Sound Source Localization in 360 Degrees Using a Circular Microphone Array

Mohamed Gaber, Tamer Elnady and Adel Elsabbagh
Center for Sound, Vibration and Smart Structures (CVS3), Ain Shams University, Cairo, Egypt.

Summary
Sound source localization is useful in several applications such as noise activated cameras used in surveillance or used for video conferencing in meeting rooms. These cameras provide many advantages over a regular camera which keeps shooting in a certain direction. They can rotate and look in the direction of the sound source, then start recording. Large areas can be covered with minimum number of cameras. In this paper, sound source localization is performed using 15 free field microphones arranged in a circular array. All microphones acquire the sound signals simultaneously. From these signals, the Time Difference of Arrival (TDOA) between the microphones is measured. Using trilateration technique, the position of a sound source can be pin pointed. Trilateration calculations are very simple which make them ideal for real time applications where the computation time is of critical importance. The experimental setup includes 15 high precision microphones, connected to a National Instruments' PXI Data Acquisition system. A program is written in LabVIEW to manage the data acquisition, TDOA estimation, trilateration calculation and the results display. A rotary mechanism is attached on top of the microphone holder with a stepper motor to point at the sound source and a character LCD to show the calculated values. The system is validated for a variety of angles and radial distances from the center of the array. The system is then ported to single board RIO (sbRIO) to act stand-alone, using low-cost commercial microphones which proved to give the same performance.

1. Introduction

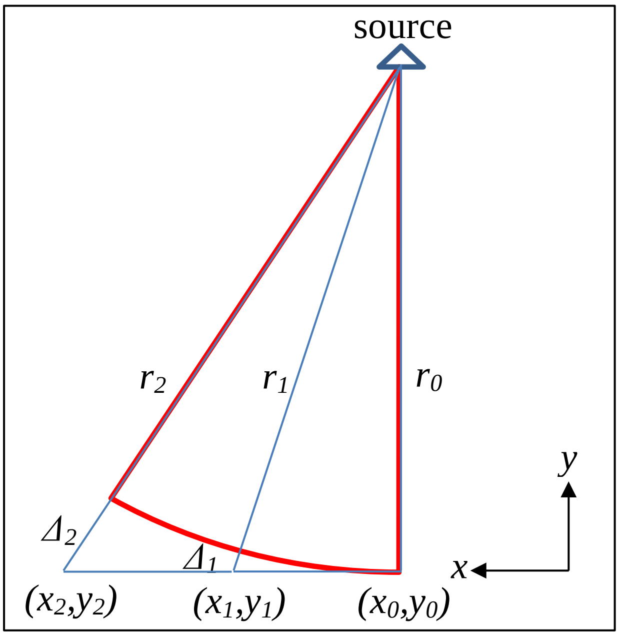
Many researchers have proposed different sound source localization techniques. It can be achieved using calculations in the frequency-domain or time-domain. The most basic technique that is used in the time domain is the Generalized Cross-Correlation (GCC) technique [1]; which performs cross-correlation between 2 microphones to determine the Time Difference Of Arrival (TDOA) between the microphones. From the TDOA, using trilateration calculations, the position of the sound source can be detected. The most important advantage about the GCC technique is its low computational cost which makes it ideal for real time applications [2]. However, there exist few disadvantages such as lower accuracy under reverberant conditions or low signal-to-noise ratio (SNR). To compensate for these setbacks, weighing techniques are used such as in phase transform (PHAT) technique [3]. Better accuracy is achieved on the cost of computation time [2]. The other major localization techniques are based on beamforming techniques, in which calculations are performed in the frequency domain [4]. Steered beamforming has the advantage of localization in a single step by steering the array to the source direction, but the major problem lies when trying to localize a broadband source, as it can only locate one frequency at a time; this significantly increases the computation time when trying to localize a broadband sound source [5][6]. To choose a suitable technique for locating a sound source in real-time, it is important to use a method that provides fast computation time. For this, GCC will be used to determine the source location. To further enhance the performance of the system Field Programmable Gate Array (FPGA) will be used as it can provide accurate timing and fast data acquisition which are both very important for the TDOA estimation. FPGA performance was tested in multiple papers and has proven to provide superior performance, especially in stand-alone systems [7][8]. The microphone array will be tested in an open field to verify the performance for different sound sources at different angles and distances from the array.

Copyright © 2018 | EAA – HELINA | ISSN: 2226-5147
All rights reserved

2. Theory of operation

In this paper, Time Difference Of Arrival (TDOA) technique is utilized to calculate the location of a sound source. This technique is chosen because of its simplicity which is necessary for real-time applications even for short bursts of sound [9].

