Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 3, June 2026, pp. 2178∼2190
ISSN: 2302-9285, DOI: 10.11591/eei.v15i3.10523

An open-source low-cost TDOA-based sound source localization system

Mahmoud A. Alnaanah[1], Amir Abu-Al-Aish[2,3], Mohd H.S. Alrashdan[1], Mohammad Zayed Ahmed[1], Haitham A. Alasha'ary[4], Hamzah Hmeidi[5]

[1]Department of Electrical Engineering, College of Engineering, Al-Hussein Bin Talal University, Ma'an, Jordan
[2]Department of Communication Engineering, College of Engineering, Al-Hussein Bin Talal University, Ma'an, Jordan
[3]Department of Aviation Science and Management, Faculty of Aviation Sciences, Amman Arab University, Amman, Jordan
[4]Department of Computer Engineering, College of Engineering, Al-Hussein Bin Talal University, Ma'an, Jordan
[5]Department of Biomedical Engineering, School of Engineering and Computing, American International University, Al Jahra, Kuwait

Article Info

Article history:

Received Apr 22, 2025
Revised Jan 4, 2026
Accepted Mar 10, 2026

Keywords:

GStreamer
Octave
Open-source
Sound source localization
Time difference of arrival

ABSTRACT

Sound source localization (SSL) has many civilian and military applications, such as robotics, gunshot localization in nature reserves and warfare, surveillance, wild animal tracking, and rescue missions. One effective method of SSL is measuring the time difference of arrival (TDOA) for an array of microphones. The TDOA method requires designated hardware that is quite expensive and might require specific proprietary software; this makes SSL research expensive and difficult to develop software for, especially for hobbyists and educators. This paper presents a low-cost and easy-to-build SSL system that is built using 4 USB sound cards along with 4 microphones and a USB hub. The code for the system is made open for the public, and it is based on open-source applications, which are Linux, GStreamer, and Octave, to provide an open environment for studying and researching SSL. The localization process relies on finding the TDOA using generalized cross correlation (GCC) thresholding and finding the intersection point of three two-sheeted hyperboloids. The theoretical and experimental descriptions of the system are presented in this paper, along with some of the challenges, such as the microphones' timing and position calibration. Despite the low sampling rate and high noise of the used USB sound cards, the system was able to locate the sounds within a 2.4 m radius with a root mean square error (RMSE) of 7.23% and a mean error of 6.12%.

Corresponding Author:

Mahmoud A. Alnaanah
Department of Electrical Engineering, College of Engineering, Al-Hussein Bin Talal University
P.O. Box 71111, Ma'an, Jordan
Email: mahmoud.alnaanah@ahu.edu.jo

1. INTRODUCTION

Sound source localization (SSL) has many civilian and military applications [1]-[7], such as making robots and cameras move automatically toward the speaker [8]-[11], monitoring hunting gunshots and animal movements in wildlife reserves, localizing gunshots and bombing in military applications, and tracking underwater activities [12]-[16]. There are many SSL methods that rely on microphone arrays, such as time of arrival (TOA), time difference of arrival (TDOA), and direction of arrival (DOA). The TDOA method relies on measuring the difference in sound arrival time between the microphones by calculating the generalized cross

correlation (GCC) between the audio signals of two microphones. The TDOA method was selected in this research work because of its simplicity, and not requiring synchronization with the sound source, i.e., knowing the time of sound emission; however, TDOA requires synchronization among the microphones. In the TOA method, the sound arriving time to each microphone is measured, and by converting time into distance and using triangulation, the location of the sound source can be found. The TOA method is straightforward and has high accuracy, but it requires synchronization with the sound source. In DOA, the delay in time arrival between two microphones is measured using GCC, and the angle from which the sound originates is determined using beamforming. Compared to TDOA, DOA is more complicated and might require training, and it is less accurate in finding the absolute location of the sound source [17]-[20].

SSL systems require expensive and sophisticated hardware, and most of the time, they use proprietary software, which increases the cost of their research and complicates their code development, especially for hobbyists and educators. This paper presents an extremely low-cost SSL system that is built using 4 USB sound cards along with 4 microphones and a USB hub. The system is based on open-source applications, which are the Linux operating system, GStreamer for capturing audio signal, and GNU Octave for signal analysis and application interface. The code for the system was made open for the public to provide an environment to study and research SSL systems. The contributions of this paper are summarized as follows:

- Proposing a low-cost open-source SSL system that is easy to build, analyze, and improve.
- Presenting the theoretical and experimental details for the system to make it easy to build, study, and improve the system for researchers, hobbyists, and educators.
- Building the system based on an open-source environment and making the code for the system open to the public, which facilitates collaboration and improvement of the system.
- Addressing some of the practical challenges in the system design, such as signal latency and roughly positioned non-orthogonal microphones.

One of the earliest works to discuss and implement a low-cost SSL system that is built with off-the-shelf components was presented by James Scott and Boris Dragovic at Intel Research Cambridge [21], where indoor tagged and untagged SSL systems and their application in a 3D audio interface were discussed. An untagged SSL system was implemented using six low-cost microphones and six low-cost PCI sound cards. The positioning was based on (1), where the Levenberg-Marquardt method [22] is used to solve a non-linear set of equations to find the time-of-send tos and the sound position $soundpos$ with minimum error err , where known values are the time-of-arrival toa_i and the position $micpos_i$ for each microphone i. The system was able to locate finger clicking in 3D space with an error of less than 27 cm 90% of the time. One issue in this work is the large spacing between the microphones, with about 0.6 m vertically and 3.0 m horizontally, which makes it hard to implement for certain applications like robotics. Another issue is that the source code was not made publicly available, making it difficult to validate or improve upon the results. PCI sound cards remain expensive compared to USB alternatives and cannot be installed in laptops or single-board computers like the Raspberry Pi.

$$toa_i = tos + \frac{|micpos_i - soundpos|}{\text{Speed of sound}} + err_i \tag{1}$$

Michaud *et al.* [23] presented SmartBelt, which is a low-cost 8-microphone array mounted on a waist belt along with 15 haptic motors to indicate the DOA of the sound. The belt is meant to provide feedback about sound source angle, especially for people with hearing impairment. The microphones are connected to USB sound cards, which are connected to a Raspberry Pi 4 and powered by a power bank. The DOA calculation is based on the GCC with phase transform (GCC-PHAT). The system has a mean average error (MAE) of 2.90° and a haptic motor correct selection rate of 92.3%. The system lacks the ability for distance localization, and the code is not open-source.

