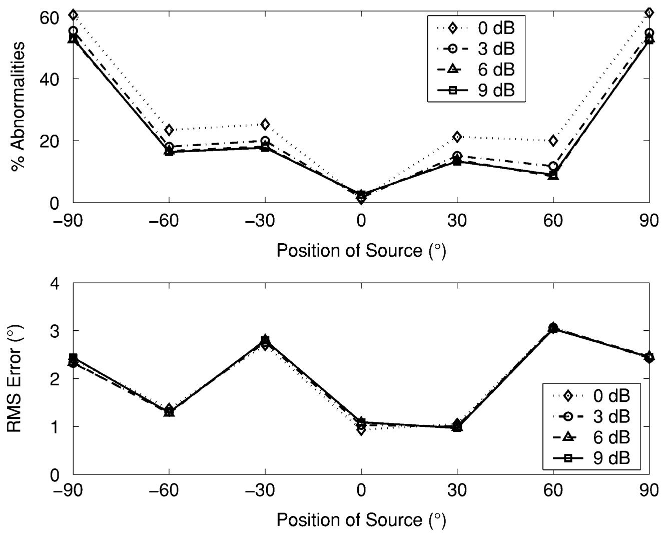


Fig. 13. TDOA estimate accuracy results obtained for prerecorded speech signals played at seven positions in a lab environment for different SNR ratios: 20, 30, 40, and 50 dB.

The noncontinuous nature of speech increases the number of abnormal measurements obtained. Moreover, normalizing different speech samples is difficult as the recorded speakers did not speak using the same tempo. This made normalizing the speech samples to obtain precise SNRs difficult. To offset this shortcoming, four different speech samples in 12 different permutations are used, thereby averaging out problems introduced by SNR variations.

These results show that the average RMS DOA error is 1.5° over various DOAs. On the other hand, the number of measurement errors increases gradually with decreasing SNR.

C. Tests in a Room Environment

The previous tests have shown the accuracy of the TDOA chip in a controlled environment; the following test measures the performance of the TDOA chip in a lab environment, which includes uncontrollable noise sources, reverberations, and temperature variations. In this test, computer speakers are placed in an arc around the test board, as shown in Fig. 7. The room's background SNR is approximately 20 dB, and the reverberation time is approximately 0.1 s.

The same speech samples, angles, and SNR are used as for the previous tests. Figs. 12 and 13 highlight the results obtained from these environmental tests.

The large number of abnormal measurements seen in these results is caused by the variance of the speed of sound with respect to temperature. In the synthetic environment, the delay of a sound source was controllable to within the 0.1 samples required. However, the delay in these tests was largely dependent on the precise placement of the computer speakers, the precise microphone separation distance, and the temperature dependance of the speed of sound. The speed of sound was estimated before each test was started; however, since these tests were carried out over several days, the temperature in the lab would vary considerably. This temperature variance is capable of producing a TDOA error of up to one sample.

The results (see Figs. 12 and 13) show an average RMS DOA of 2.2° with the same gradual increase in errors with decreasing SNR as seen in the previous results. Thus, the chosen algorithm performs well, even with the presence of reverberations in the room.

V. Power Utilization

This implementation of the PHAT TDOA algorithm was not specifically optimized for low power. However, algorithmic optimizations were considered in order to minimize power and space. In comparison to other implementations such as the FPGA implementation by [12], most of the power savings are a result of utilizing an aggressive target technology.

The TDOA chip when in reset requires 5.2 mA of current for its 1.8-V core. In normal operation, the chip requires 16.1 mA of current. Thus, the average power consumption is 9.36 mW during reset and 28.98 mW during normal operation. These power measurements are only for the chip's 1.8 V core; the power required for the 3.3-V I/O drivers is approximately 0.4125 mW.