Figure 1: Sound wave propagation towards 3 microphones.



By knowing the time difference "τ_n" between microphones "1" and "2" relative to "0", we can calculate the distances "Δ_1" and "Δ_2" by multiplying by the speed of sound c

$$\Delta_n = \tau_n * c \tag{1}$$

where n is the number of the microphone. For a source located at (x, y), and microphones located at (x_n, y_n) TDOA calculations are very simple and can be written as:

$$r_0^2 = (x - x_0)^2 + (y - y_0)^2 \tag{2}$$
$$r_1^2 = (x - x_1)^2 + (y - y_1)^2 \tag{3}$$
$$r_2^2 = (x - x_2)^2 + (y - y_2)^2 \tag{4}$$

"r_1" and "r_2" can be substituted in the above equations by

$$r_1 = r_0 + \Delta_1 \tag{5}$$
$$r_2 = r_0 + \Delta_2 \tag{6}$$

To give:

$$(r_0 + \Delta_1)^2 = (x - x_1)^2 + (y - y_1)^2 \tag{7}$$
$$(r_0 + \Delta_2)^2 = (x - x_2)^2 + (y - y_2)^2 \tag{8}$$

By combining equations 2, 7 and 8, one can get two equations to be solved simultaneously to give two sets of solutions for (x, y). Hence, a rule to be used to eliminate one of the solutions is needed.

To reduce the computation time of solving simultaneous non-linear equations, a fourth microphone can be added making the system over determined. By doing so, the system will be turned into a linear one. Which is much easier to be solved.

Adding a new microphone adds the equation:

$$(r_0 + \Delta_3)^2 = (x - x_3)^2 + (y - y_3)^2 \tag{9}$$

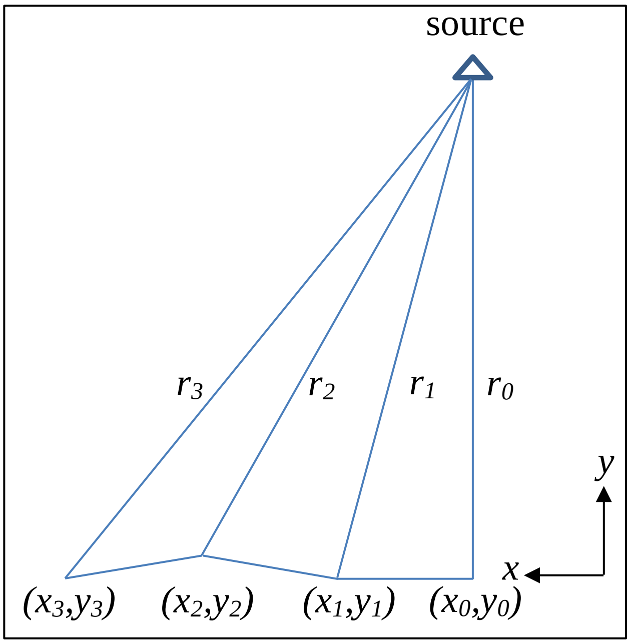
By doing so all the squared terms (x^2, y^2, r_0^2) will be eliminated and the final system can be described by the following equations:

$$(2x_1 - 2x_0)x + (2y_1 - 2y_0)y + 2\Delta_1 r_0 = x_1^2 + y_1^2 - x_0^2 - y_0^2 - \Delta_1^2 \tag{10}$$
$$(2x_2 - 2x_0)x + (2y_2 - 2y_0)y + 2\Delta_2 r_0 = x_2^2 + y_2^2 - x_0^2 - y_0^2 - \Delta_2^2 \tag{11}$$
$$(2x_3 - 2x_0)x + (2y_3 - 2y_0)y + 2\Delta_3 r_0 = x_3^2 + y_3^2 - x_0^2 - y_0^2 - \Delta_3^2 \tag{12}$$

If all the microphones were arranged in line (y_0 = y_1 = y_2 =*Zero*) the coefficient of the y-term in all the above equations will be equal to zero, making the system unsolvable. To avoid this, at least one microphone must be moved in the y-direction.

To obtain the time difference "τ_m", cross-correlation between signals received by the microphones has to be done relative to one reference microphone.