Several low-cost, open-source sound localization and detection systems have been presented for wildlife monitoring, such as Conservation at Range through Audio Classification and Localization (CARACAL) by Wijers *et al.* [24]. Multiple CARACAL stations are distributed across the wildlife area. Each station features four microelectromechanical systems (MEMS) microphones connected to an advanced RISC machines (ARM) M4 Cortex microcontroller. Data are timestamped using global positioning system (GPS) and saved to micro-SD cards. The CARACAL system was used for gunshot localization within an array of 7 stations that have a separation of 500 m with an accuracy of 33.2±15.3 m. For animal call detection and localization, the

system demonstrated long-range (>1 km) localization of three different species, which are the Cape buffalo, the chacma baboon, and the spotted hyena. Even though each CARACAL costs about £150, many stations are needed to cover a large area, which makes the total cost quite high. Also, the system is designed for a specific task and requires predetermined fabrication.

Heath *et al.* [25] presented Multichannel Acoustic Autonomous Recording Unit (MAARU), which is a low-cost open-source outdoor acoustic monitoring and localization system. MAARU consists of a Raspberry Pi, a circular array of six Seeed Studio ReSpeaker microphones, a solar panel system with a rechargeable battery and a charge controller, and internet connectivity. MAARUs detect the azimuth for pure tones up to 6 kHz, and bird calls as far as 8 m away with an accuracy of ±10° range. For DOA estimation, HARKBird [26] is used to generate a multiple signal classification (MUSIC) power spectrum [27]. The system lacks the ability of 3D position localization, and it needs a specially fabricated board, which increases its cost and makes it not suitable for the general study of SSL systems, where the parameters of the system, like the spacing between the microphones, need to be changed.

As an improvement on the work by Heath *et al.* [25], Okamoto and Oguma [28] presented ChirpArray, which is an open-source low-cost microphone array for outdoor ecoacoustic monitoring and localization. ChirpArray consists of four microphones, which are used to estimate sound source direction and identify individual animal sounds. ChirpArray is built using a Sony Spresense microcontroller board, along with an extension board, Spresense four-channel microphone inputs, a global navigation satellite system (GNSS) module, and an internal real-time clock (RTC). Four electret condenser microphones (CME-1538-100LB, CUI Devices) are used for sound capturing. The system is equipped with batteries and a solar panel as a power source. After signal filtering, the MUSIC method [27] is used to estimate the azimuth of the sound source. Audio analysis and localization were carried out using the Pyroomacoustics open-source Python package [29]. For experimental testing, two ChirpArrays with a separation of 10 m were used to localize and classify different sounds from different distances. Azimuth detection remains consistent across varying ranges; however, the localization error scales with distance. Specifically, the average error increases from 0.42 m at 5 m to 4.25 m at 30 m, with intermediate values of 0.67 m and 2.10 m at 10 m and 20 m, respectively. Some of the shortcomings of the ChirpArray system are the inability to locate objects in the z-direction, a large error in distance estimation, and the system requires quite expensive hardware to build. In addition, fabricating the system on a single board might not be suitable for studying the different parameters of the system, such as microphone spacing.

From the literature review, a number of shortcomings can be observed. These shortcomings can be summarized as follows:

- Unavailability of the source code for the public with an open-source license, as in [21], [23]. In contrast, the code for the proposed SSL is open-source, and the description of theoretical and practical details is presented in this paper to facilitate building and improving the system.
- Lacking 3D localization of the sound source, where only the DOA is detected as in [23], [25], or the Z-direction is not detected, as in [24], [28]. The proposed system has the ability to localize the sound source in 3D space, which makes it suitable for applications like robotics.
- For all of the reviewed papers, the system is fabricated for a specific task and is not suitable for the general study of SSL, where the system parameters, such as microphone spacing, can be changed. The proposed system is designed for the general study of the SSL system, and avoids the need for custom PCB fabrication. This makes the system easy to assemble and ensures that components remain undamaged and reusable.

The rest of this paper is organized as follows: in section 2, the method is presented, which explains the theoretical and experimental details of the proposed system and the solutions for the different aspects of the system design, along with the application interface. In section 3, the actual measurements of the system are presented and discussed. Finally, the conclusion for the work in this paper is presented in section 4.

2. METHOD

As shown in Figure 1, the SSL process begins with calibrating the system to determine the exact positioning of the microphones and the delay between them. After recording the sound from all microphones, the TDOA for the sound source is found by GCC thresholding, then the TDOA for x, y, and z axes is converted to distance and used in constructing the equations for three two-sheeted hyperboloids. Localization is achieved by calculating the intersection point of the hyperboloids. The code used in this paper is made publicly available under the GNU Public License (GPL) V2 [30]. The SSL system proposed in this paper relies on open-source

applications, which are: Linux as the operating system, GStreamer for sound recording, and GNU Octave for analysis and application interface. The rest of this section will present the theoretical basis for the proposed SSL system along with its actual setup.

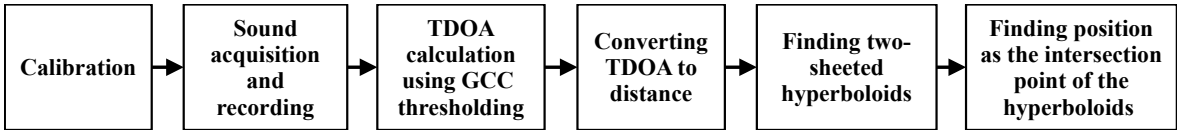


Figure 1. Simple block diagram of the localization process

The SSL process relies on measuring the TDOA between the reference microphone at the origin and the three microphones positioned along the x, y, and z axes, as shown in Figure 2. To illustrate the localization process, suppose there are two microphones, Mic_O and Mic_X. Mic_O represents the origin point, and Mic_X represents a point on the x-axis. The sound signals arriving at the two microphones are recorded, and a time interval T_S is taken at the same instant from the microphone recordings. T_S was selected to be 0.5 s, which is enough for calculating the GCC for the signal while keeping it short to reduce the computation time. The TDOA is located at the maximum point of the GCC, as shown in Figure 3. The maximum point can be found using thresholding.

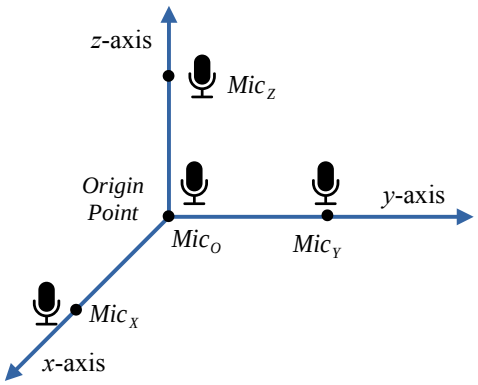


Figure 2. Microphone locations

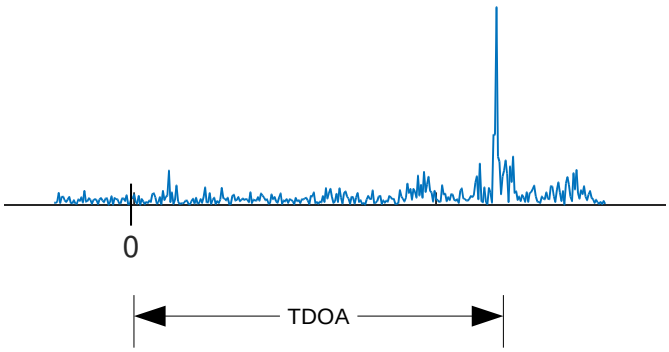


Figure 3. GCC plot

The number of samples N_S representing the TDOA is taken from the GCC and converted into a distance d according to (2), where F_S is the sampling frequency and S is the speed of sound in the air. S varies depending on the temperature and humidity of the air, and it was selected to be 340.29 m/s [31]. The sampling frequency in our case is 24 kHz.