To further reduce power consumption peripheral circuitry required by the test chip would have to be integrated on chip along with the custom DSP block; a system designed using off-the-shelf components would use more power than a single chip solution. Work is currently proceeding on reducing the overall system power consumption, both on an algorithmic level and implementation level. Our intent is to be capable of more rigorously controlling the power consumption of the system as a whole, thus designing a true low-power sound localization and sound separation platform.

VI. Conclusion

This work implements a phase transform sound localization algorithm in 0.18-μm CMOS technology. The accuracy of the resulting chip is demonstrated using three different test scenarios. An average 2.2° root-mean-squared direction of arrival error is measured when the chip was tested in a room environment using different combinations of prerecorded voice samples. The average power consumption of the resulting chip is 28.98 mW. Power reduction was a result of combining algorithmic strength reductions coupled with an custom design implemented in an aggressive technology.

REFERENCES

[1] P. Aarabi, G. Shi, and O. Jahromi, "Robust speech separation using time-frequency masking," in *Proc. ICME*, Baltimore, MD, Jul. 2003, pp. 741–744.

[2] G. Shi and P. Aarabi, "Robust digit recognition using phase-dependent time-frequency masking," in *Proc. ICASSP*, Hong Kong, Apr. 2003, pp. 684–687.

[3] J. Bitzer, K. U. Simmer, and K. D. Kammeyer, "Multimicrophone noise reduction techniques for hands-free speech recogition—A comparative study," in *Proceedings of ROBUST*, Tampere, Finland, May 1999, pp. 171–174.

[4] N. Sakamoto, W. Kobayashi, T. Onoye, and I. Shirakawa, "DSP implementation of 3D sound localization algorithm for monaural sound source," in *Proc. 8th IEEE Int. Conf. Electron., Circuits, Syst.*, 2001, pp. 1061–1064.

[5] T. Arikawa, N. Kanemaki, and H. Ichihara, "Evolution of computer-human interface design for the personal multimedia desktop conference system," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, Geneva, May 1993, pp. 1172–1176.

[6] J. Huang, T. Supaongprapa, I. Terakura, N. Ohnishi, and N. Sugie, "Mobile robot and sound localization," in *Proc. 1EEE/RSJ Int. Conf. Intelligent Robots Syst.*, vol. 2, Sep. 1997, pp. 683–689.

[7] P. Aarabi, "Robust sound localization using the formant phase transform," in *Proc. 5th IEEE Workshop Nonlinear Signals Inf. Process.*, Jun. 2001.

[8] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[9] P. Aarabi, "The Integration and Localization of Distributed Sensor Arrays," Ph.D. dissertation, Stanford Univ., Stanford, CA, May 2001.

[10] M. Brandstein and H. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in *Proc. ICASSP*, May 1997, pp. 375–378.

[11] C. H. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-24, no. 4, pp. 320–327, Aug. 1976.

[12] D. P. Nguyen, P. Aarabi, and A. Sheikholeslami, "Real-time sound localization using field-programmable gate arrays," in *Proc. ICASSP*, vol. 2, Hong Kong, Apr. 2003, pp. 573–576.

[13] J. DiBiase, H. Silverman, and M. Brandstein, *Microphone Arrays: Signal Processing Techniques and Applications*, M. Brandstein and D. Ward, Eds. New York: Springer-Verlag, Sep. 2001.

[14] T. Gustafsson, B. D. Rao, and M. Trivedi, "Analysis of time-delay estimation in reverberant environments," in *Proc. ICASSP*, vol. 2, Orlando, FL, May 2002, pp. 2097–2100.

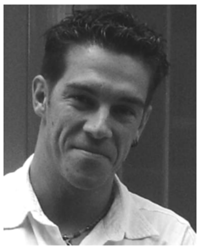
[15] P. Aarabi and A. Mahadavi, "The relationship between speech segment selectivity and time-delay estimation accuracy," in *Proc. ICASSP*, May 2002.