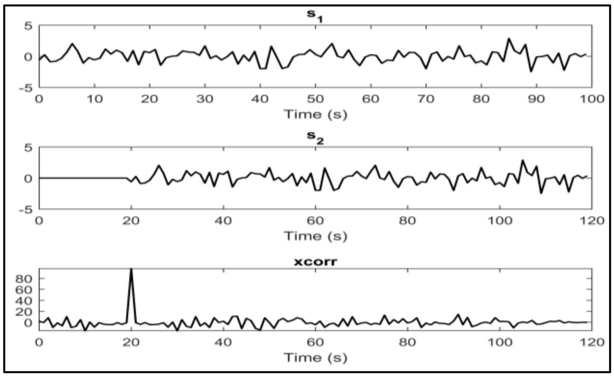
Figure 2: Sound wave propagation towards 4



3. Simulating cross-correlation

To verify the performance of the cross-correlation and to find the limitation of such technique, it is simulated using MATLAB and LabVIEW. Two white noise signals with a prescribed shift are simulated. Cross-correlation is performed to check if the calculated shift matches the prescribed one or not.

Figure 3: MATLAB simulation of cross-correlation (First white noise signal (top), second white noise signal with pre-scribed shift (middle), and cross-correlation (bottom).

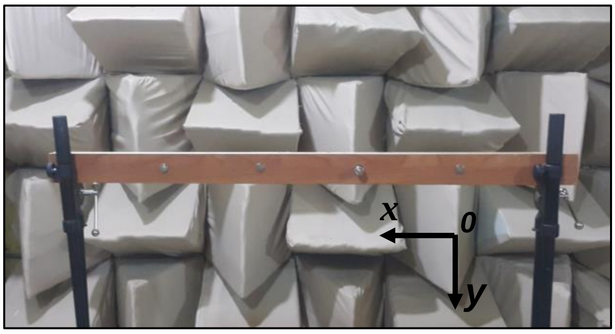


The results confirm the performance of cross-correlation in accurately calculating the shift in signals. One major limitation is the inability of cross-correlation to calculate the shift for periodic signals such as a sine wave. As it will always give a result between -1 and 1 cycle which is the minimum shift to make 2 periodic signals completely in phase. For this reason, only broad-band signals are used with this technique.

4. First prototype

The first prototype consisted of 4 "1/2 inch" free-field BSWA microphones mounted on a wooden holder with a spacing of 150 mm, one microphone is shifted 20 mm to the front. This prototype is used as a proof of concept to see how the array will perform. To simulate a free-field environment, all the tests were performed in a semi-anechoic room.
Figure 4 shows a picture of the 4-microphone array placed inside the semi-anechoic chamber. The second microphone from the right is shifted 20mm in the y-axis direction.
The microphones are connected to a compact DAQ (cDAQ) through a C-Series 4-channel analog input card (NI 9234). The cDAQ is connected to a computer on which a LabVIEW virtual instrument (VI) was made to acquire the data from the 4 microphones and perform the required calculations.

Figure 4: Linear microphone array inside a semi-anechoic chamber.



A sampling rate "f_s" of 51.2 kHz is used per channel. The system performed cross-correlation relative to the second microphone from the left. The cross-correlation calculations yield 3 sample differences "S_n" which had to be converted to time difference "τ_n".

$$\boldsymbol{\tau_n} = \boldsymbol{S_n}/\boldsymbol{f_s} \qquad (13)$$

After getting *"τ_n"*, equations 10, 11, and 12 can be solved to obtain the location of the sound source.

4.1. Results

The results obtained from the first prototype were very promising. A mobile phone emitting white noise is used as the sound source. Measurements at different locations are recorded.
The results showed an angle error of less than 2 degrees for an angle ±65 degrees from the reference microphone and up to 3.5 meters away from the array. The distance error is found to be less than 8% in a range up to 3.5 meters away from the array.

5. Stand-alone circular microphone array

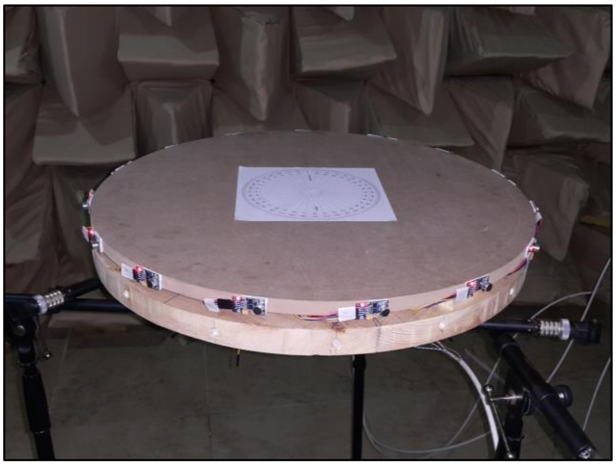
Several prototypes have been developed to reach the final stand-alone system. One change was made at a time to ensure the progress of development. The second prototype used 4 B&K "1/4 inch" microphones in a similar linear arrangement as prototype 1. The third prototype used 15 of the same B&K microphones arranged in a circular array with 600 mm diameter connected to PXI data acquisition system. The fourth prototype used 15 Arduino microphones