$$d = \frac{N_S}{F_S} \times S \tag{2}$$

The distance $d_X = d_{X1} - d_{X2}$ is the difference in distance from the sound source to Mic_O (d_{X1}) and Mic_X (d_{X2}). As shown in Figure 4, all of the possible points in 3D space that have this distance difference make a two-sheeted hyperboloid. d_X is the distance between the intersection points of the two-sheeted hyperboloid with the x-axis. The focal points of the hyperboloid are at the position of Mic_O and Mic_X. The distance between the focal points and the hyperboloid intersection points with the x-axis is $d_X/2$. M_X is the distance between Mic_X and Mic_O. The right side of the hyperboloid corresponds to the positive values of d_X where $d_{X1} > d_{X2}$, and the left side corresponds to the negative values, where $d_{X1} < d_{X2}$. When $d_{X1} = d_{X2}$, the hyperboloid becomes a plane at the mid-point between Mic_O and Mic_X. The formula for the hyperboloid on the x-axis is shown in (3).

$$\sqrt{x^2 + y^2 + z^2} - \sqrt{(x - M_X)^2 + y^2 + z^2} = d_X \tag{3}$$

Utilizing four microphones, one at the origin and one along each coordinate axis, generates three perpendicular hyperboloids. The intersection point of these surfaces defines the sound source's location in 3D space, as illustrated in Figure 5. The equations for the y and z-axis hyperboloids are provided in (4) and (5), where M_Y and M_Z represent the distances from the origin to Mic_Y and Mic_Z, respectively. $d_Y = d_{Y1} - d_{Y2}$, where d_{Y1} and d_{Y2} are the distances between the sound location and microphones Mic_O and Mic_Y, respectively. $d_Z = d_{Z1} - d_{Z2}$, where d_{Z1} and d_{Z2} are the distances between the sound location and microphones Mic_O and Mic_Z, respectively.

$$\sqrt{x^2+y^2+z^2} - \sqrt{x^2+(y-M_Y)^2+z^2} = d_Y \tag{4}$$

$$\sqrt{x^2+y^2+z^2} - \sqrt{x^2+y^2+(z-M_Z)^2} = d_Z \tag{5}$$

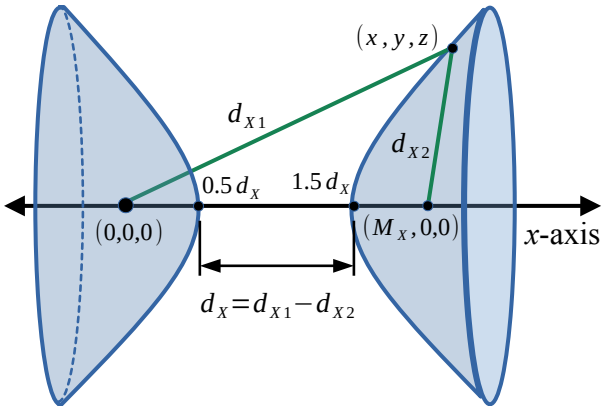


Figure 4. Two-sheeted hyperboloid

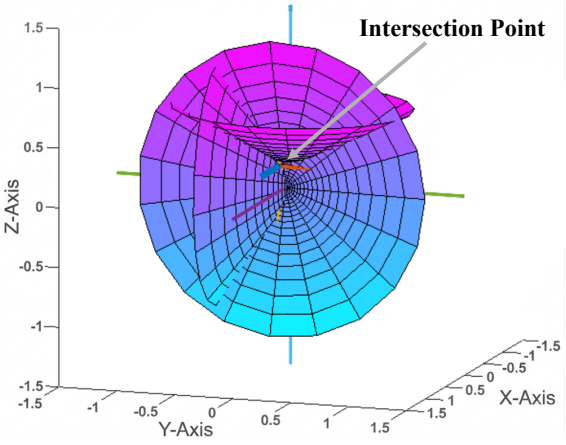


Figure 5. The intersection point of the hyperboloids

2.1. Equations for non-perpendicular axes

The proposed system accounts for potential misalignments of the microphone array, ensuring accuracy even when the microphones are not perfectly aligned with perpendicular axes. The four microphones make a non-uniform tetrahedron, as shown in Figure 6, where the points O, A, B, and C correspond to Mic_O, Mic_X, Mic_Y, and Mic_Z, respectively. The orientation of the microphones is assumed to be as follows: Mic_O is at the origin point (0, 0, 0), Mic_X is on the x-axis, Mic_Y is on the XY plane and has a horizontal angle θ_{YH} from the y-axis, microphone Mic_Z has a vertical, or zenith, angle θ_{ZV} from the z-axis and a horizontal, or azimuth, angle θ_{ZH} from the x-axis. θ_{YH} and θ_{ZH} are positive in the counterclockwise direction.

Misalignment of the microphones results in a rotation of the hyperboloids. To account for this, the hyperboloid equations are rotated according to (6) and (7). Figure 7 illustrates the rotation of the point (x,y) around the z-axis by an angle θ in the positive counterclockwise direction. The hyperboloid in the y direction is rotated around the z-axis by the angle θ_{YH} by replacing the x and y variables with x_α and y_α as in (8) and (9). The hyperboloid in the Z direction is rotated first around the y-axis by replacing the z and x variables with z_β and x_β according to (10) and (11), then it is rotated around the z-axis by replacing x_β and y with x_γ and y_γ according to (12) and (13).

The angles θ_{YH}, θ_{ZV}, and θ_{ZH} are calculated using the distances between the four microphones, i.e., the points O, A, B, and C, as shown in Figure 6. The distances between the microphones are d_{OA}, d_{OB}, d_{OC}, d_{AB}, d_{AC}, and d_{BC} as shown in Figure 6. The angle θ_{YH} equals $\theta_{AOB} - 90°$, where θ_{AOB} is the angle between the points A, O, and B, and it is calculated using the law of cosines shown in (14).

$$x' = x\cos\theta + y\sin\theta \tag{6}$$

$$y' = -x\sin\theta + y\cos\theta \tag{7}$$

$$x_\alpha = x\cos\theta_{YH} + y\sin\theta_{YH} \tag{8}$$

$$y_\alpha = -x \sin\theta_{YH} + y \cos\theta_{YH} \tag{9}$$

$$z_\beta = z \cos\theta_{ZV} + x \sin\theta_{ZV} \tag{10}$$

$$x_\beta = -z \sin\theta_{ZV} + x \cos\theta_{ZV} \tag{11}$$

$$x_\gamma = x_\beta \cos\theta_{YH} + y \sin\theta_{YH} \tag{12}$$

$$y_\gamma = -x_\beta \sin\theta_{YH} + y \cos\theta_{YH} \tag{13}$$

$$\theta_{AOB} = \cos^{-1}\left(\frac{d_{OA}^2 + d_{OB}^2 - d_{AB}^2}{2(d_{OA} \times d_{OB})}\right) \tag{14}$$