[16] R. Andraka, "A survey of CORDIC algorithms for FPGA based computers," in *Proc. ACM/SIGDA Sixth Int. Symp. FPGAs*, Monterey, CA, 1998, pp. 191–200.

[17] E. Grass, B. Sarker, and K. Maharatna, "A dual-mode synchronous/asynchronous CORDIC processor," in *Proc. Eighth Int. Symp. Asynchronous Circuits Syst.*, Apr. 2002, pp. 76–83.

[18] M. W. Kharrat, M. Loulou, N. Masmoudi, and L. Kamoun, "A new method to implement CORDIC algorithm," in *Proc. 8th IEEE Int. Conf. Electron., Circuits, Syst.*, vol. 2, 2001, pp. 715–718.

[19] S. Freeman and M. O'Donnell, "A complex arithmetic digital signal processor using CORDIC rotators," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, 1995, pp. 3191–3194.



David Halupka (S'99) received the B.A.Sc. degree in computer engineering (hardware option) from the University of Toronto, Toronto, ON, Canada, in 2002. He is currently pursuing the M.A.Sc. degree with Electronics Group at the University of Toronto.

He is currently working on a low-power implementation of sound localization and speech separation algorithms. He is also affiliated with the Communications Group, specifically the Artificial Perception Lab. During May 2000 to September 2001, he was involved in the design of enterprize routers and of a network security surveillance platform with Motorola Canada (currently Vanguard Managed Solutions). His research interests include development of digital signal processing algorithms for low-power applications.

Nebu John Mathai (S'00) received the B.A.Sc. degree from the Division of Engineering Science (Computer Option) in 2000, and the M.Eng. degree in electrical engineering in 2003, both from the University of Toronto, Toronto, ON, Canada. He is currently pursuing the Ph.D. degree at Texas A&M University, College Station.

From May 2000 to May 2003, he was a full-time digital ASIC engineer with Cogency Semiconductor, Toronto, where he was involved in the design of devices for powerline networking. His research interests include nonlinear dynamical systems and exotic implementation media.

Parham Aarabi (S'97–M'01) received the B.A.Sc. degree in engineering science (electrical option) in 1998 and the M.A.Sc. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 1999, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, Ca, in 2001.

He is a Canada Research Chair in Multi-Sensor Information Systems, an Assistant Professor with The Edward S. Rogers, Sr. Department of Electrical and Computer Engineering, and the founder and director of the Artificial Perception Laboratory. His current research focuses on multi-sensor information fusion, human-computer interactions, and VLSI implementation of sensor fusion algorithms.

Prof. Aarabi has received numerous teaching and research awards, including the Professor of the Year Award (2002 and 2004) and the IEEE Mac Van Valkenburg Early Career Teaching Award.

Ali Sheikholeslami (S'98–M'99–SM'02) received the B.Sc. degree from Shiraz University, Shiraz, Iran, in 1990 and the M.A.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, in 1994 and 1999, respectively, all in electrical and computer engineering.

In 1999, he joined the Department of Electrical and Computer Engineering, University of Toronto, where he is currently an Associate Professor. His research interests are in the areas of high-speed signaling, VLSI memory design (including SRAM, DRAM, FeRAM, and CAM), and circuits for DSPs. He is currently supervising three active research groups in the areas of ferroelectric memories, CAMs, and high-speed signaling. He has coauthored several journal and conference papers, as well as a book chapter on ferroelectric memories. He holds several patents on both ferroelectric memories and CAMs.

Dr. Sheikholeslami received the Best Professor of the Year Award in 2000 and 2002 by the popular vote of the undergraduate students in the Department of Electrical and Computer Engineering, University of Toronto. He has served on the Memory and the Technology Directions Subcomitties of the IEEE International Solid-State Circuits Conference (ISSCC) from 2001 to 2005. He presented a tutorial on ferroelectric memory design at the ISSCC 2002. He is a registered Professional Engineer in the Province of Ontario.