(omni-directional, sensitivity -46±3 dB and SNR (min.) 60dB) with the same arrangement and connected to PXI. Which is shown in Figure 5. The fifth prototype had a cRIO connected to 8 Arduino microphones to form half an array to test FPGA performance and to reduce the system cost. And the Final Prototype is the one described below using 15 Arduino microphones connected to single board RIO (sbRIO). To increase the array range to cover 360 degrees. More microphones had to be used. 15 free field microphones are arranged on a circular array with a diameter of 600mm. The microphones are equally spaced on a circle with a 126 mm spacing in between. To conclude if the Arduino microphones can replace the high precision ones or not, the factors affecting the accuracy of TDOA estimation had to be identified. Such factors include 1) Signal to Noise Ratio SNR 2) Sampling rate 3) Accuracy of sensor positions 4) Estimating sound speed accurately. All the factors mentioned do not depend on the microphone performance except for the SNR which can be related to the sensitivity of the microphone being used. After comparing the performance of both types of microphones, the high precision microphones were found to have lower noise. This will affect the range that the Arduino microphones can achieve because they are unable to detect faint sources due to their high internal noise. The decision was to go on with this choice even if it will affect the range of detection.
The second element that needed addressing is finding a suitable Data Acquisition System that can be easily used in a stand-alone system. That is to say it can be battery powered and has a small size, while maintaining high performance. To achieve this, Field Programmable Gate Array (FPGA) is used. Single board RIO (sbRIO) is chosen for this task as it combines both FPGA and real-time processor to handle fast data acquisition and calculations. General Purpose Inverter Controller (GPIC) will be attached to the (sbRIO) to add the necessary analog inputs with fast analog to digital converter (ADC) with up to 180kHz. sampling rate per channel and 16-bit accuracy. There are analog and digital outputs that will be used later to attach a stepper motor to point at the sound source and a character LCD to display the angle and distance.

5.1. Determination of the center microphone

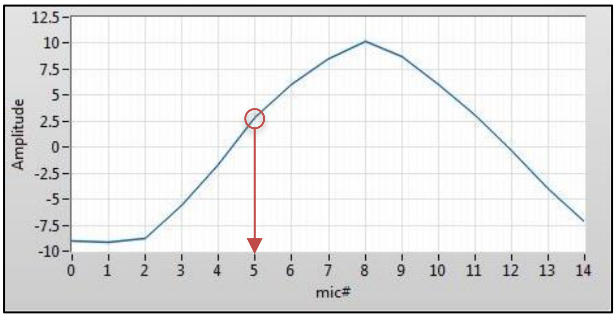
First step in determining the location of the source is to know which microphones are picking up the signals and which will not be participating in the calculations. To do so, the center microphone which is the microphone that is most facing the sound source needs to be determined.

Figure 5: Circular array of 15 Arduino microphones.



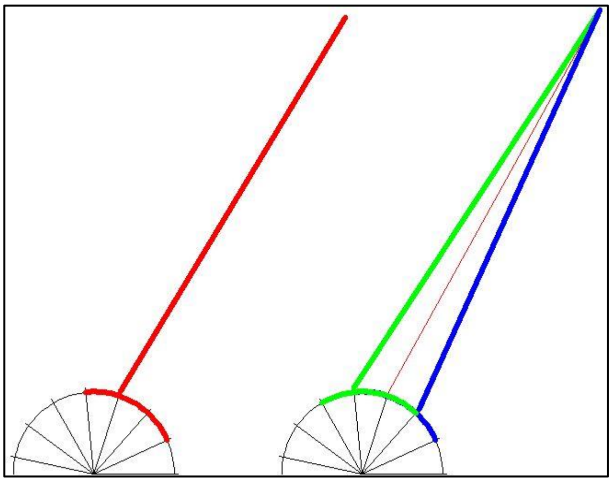
This is done by doing cross-correlation between every microphone and the one next, by obtaining these values. It will be noticed that half of the microphones have a negative cross-correlation which means they are leading the one they are being compared to, whereas the other half will have positive cross-correlation meaning they are delayed to the one they are being compared to. Figure 6 shows an example of determining the center microphone. The first positive cross-correlation for a microphone after all the negatives can be determined to be the microphone which is facing the sound source.

Figure 6: Determination of the center microphone from the cross-correlations between array microphones.