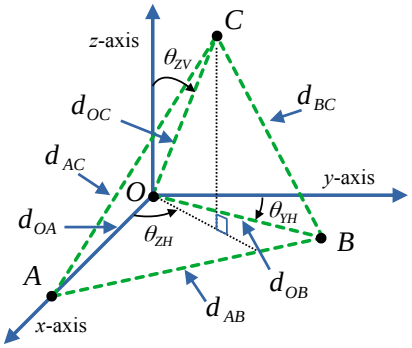


Figure 6. The tetrahedron formed by the microphones

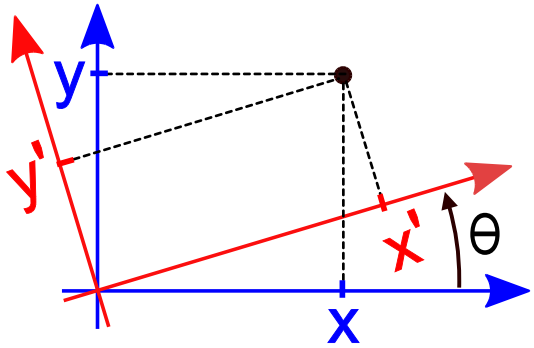


Figure 7. Axis rotation around the z-axis

The angle θ_{ZV} is calculated by finding the volume of a tetrahedron V using the Cayley-Menger determinant [32], as in (15), and substituting it in (16). The vectors $\overrightarrow{OA}$, $\overrightarrow{OB}$, and $\overrightarrow{OC}$ are formed by the points O and A, O and B, and O and C. The symbols (×) and (.) are the cross and dot products. The angle θ_{ZV} then becomes as in (17).

$$V^2 = \frac{1}{288}\begin{vmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & d_{OA}^2 & d_{OB}^2 & d_{OC}^2 \\ 1 & d_{OA}^2 & 0 & d_{AB}^2 & d_{AC}^2 \\ 1 & d_{OB}^2 & d_{AB}^2 & 0 & d_{BC}^2 \\ 1 & d_{OC}^2 & d_{AC}^2 & d_{BC}^2 & 0 \end{vmatrix} \tag{15}$$

$$V = \frac{1}{3}(\text{Area of the base})(\text{Height}) = \frac{1}{3}\left(\frac{1}{2}\left(\overrightarrow{OA} \times \overrightarrow{OB}\right)\right) \cdot \overrightarrow{OC} \tag{16}$$

$$\theta_{ZV} = \sin^{-1}\left(\frac{6V}{d_{OA} d_{OB} d_{OC} \cos(\theta_{AOB})}\right) \tag{17}$$

To find angle θ_{ZH}, the x, y, and z coordinates of point C are determined. The x coordinate is $x = d_{OC}\cos(\theta_{XOC})$, where θ_{XOC} is the angle formed between the vector $\overrightarrow{OC}$ and the x-axis, and it is found by applying the law of cosines to the distances d_{OA}, d_{OC}, and d_{AC}. The z coordinate of point C is found similarly as $z = d_{OC}\cos(\theta_{ZV})$. Since $d_{OC} = \sqrt{x^2 + y^2 + z^2}$, the absolute value of the y coordinate is $|y| = \sqrt{d_{OC}^2 - x^2 - z^2}$; however to find the sign of y, the projections of the two vectors formed by the points $(x, |y|)$ and $(x, -|y|)$ on the vector $\overrightarrow{OB}$ are found, and the one equals the projection of the vector $\overrightarrow{OC}$ on the vector $\overrightarrow{OB}$ is the one with the correct sign. The projection of the vector $\overrightarrow{OC}$ on the vector $\overrightarrow{OB}$ is $x = d_{OC}\cos(\theta_{BOC})$, where θ_{BOC} is the angle between the points B, O, and C, and it is found by applying the law of cosines to the distances d_{OB}, d_{OC}, and d_{BC}.

2.2. Calibration process

Significant latency exists between the microphones due to independent capture delays from each microphone and the lack of a unified clock across the separate hardware interfaces. A calibration process is performed to measure the latency and the distance between the microphones. The calibration follows a six-step sequence, where each step involves measuring the distance and latency between a specific pair of microphones. The measurement procedure is initiated by generating a sound source near the first microphone, directed toward the second, and seven consecutive readings of the distance difference are then recorded. This process is subsequently reversed by placing the sound source near the second microphone and directing it toward the first. The distance and latency for these readings are calculated using (18) and (19), where d_{12} and d_{21} represent the bidirectional distance measurements between the microphones. In this context, latency is expressed as a distance in meters (m). To ensure robustness against measurement noise, the median of the seven calculated values is used as the final result.

$$\text{distance} = \frac{d_{12} - d_{21}}{2} \tag{18}$$

$$\text{latency} = \frac{d_{12} + d_{21}}{2} \tag{19}$$

2.3. The application interface

The sound localization software [30] is built using GNU Octave, an open-source alternative to Matlab™. The main graphical user interface (GUI) of the software is shown in Figure 8. The GUI has the option to set the time sample duration T_S for calculating the GCC and an option to set the GCC ratio, which is used for thresholding. The GCC calculation is only accepted if its maximum value divided by its mean value is larger than the GCC ratio. From the *Session* menu, a new recording session can be initiated, or an already saved session can be loaded.