After determining the center microphone, similar calculations such as the ones done in the first prototype will be done to locate the sound source. To increase the accuracy of detection, two sets of 4-microphones will be used to detect the sound source. By using this method, the range detection can be improved as now the system acts as if there were two arrays working together to detect the sound source. However, this increases the computation time. Figure 7 shows the difference between using one set of 4-microphones versus two sets of 4-microphones to detect the position of the sound source.

Figure 7: Difference between using one and two sets of 4-microphones to detect the sound source location.



5.2. Acquisition parameters and data flow inside sbRIO

Data flow inside the sbRIO must be managed efficiently because there are two different systems running on board (FPGA and real-time processor). Both share the same resources and need to communicate. Direct Memory Access (DMA) First In First Out (FIFO) will be used to transfer the data between the FPGA and the real-time processor. This ensures no data points will be lost and the order of points will be preserved. DMA FIFO behaves differently on both the real-time processor and the FPGA. Table 1 shows these differences.

Table 1. DMA FIFO in FPGA and real-time processor

	FPGA	*Real-time*
Data acquisition speed	Very fast	Slow
Points	Written to memory Individually	Read from memory in bulks
Memory allocated for FIFO	Small typically 1024 elements	Large can be as large as 100s of thousands

Table 1 shows that to achieve balance in data transfer between FPGA and real-time processor, the FPGA will acquire the data with the specified rate, then write it to the DMA FIFO. After that the real-time processor will start reading from the DMA FIFO buffer in bulks of predetermined size to start making the calculations on the points. After experimenting with the acquisition rate and number of samples that the real-time processor read every time, it was found that the fastest acquisition rate that can be achieved by the system is 25 kHz. And the number of samples *"N_s"* that can be read by the real-time processor can be varied from 700 samples up to 2500 samples. This will determine how often the system will be taking a calculation.

$$N_s * \frac{1}{f_s} = 700 * \frac{1}{25000} = 28\ ms \quad (14)$$

$$N_s * \frac{1}{f_s} = 2500 * \frac{1}{25000} = 100\ ms \quad (15)$$

This parameter is left as a user defined value in the final user interface to control the update speed of the system depending on the nature of the sound source.

5.3. Additional components added to make the system stand-alone

For the system to be stand-alone, it must have some way to show the results being calculated. For this purpose, a stepper motor is added. The stepper motor has 200 steps meaning that the minimum angle it can move is 1.8 degrees. This can be further increased using a stepper motor driver with micro-stepping. It was found that using 1/2 step is the only reliable setting and any further micro-stepping isn't reliable and it could miss steps. Using half a step means a resolution of 0.9 degrees for the motor. To indicate the angle of the detected source and the distance of the source, a character LCD was attached to the system. To ensure that the stepper motor always starts at the zero position, an optical interrupt was added at the zero position; such that at system start-up the stepper motor will start rotating until it reaches the optical interrupt, at that point it starts taking inputs and starts moving.

The final addition to the system is a rechargeable battery to power all the systems. The sbRIO is rated to have a maximum power consumption of 28 W, and it runs on 9-30 VDC. The stepper motor runs on 12 VDC and draws 0.2 Amp. resulting in a 2.4 W power consumption. The microphones, motor driver and optical interrupt all run on 5 VDC provided by the onboard USB in sbRIO, meaning that they are already considered in the 28 W drawn by the sbRIO. A Li-ion battery with ($V_{bat}=12\ V$) with 10 Ah capacity (cap_{full}) is chosen. The minimum time (T_{min}) the battery will last is:

$$Amp_{cons} = {}^{Power}/_{V_{bat}} = \frac{30.4}{12} \cong 2.5Amp \quad (16)$$

$$T_{min} = {}^{cap_{full}}/_{Amp_{cons}} = \frac{10}{2.5} = 4\ hours \quad (17)$$

The calculations show that the battery can last at least 4 hours on 1 full charge.

Figure 8: Electric and logic connections.

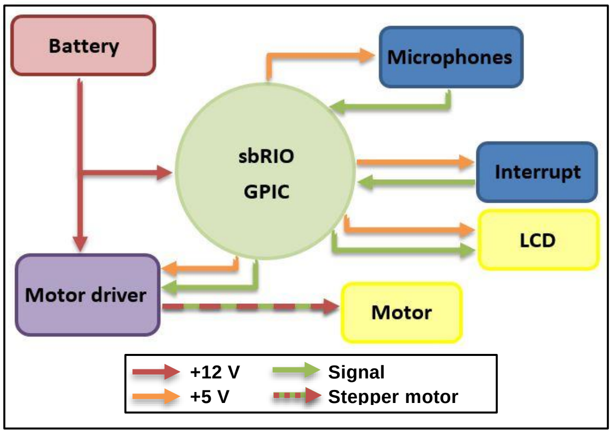


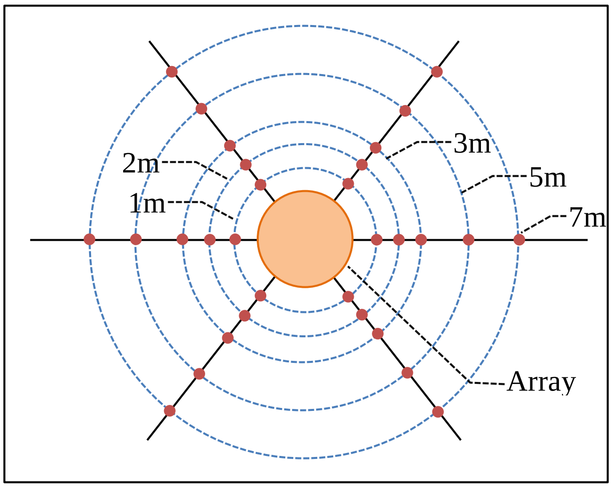
Figure 8 shows the color coded electric and logic connections of the entire system.
There are 3 loops running in parallel on the FPGA, one for the data acquisition running at 25 kHz., another for the motor which runs at 2 Hz. and last for the LCD running at 1 Hz. On the real-time side only one major loop is running which performs all the acquisition and calculations.

6. Experimental Results

The performance of the system was tested in the middle of a football field. A sound source was moved around the array at step angle of 60 degree at different distances. The calculated angle and distance were recorded and compared to the measured actual values to find the errors. A speaker connected to an amplifier acted as the sound source. Two different sound signals were used; a pink noise signal to simulate a continuous broadband sound source, and a gun-shot sound to simulate a broadband burst sound source.
Figure 9 shows the microphone array placed in the middle of a football field.
Figure 10 shows a schematic of the measurement setup and the points at which a reading was recorded relative to the microphone array.

Figure 9: Microphone array placed in the middle of the field.

Figure 10: The measurments locations relative to the array.



The results obtained from the array to localize a pink noise and a gun-shot recording signals emitted from a movable speaker were recorded at different distances (1, 2, 3, 5 and 7 meters) for the pink noise and (5, 7, 10, 12, 15, 25 and 30 meters) for the gun-shot playback and at different angles (0, 60, 120, 180, 240 and 300 degrees).

Figure 11 shows that the proposed array configuration has an error of ±1 degree for the continuous sound signal (pink noise) at up to 7 meters away which was limited by the power of the speaker, while Figure 13 shows the angle error for a short sound signal (single gun-shot playback) was ±2 degrees at up to 30 meters away from the array in all directions. Distance % error can be seen in Figure 12 for the pink noise source where the minimum error occurred at angles 120 and 300. And in Figure 14 for the gun-shot playback, the minimum error occurred at angles 60 and 240.

Figure 11: Absolute angle error for a pink noise source.

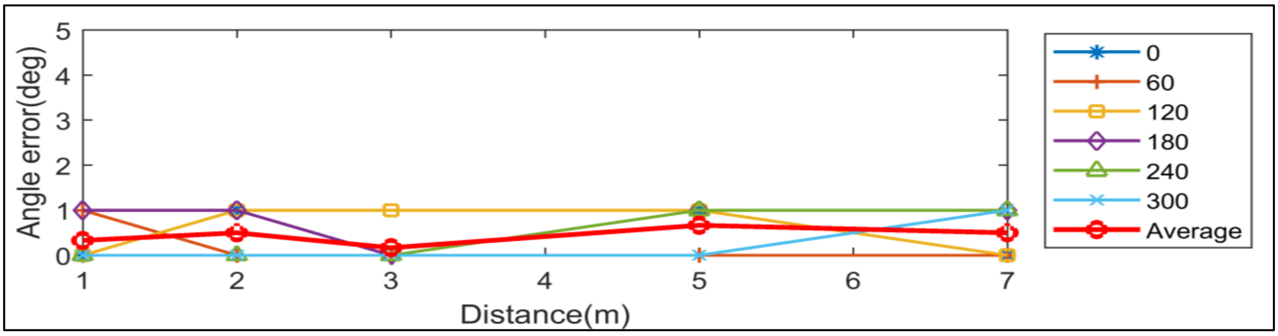


Figure 12: Distance % error for a pink noise source.