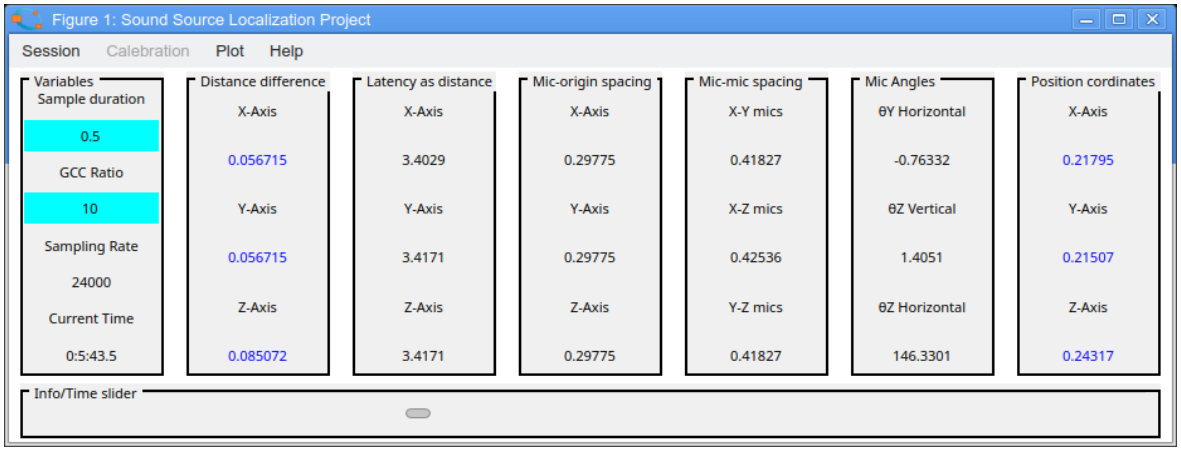


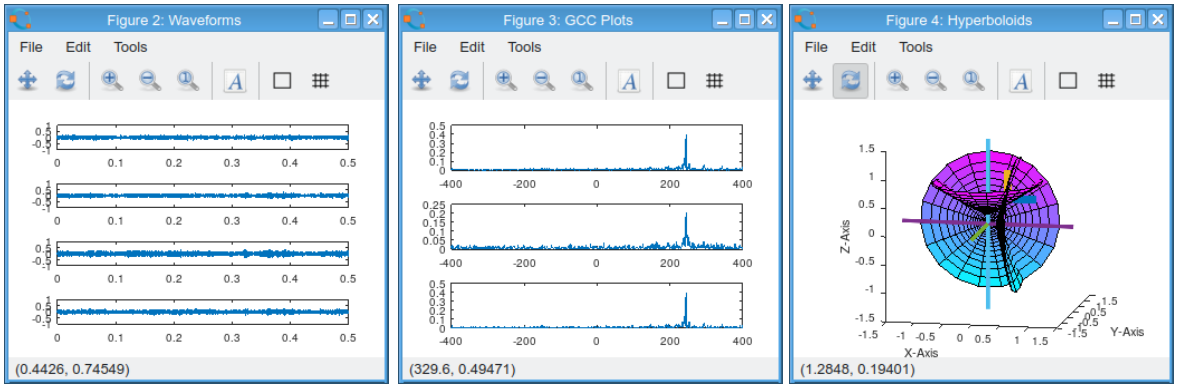
Figure 8. The application GUI

The option *Session>Start Recording* is selected to start a new recording session. The sound localization process starts by recording from all four microphones at once. Audio recording is performed using the *gst-launch* utility, which is a part of the *GStreamer* command-line framework for multimedia processing. After saving 4 audio files on the hard drive, one file for each microphone, T_S time interval is taken from each audio file. The GCC between the microphone at each axis and Mic_O is then calculated. For the GCC signal, the number of samples N_S between the origin point and the maximum point is found. N_S is only accepted if the ratio between the maximum and mean values of the GCC signal is larger than the GCC ratio. The distance difference between the microphones at each axis and the origin microphone is found according to (2), and then it is displayed in the GUI. The color of the displayed distance is blue if the GCC is accepted, and red if it is not.

The steps of the calibration process for each pair of microphones can be carried out by selecting the menu *Calibration*, which is only activated after starting the recording. After finishing all of the calibration

steps, the calibration information can be saved to the current session folder *ssloc_session*, which is located in the user's home, i.e., personal, folder. If the calibration process is complete, the latency values between the microphone on each axis and Mic_O are converted to distances and subtracted from the calculated ones. The coordinates of the location of the sound source can now be calculated and displayed in the GUI. The coordinates are accepted under two conditions. The first one is the measured distance difference, minus the latency, for all microphones, which should not be greater than the corresponding distance between the microphones. The second one is the thresholding for GCC calculations must be accepted for all axes. If the specified conditions are not met, the displayed coordinates are grayed out; otherwise, they appear in blue.

Figure 9 shows some plot windows from the application, which are: Figure 9(a) the plots of the sound samples, Figure 9(b) the GCC plot, and Figure 9(c) the plot of the hyperboloids. These plots can be displayed from the *Plot* menu. On the hyperboloid plot, a crosshair is displayed at the calculated coordinates to compare it with the intersection point of the hyperboloids. The recording is stopped by selecting *Session>End session* menu. The saved recordings, along with the calibration data, can be loaded later by selecting *Session>Load session* menu. The same localization procedures carried out during recording are applied to the loaded data, except that the time is controlled by a time slider that is visible only when a session is loaded.



(a) (b) (c)

Figure 9. Plots for: (a) the sound waveforms, (b) the GCC, and (c) the hyperboloids

2.4. The actual system setup

The proposed system was constructed using inexpensive off-the-shelf components, which are 4 microphones, 4 USB sound cards with a sampling frequency of 24 kHz, and a USB hub. The price of the components varies depending on their quality and the source of purchase, therefore it is not listed here. The calibration results are shown in Table 1. The distance between the microphones along each axis and the microphone at the origin is approximately 30 cm. This distance was selected so that it is enough to locate sounds within room distance while keeping the dimensions of the microphone array small enough and suitable for applications like robotics. In addition, this distance provides good sound levels for all microphones. The time interval T_S for calculating the GCC was selected to be 0.5 seconds, which is enough for calculating the GCC for the signal while keeping it short to reduce the computation time. The system was tested on Linux (Kubuntu version 22.04), using GNU Octave version 6.4.0, and GStreamer version 1.20.1. The actual setup for the system is shown in Figure 10. Figure 11 shows the hyperboloid plots for the points: Figure 11(a) (0.4,0.4,0.4), Figure 11(b) (0.4,0.0, 0.4), and Figure 11(c) (-0.2,0.2,0.2). The intersection of the hyperboloids is the location of the sound source in 3D space. The following section will discuss the experimental results for the system.

Table 1. Calibration results

Latency as a distance (m)		Mic-origin spacing (m)		Mic-mic spacing (m)		Mics angles (degree)	
x-axis	3.403	Mic_X	0.298	x-y Mics	0.418	θ_{YH}	$-0.76°$
y-axis	3.417	Mic_Y	0.298	x-z Mics	0.425	θ_{ZV}	$1.41°$
z-zxis	3.417	Mic_Z	0.298	y-z Mics	0.418	θ_{ZH}	$146.33°$