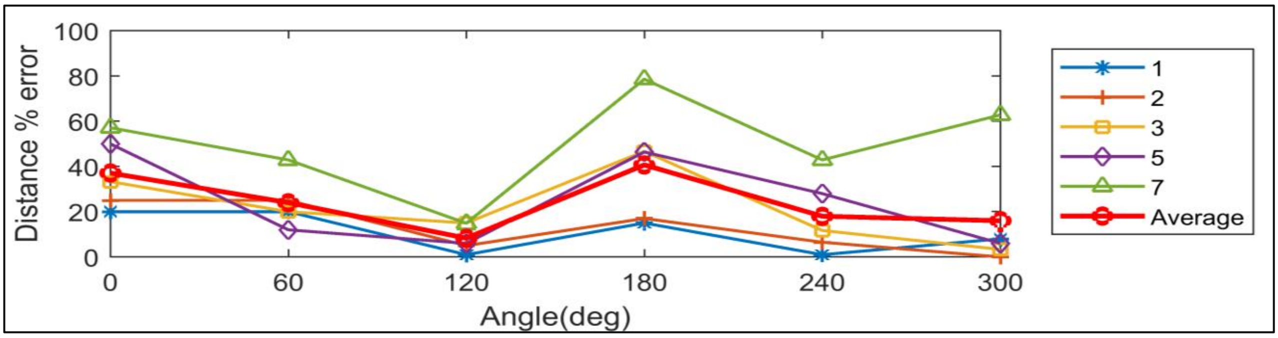


Figure 13: Absolute angle error for a gun-shot source.

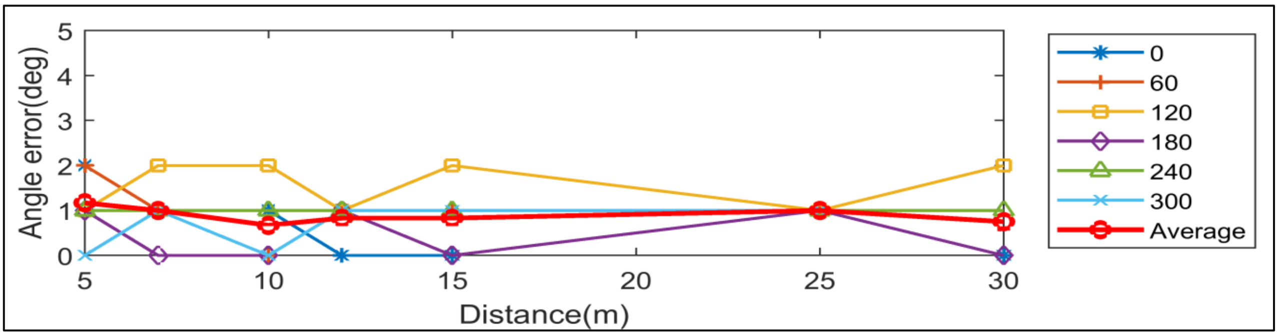
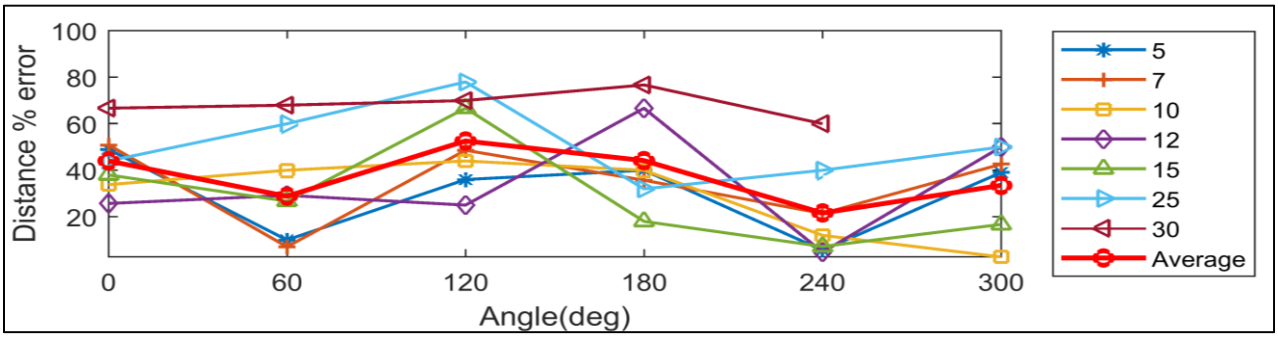
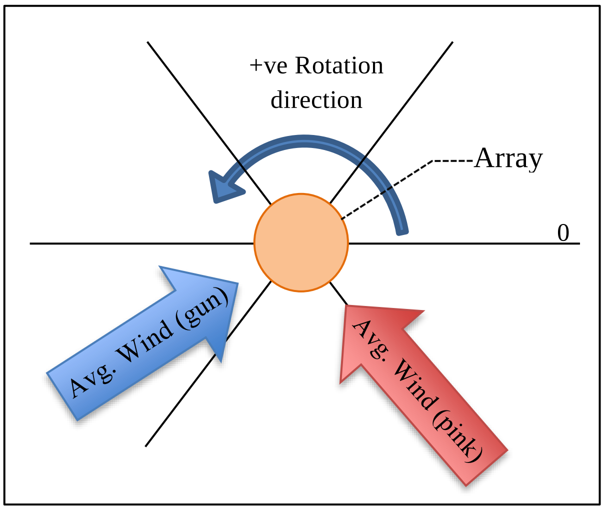


Figure 14: Distance % error for a gun-shot source.



The wind conditions at the time of measurement was recorded and it was found that during the pink noise measurements the average wind speed was 16 km/h with an angle of 300 relative to the array zero. While during the gun-shot playback the average wind speed was 18 km/h with an angle of 215 relative to the array zero. Figure 15 shows the wind direction relative to the array. After inspecting the wind conditions. It can be seen that the minimum distance % error occurs when the wind is in the same or the opposite direction to the measurement (Pink noise minimum distance % error at 120 and 300 degrees. While, gun-shot playback minimum distance % error occurred at angles 60 and 240 degrees which is closest to the wind direction of 215 degrees). However, the angle error was not affected by the wind direction or the distance. This shows that angle detection for this technique is robust regardless of the wind or distance for both sound sources used.

Figure 15: Average wind direction relative to the array during measurements.



Conclusion

In this paper a circular microphone array consisting of 15 free field Arduino microphones was used to locate a broad-band sound source in 360 degrees. The GCC time domain technique was chosen for its simplicity making it suitable for a real-time application. sbRIO and LabVIEW were used to produce a final stand-alone system. FPGA in the sbRIO provided fast acquisition rate of up to 25 kHz per channel which is necessary for capturing sound signals and performing cross correlation with high accuracy. The sbRIO was used to produce a stand-alone system which was tested in a free field to verify its performance. Two sound sources where used to test the system; a pink noise (broad-band continuous source) and a gun-shot playback (broad-band short burst source). The system had an angle error of ±1 degree for the pink noise source at up to 7 meters distance and ±2 degrees for the gun-shot playback source at up to 30 meters distance. The distance % error showed great dependency on the wind direction where the minimum error occurred when the wind direction was the same or opposite to the measurement direction. The Arduino microphones proved that they can achieve high accuracy with a fraction of the cost of high precision microphones in this application. The system was tested to run continuously for 7 hours on a single battery charge.

References

[1] S. Argentieri, P. Danès, P. Souères: A survey on sound source localization in robotics: From binaural to array processing methods. Computer Speech and Language 2015;Vol.34:87–112.

[2] Y. Zhao, X. Chen, B. Wang: Real-time sound source localization using hybrid framework. Applied Acoustics 2013;Vol.74:1367–73.

[3] T. Padois, F. Sgard, O. Doutres, A. Berry: Acoustic source localization using a polyhedral microphone array and an improved generalized cross-correlation technique. Journal of Sound and Vibration 2017;Vol.386:82–99.

[4] J. Valin, J. Rouat, L. Dominic: Robust Sound Source Localization Using a Microphone Array on a Mobile Robot. Proc IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2003:1228–33.

[5] F. Miao, D. Yang, J. Wen, X. Lian: Moving sound source localization based on triangulation method. Journal of Sound and Vibration 2016;Vol.385:93–103.

[6] X. Wang, B. Quost, J.D. Chazot, J. Antoni: Iterative beamforming for identification of multiple broadband sound sources. Journal of Sound and Vibration 2016;Vol.365:260–75.

[7] L. Petrica: An evaluation of low-power microphone array sound source localization for deforestation detection. Applied Acoustics 2016;Vol.113:162–9.

[8] P. Volgyesi, G. Balogh, A. Nadas, C.B. Nash, A. Ledeczi: Shooter localization and weapon classification with soldier-wearable networked sensors. Mobisys 2007:113.

[9] P. Song, C. Hao, J. Wu, C. Yang: Acoustic source localization using 10-microphone array based on wireless sensor network. Sensors and Actuators, A: Physical 2017;Vol.267:376–84.