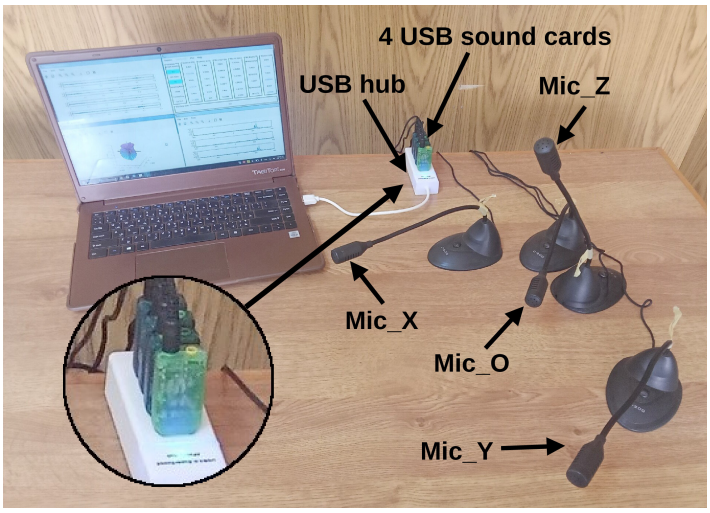
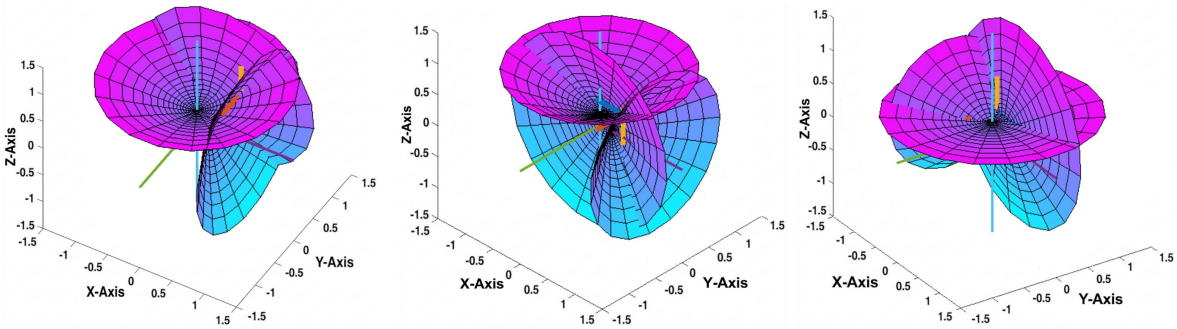


Figure 10. The actual system setup



(a) (b) (c)

Figure 11. Hyperboloid plots for the points: (a) (0.4,0.4,0.4), (b) (0.4,0.0,0.4), and (c) (-0.2,0.2,0.2)

3. RESULTS AND DISCUSSION

Table 2 presents the experimental results for the implemented system, comparing the ground-truth sound source locations with the coordinates measured by the system. The percentage localization error e is presented in Table 2, with its relationship to distance illustrated in Figure 12. As defined in (20), e is the ratio of the Euclidean distance between the measured and actual locations to the distance between the actual location and the origin. Here, (x_i, y_i, z_i) represent the actual coordinates, while $(\widehat{x}_i, \widehat{y}_i, \widehat{z}_i)$ denote the measured coordinates. The root mean square error (RMSE) for the measured locations, calculated via (21), is 7.23%, with a corresponding mean error of 6.12%.

$$e = \sqrt{\frac{(x_i - \widehat{x}_i)^2 + (y_i - \widehat{y}_i)^2 + (z_i - \widehat{z}_i)^2}{x_i^2 + y_i^2 + z_i^2}} \times 100\% \tag{20}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_i^2} \times 100\% \tag{21}$$

Table 2 shows that the measurements are more accurate for distances closer to the microphones. For example, for distances less than 1 m, the RMSE is 5.58%, and e is 4.88%. It can be noticed from Figure 12 that e generally increases as the distance from the microphones increases. The increase in e is due to the low

sensitivity, high noise, and low sampling frequency of the microphones. Also, inaccurate calibration will result in a high localization error for long distances. Increasing the distance between the microphones will reduce the effect of calibration error and increase the localization accuracy; however, this requires microphones with a higher sensitivity because the sound will be more attenuated at the microphones farther from the sound source. In addition, increasing the distance between the microphones might not be suitable for applications like robotics, where the space for the microphone array is limited. The main limitations of using low-quality microphones and separate USB sound cards are: high noise, high latency, and high sampling time drift between the microphones. These limitations can be solved by using a high-quality single-board sound system, but this will increase the cost of the SSL system.

Table 2. Actual positions, measured positions, and percentage error in distance

Actual position (m)	Measured position (m)	% error (e)	Actual position (m)	Measured position (m)	% error (e)
(0.2, 0.2, 0.2)	(0.206, 0.212, 0.208)	4.51	(0.0, 0.4, 0.4)	(0.031, 0.417, 0.396)	6.29
(0.2, 0.4, 0.2)	(0.210, 0.386, 0.198)	3.54	(0.0, -0.4, 0.4)	(0.024, -0.417, 0.396)	5.25
(0.4, 0.2, 0.2)	(0.409, 0.192, 0.209)	3.07	(0.4, 0.0, 0.4)	(0.412, 0.010, 0.383)	4.08
(0.2, 0.2, 0.4)	(0.213, 0.208, 0.414)	4.23	(-0.4, 0.0, 0.4)	(-0.381, 0.078, 0.390)	14.30
(0.4, 0.4, 0.4)	(0.400, 0.414, 0.391)	2.40	(-0.2, 0.2, 0.2)	(-0.201, 0.212, 0.191)	4.34
(0.6, 0.6, 0.6)	(0.635, 0.641, 0.575)	5.72	(0.2, -0.2, 0.2)	(0.211, -0.210, 0.203)	4.38
(0.8, 0.8, 0.8)	(0.866, 0.854, 0.763)	6.71	(-0.2, -0.2, 0.2)	(-0.206, -0.217, 0.210)	5.95
(1.0, 1.0, 1.0)	(0.916, 0.924, 0.902)	8.65	(-0.4, 0.4, 0.4)	(-0.416, 0.407, 0.386)	3.23
(1.2, 1.2, 1.2)	(1.076, 1.054, 1.110)	10.18	(0.4, -0.4, 0.4)	(0.422, -0.392, 0.387)	3.86
(1.4, 1.4, 1.4)	(1.142, 1.163, 1.140)	17.99	(-0.4, -0.4, 0.4)	(-0.392, -0.413, 0.421)	3.75

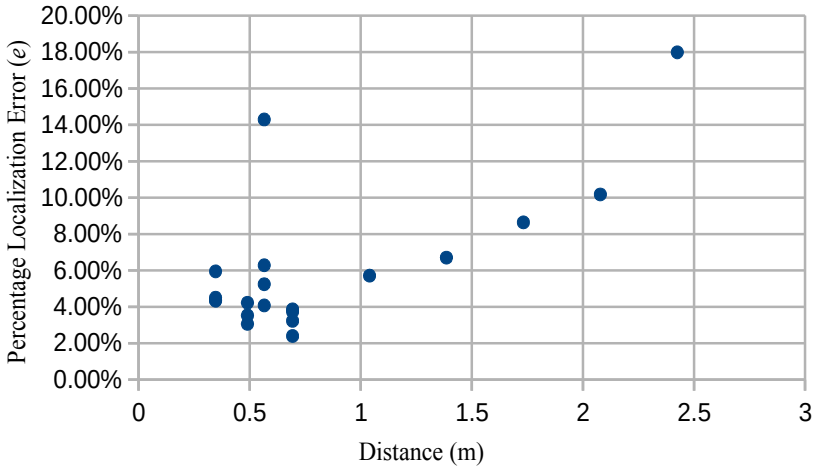


Figure 12. Percentage localization error versus distance

4. CONCLUSION

This research aims to develop a low-cost, open-source SSL system. Its primary advantage lies in the utilization of off-the-shelf components, which facilitate assembly without the necessity for custom PCB fabrication or expensive, proprietary hardware. The proposed system is suitable for hobbyists and educators where a general study of SSL systems is needed, and the parameters of the system, such as the spacing of the microphones, can be changed. Additionally, the used hardware can be disassembled and reused for other purposes. Another advantage is the use of open-source applications, which are the Linux operating system, GStreamer, and GNU Octave, which makes the system easy to study and improve. Despite the limitations of the system, such as high noise, high latency, and low sampling rate, the system was able to locate sound within a radius of 2.4 m with an RMSE of 7.23% and a mean error of 6.12%. The error here is the difference in distance between the measured and the actual location with respect to the actual distance between the sound source and the origin point. The small spacing of 30 cm between the microphones and origin point makes the system suitable for indoor applications like robotics. The proposed system can be used as a base for future work, such as using a high-sensitivity MEMS microphone array that is integrated with a single board interface

and synchronization with GPS signals, this will reduce synchronization and latency problem. In addition, the proposed system can be developed into a distributed internet of things (IoT) autonomous system with multiple units to cover a large area for applications like wildlife monitoring.

FUNDING INFORMATION

No funding was received for the this research work.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mahmoud A. Alnaanah	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Amir Abu-Al-Aish	✓	✓		✓	✓	✓			✓	✓				
Mohd H.S. Alrashdan	✓	✓			✓	✓				✓				
Mohammad Zayed Ahmed		✓			✓					✓				
Haitham A. Alasha'ary		✓			✓					✓				
Hamzah Hmeidi		✓			✓					✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

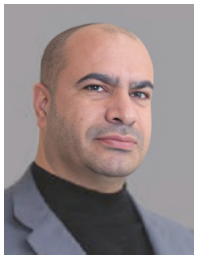
Data availability is not applicable to this paper.

REFERENCES

[1] G. Jekateryńczuk and Z. Piotrowski, "A survey of sound source localization and detection methods and their applications," *Sensors*, vol. 24, no. 1, pp. 1-25, 2023, doi: 10.3390/s24010068.
[2] A. Khan, A. Waqar, B. Kim, and D. Park, "A review on recent advances in sound source localization techniques, challenges, and applications," *Sensors and Actuators Reports*, vol. 9, p. 100313, 2025, doi: 10.1016/j.snr.2025.100313.
[3] R. Zaheer, I. Ahmad, D. Habibi, K. Y. Islam, and Q. V. Phung, "A survey on artificial intelligence-based acoustic source identification," *IEEE Access*, vol. 11, pp. 60078–60108, 2023, doi: 10.1109/ACCESS.2023.3283982.
[4] H. Y. Abuaddous *et al.*, "Repulsion-based grey wolf optimizer with improved exploration and exploitation capabilities to localize sensor nodes in 3d wireless sensor network," *Soft Computing*, vol. 27, no. 7, pp. 3869–3885, 2023, doi: 10.1007/s00500-022-07590-y.
[5] M. U. Liaquat, H. S. Munawar, A. Rahman, Z. Qadir, A. Z. Kouzani, and M. P. Mahmud, "Localization of sound sources: A systematic review," *Energies*, vol. 14, no. 13, pp. 1-17, 2021, doi: 10.3390/en14133910.
[6] D. Desai and N. Mehendale, "A review on sound source localization systems: D. desai, n. mehendale," *Archives of Computational Methods in Engineering*, vol. 29, no. 7, pp. 4631–4642, 2022, doi: 10.1007/s11831-022-09747-2.
[7] P.-A. Grumiaux, S. Kitić, L. Girin, and A. Guérin, "A survey of sound source localization with deep learning methods," *The Journal of the Acoustical Society of America*, vol. 152, no. 1, pp. 107–151, 2022, doi: 10.1121/10.0011809.
[8] D. Gala and L. Sun, "Moving sound source localization and tracking for an autonomous robot equipped with a self-rotating bi-microphone array," *The Journal of the Acoustical Society of America*, vol. 154, no. 2, pp. 1261–1273, 2023, doi: 10.1121/10.0020583.
[9] P. Dwivedi, G. Routray, D. K. Jha, and R. M. Hegde, "Improving source tracking accuracy through learning-based estimation meth ods in sh domain: A comparative study," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 8, pp. 3974–3984, 2024, doi: 10.1109/TAI.2024.3355323.
[10] M. Yasuda, S. Saito, A. Nakayama, and N. Harada, "6dof seld: Sound event localization and detection using microphones and

motion tracking sensors on self-motioning human," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seoul, Korea, Republic of, 2024, pp. 1411–1415, doi: 10.1109/ICASSP48485.2024.10446749.
[11] X. Qian, Z. Wang, J. Wang, G. Guan, and H. Li, "Audio-visual cross-attention network for robotic speaker tracking," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 550–562, 2022, doi: 10.1109/TASLP.2022.3226330.
[12] Y. Teng *et al.*, "Gunshots detection, identification, and classification: Applications to forensic science," *Science & Justice*, vol. 64, no. 6, pp. 625–636, 2024, doi: 10.1016/j.scijus.2024.09.007.
[13] S. J. Sankavaram, F. Azam, S. VS, M. Bonagiri, and K. R. Jetti, "A comprehensive review of deep learning techniques for direction of arrival for gunshot detection," in *2025 International Conference on Circuit, Systems and Communication (ICCSC)*, Fez, Morocco, 2025, pp. 1–6, doi: 10.1109/ICCSC66714.2025.11135241.
[14] Y. TAşDELEN, B. B. BAğCı, U. M. GöK, and O. T. ŞEN, "Development of a gunshot location detection system based on acoustic data," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, 2024, vol. 270, no. 4, pp. 7495–7500, doi: 10.3397/IN_2024_3967.
[15] B. R. Smith *et al.*, "Acoustic localisation of wildlife with low-cost equipment: lower sensitivity, but no loss of precision," *Wildlife Research*, vol. 49, no. 4, pp. 372–381, 2021, doi: 10.1071/WR21089.
[16] L. Lellouch, S. Haupert, and J. Sueur, "Sound source localization in a natural soundscape with autonomous recorder units based on a new time-difference-of-arrival algorithm," *Applied Acoustics*, vol. 235, p. 110648, 2025, doi: 10.1016/j.apacoust.2025.110648.
[17] H. T. Çalışkan and H. Karacan, "Sound source localization: A survey," *International Journal on Advanced Technology, Engineering, and Information System (IJATEIS)*, vol. 4, no. 2, 2025, doi: 10.55047/ijateis.v4i2.1741.
[18] M.-A. Chung, H.-C. Chou, and C.-W. Lin, "Sound localization based on acoustic source using multiple microphone array in an indoor environment," *Electronics*, vol. 11, no. 6, p. 890, 2022, doi: 10.3390/electronics11060890.
[19] A. Malmgren, "Sound source localization for an urban outdoor setting: A systematic review," M.S. thesis, Linnaeus Univ., Växjö, Sweden, 2022.
[20] G. K. J. Fischer *et al.*, "A systematic survey and analysis of angular-based indoor localization and positioning," *Authorea Preprints*, 2024, doi: 10.36227/techrxiv.172841698.83324280/v1.
[21] J. Scott and B. Dragovic, "Audio location: Accurate low-cost location sensing," in *International Conference on Pervasive Computing*, Springer, 2005, pp. 1–18, doi: 10.1007/11428572 1.
[22] S. P. Ahmadi, A. Hansson, and S. K. Pakazad, "Distributed localization using levenberg-marquardt algorithm," *Eurasip Journal on Advances in Signal Processing*, vol. 2021, no. 1, p. 74, 2021, doi: 10.1186/s13634-021-00768-w.
[23] S. Michaud, B. Moffett, A. T. Rousiouk, V. Duda, and F. Grondin, "Smartbelt: A wearable microphone array for sound source localization with haptic feedback," in *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, Busan, Korea, Republic of, 2023, pp. 1950–1955, doi: 10.1109/RO-MAN57019.2023.10309323.
[24] M. Wijers, A. Loveridge, D. W. Macdonald, and A. Markham, "Caracal: a versatile passive acoustic monitoring tool for wildlife research and conservation," *Bioacoustics*, vol. 30, no. 1, pp. 41–57, 2021, doi: 10.1080/09524622.2019.1685408.
[25] B. E. Heath *et al.*, "Spatial ecosystem monitoring with a multichannel acoustic autonomous recording unit (maaru)," *Methods in Ecology and Evolution*, vol. 15, no. 9, pp. 1568–1579, 2024, doi: 10.1111/2041-210X.14390.
[26] R. Suzuki, S. Matsubayashi, R. W. Hedley, K. Nakadai, and H. G. Okuno, "Harkbird: Exploring acoustic interactions in bird communities using a microphone array," *Journal of Robotics and Mechatronics*, vol. 29, no. 1, pp. 213–223, 2017, doi: 10.20965/jrm.2017.p0213.
[27] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986, doi: 10.1109/TAP.1986.1143830.
[28] R. Okamoto and H. Oguma, "Chirparray: A low-cost, easy-to-construct microphone array for long-term ecoacoustic monitoring," *Methods in Ecology & Evolution*, vol. 16, no. 2, 2025, doi: 10.1111/2041-210X.14474.
[29] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018, pp. 351–355, doi: 10.1109/ICASSP.2018.8461310.
[30] M. Alnaanah, "Sound source localization software," [Online]. Available: https://github.com/malnaanah/sound-localization. (Accessed: 15 Feb 2023).
[31] R. Dancila and R. Botez, "New flight trajectory optimisation method using genetic algorithms," *The Aeronautical Journal*, vol. 125, no. 1286, pp. 618–671, 2021, doi: 10.1017/aer.2020.138.
[32] F. Thomas and J. M. Porta, "Clifford's identity and generalized cayley-menger determinants," in *International Symposium on Advances in Robot Kinematics*, Springer, 2020, pp. 285–292, doi: 10.1007/978-3-030-50975-0_35.

BIOGRAPHIES OF AUTHORS



Mahmoud A. Alnaanah was born in Tafila, Jordan, in the 5th of April, 1979. He received his B.Sc. degree in Electrical Engineering from the University of Jordan in 2002 and his M.Sc. degree in Telecommunications from Mutah University, Jordan, in 2006. He received his Ph.D. degree in Electrical and Electronic Engineering from Staffordshire University, UK, in 2019. His current research interest is in image/signal processing and deep neural networks. Currently, he is an associate professor at Al-Hussein Bin Talal University, Ma'an, Jordan, where he started working in 2020. He can be contacted at email: mahmoud.alnaanah@ahu.edu.jo.

Amir Abu-Al-Aish received the B.S degree in Electronic Engineering from Yarmouk University in 2003. Received M.S. and Ph.D. degrees in electrical engineering [Control and Automation] from University Science of Malaysia (USM) in 2006 and 2013 respectively. Since 2013 until present, he works in AL-Hussein Bin Talal University, Ma'an, Jordan as Associate Professor. He can be contacted at email: Amir@ahu.edu.jo.

Mohd H.S. Alrashdan was born in Dair Abi Said, April 1, 1983, received the B.S. degrees in Biomedical Engineering from Jordan University of Science And Technology in 2006. He received his M.S. and Ph.D. degree in Electrical engineering- Microelectronics from National University of Malaysia (UKM) in 2007 and 2016 respectively. From 2016 until present, he worked as a full time lecturer in Al- Hussein Bin Talal University, Ma'an, Jordan. During 2007-2016, he finished his Ph.D. and focused in modeling, simulation, and fabrication of piezoelectric micro power generator for low frequency application using MEMS technology, during 2016-present he works as a full time lecturer and focused in power electronics and power systems analysis. He is a member in Jordan Engineers Association, cooperate with institute of micro engineering and nano electronics. He can be contacted at email: moh.alrashdan@ahu.edu.jo.

Mohammad Zayed Ahmed was born in Amman, August, 1980. He received his B.Sc. in Electrical Engineering from Baghdad University in 2002, his M.Sc. in Electronics System Design from University of Science Malaysia (USM) in 2006, and his Ph.D. in Microengineering and Nano Electronics from National University of Malaysia (UKM) in 2015. He worked as a full-time lecturer in Al- Hussein Bin Talal University, Ma'an, Jordan, from 2016 until present. His current research interest is in MEMS and sensors. He can be contacted at email: Mohammad.Z.Zayed@ahu.edu.jo.

Haitham A. Alasha'ary was born in Jordan in 1976. He received his B.Sc. degree in Electronics Engineering from Yarmouk University in Jordan in 2000, his M.Sc. Degree in Biomedical Engineering from the University of New South Wales in Australia in 2003, and his Ph.D. degree in Electrical and Computer Engineering from the University of Newcastle in Australia in 2010. His research interests include neural networks, fuzzy logic, neuro-fuzzy techniques, signal and image processing, automatic control, and parallel computing. He can be contacted at email: Haitham.alashaary@ahu.edu.jo.

Hamzah Hmeidi was born in Germany in 1987. He received his B.Sc. degree in Biomedical Engineer from Jordan University of Science and Technology in Jordan in 2010, his M.Sc. degree in Biomedical Engineering from Keele University in UK in 2013, and his Ph.D. degree in Biomedical Engineering from Keele University in UK in 2019. His research interests include, signal processing, rehabilitation engineering, and biomedical instrumentation. He can be contacted at email: h.hmeidi@aiu.edu.kw.