AI-based Simultaneous Audio Localization and Communication for Robots

Mjaid, Amjad Yousef; Prasad, Venkatesha; Jonker, Mees; Van Der Horst, Casper; De Groot, Lucan; Narayana, Sujay

DOI
10.1145/3576842.3582373

Publication date
2023

Document Version
Final published version

Published in
Proceedings - 8th ACM/IEEE Conference on Internet of Things Design and Implementation, IoTDI 2023

AI-based Simultaneous Audio Localization and Communication for Robots

Amjad Yousef Majid
Martel-Innovate
Switzerland
amjad.majid@martel-innovate.com

Casper van der Horst
Delft University of Technology
the Netherlands
C.vanderHorst@student.tudelft.nl

Lucan de Groot
Delft University of Technology
the Netherlands
L.J.deGroot-1@student.tudelft.nl

Mees Jonker
Delft University of Technology
the Netherlands
m.l.jonker@student.tudelft.nl

R Venkatesha Prasad
Delft University of Technology
the Netherlands
r.r.venkateshaprasad@tudelft.nl

Sujay Narayana
Delft University of Technology
the Netherlands
sujay.Narayana@tudelft.nl

ABSTRACT

Introducing Chirpy, a hardware module designed for swarm robots that enables them to locate each other and communicate through audio. With the help of its deep learning module (AudioLocNet), Chirpy is capable of performing localization in challenging environments, such as those with non-line-of-sight and reverb. To support concurrent transmission, Chirpy uses orthogonal audio chirps and has an audio message frame design that balances localization accuracy and communication speed. As a result, a swarm of robots equipped with Chirpies can on-the-fly construct a path (or a potential field) to a location of interest without the need for a map, making them ideal for tasks such as search and rescue missions. Our experiments show that Chirpy can decode messages from four concurrent transmissions with a Bit Error Rate (BER) of $BER \approx 2\%$ at a distance of 250 cm, and it can communicate at Signal-to-Noise Ratios (SNRs) as low as -32 dB while maintaining ≈ 0 BER. Furthermore, AudioLocNet demonstrates high accuracy in classifying the location of a transmitter, even in adverse conditions such as non-line-of-sight and reverberant environments.

ACM Reference Format:
Amjad Yousef Majid, Casper van der Horst, Lucan de Groot, Mees Jonker, R Venkatesha Prasad, and Sujay Narayana. 2023. AI-based Simultaneous Audio Localization and Communication for Robots . In *International Conference on Internet-of-Things Design and Implementation (IoTDI '23), May 09–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3576842.3582373

1 INTRODUCTION

The use of audio for inter-swarm-robots communication and localization has several advantages due to the local communication range of audio, its ability to bend around obstacles, and its attenuation by dense materials like walls. These characteristics enable the development of an *efficient* search and retrieval swarm robotic system, amongst other applications. A swarm of robots with microphones and speakers can form a "vector field" that spans *connected* paths toward a location of interest (Figure 1). Such a guidance mechanism can lead a specialized robot (or a person) to retrieve an object of interest, for example. Let us consider the example presented in Figure 1 to better understand the benefits of using audio as a pathfinder and guidance system. As depicted, `robot_0` finds an object of interest and sends a chirp to other robots to communicate that. Because audio does not penetrate walls easily, `robot_9` does not hear the chirp but `robot_1` does. This is desirable because the direction of arrival (DOA) of a sound signal points to its source through an open path (the red arrows in Figure 1 represent DOAs). `robot_1` relays the message to other robots. `robot_2` hears it but the walls block the audio message from reaching `robot_8` and `robot_9`. By keep relaying the message, it will eventually reach the specialized robot at the start location. The specialized robot then follows the DOA from one swarm robot to another to reach the found object and retrieve it.

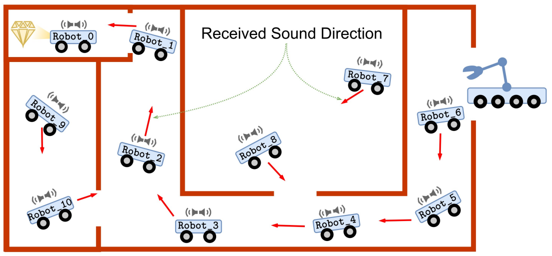


Figure 1: A swarm of simple robots find an object of interest and construct an audio-based potential field to guide a specialized robot to bring the object back.

Such a guidance system does not require the swarm robots to map their environment and synchronous the maps to localize things on a common map. Thereby, it lowers the cost of manufacturing

IoTDI '23, May 09–12, 2023, San Antonio, TX, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0037-8/23/05.
https://doi.org/10.1145/3576842.3582373

swarm robots and the technical difficulties of scaling a swarm system.

At this point, the reader might rightfully question why not use a more common communication medium such as radio transmissions (RF) and light. RF penetrates non-metal obstacles easily and therefore it is challenging to identify a traversable (or connected) path between a transmitter and receiver. However, audio signals get attenuated through walls, thus aids in avoiding too many cross talks. For double plasterboard walls, the power loss of an audio signal is 10 to 70 dB greater (based on the frequency) than that of an RF signal [14, 17, 19].

Light and mmWave-based methods, on the other hand, require a line of sight between the transmitter and receiver to function. Therefore, to ensure connectivity between the robots in an area with many obstacles the density of the swarm must be much greater than that of a swarm connected by audio. As a result, developing an RF- or light-connected swarm will be undesirable for the targeted application. In conclusion, we consider audio to be the best fit for such an application despite its low data rate. Thus we develop and present a detailed system design and experimental evaluation of *Chirpy* an audio-based localization and communication module for swarms.

A Chirpy can localize and communicate simultaneously on the same signal. It uses orthogonal chirps to enhance the communication and localization performance in adversarial environments, such as environments with background noise and/or echo, and to lower the network load as a swarm network can be dense.

We equipped Chirpies with two sound source localization approaches (i) a lightweight classical approach that targets environments with line-of-sight connectivity and (ii) a deep learning-based approach for environments with novel settings such as localizing the source in a non-line-of-sight scenario. For the DNN-based method, we formulate the localization problem as a classification problem. Consequently, Chirpy's DNN-based localizer (AudioLocNet) estimates the DOA and the distance of a sound source on a predefined grid.

Our results show that using audio and AI has great potential for swarm robotic applications, as they provide reliable local communication and accurate localization. Such characteristics enables a swarm of robots to more intelligently explore unknown areas and better coordinate their moves. In our experiments four chipies talking simultaneously achieve a bit error rate (BER) of only 1.4% while being 250 cm apart. And, the AudioLocNet can classify the location of the talker to the correct tile with an accuracy of ≈ 99%.

(1) **Chirpy.** An audio communication and localization module that can easily be attached to a robot or deployed as a static network (Figure 2). To best of our knowledge, this is the first work to combine audio, swarm robots, and AI to construct a guidance path (or field) towards a location of interest.
(2) **Audio-based communication protocol.** Utilizing orthogonal acoustic chirps, we developed a communication system that operates in dense deployment scenarios such as robotic swarms. Chirpy decodes messages from four concurrent transmissions with a BER of $BER \approx 2\%$ at 250 cm. Additionally, our system and message design enable users to increase the data rate while maintaining the same localization accuracy.
(3) **Audio-based localization.** We developed AudioLocNet, a DNN for localizing Chirpies. AudioLocNet shows high localization accuracy (on our testbed, AudioLocNet's accuracy is about 99%) in a variety of novel environments such as non-line-of-sight, and reverberant environment.
(4) **Real hardware.** We extensively evaluated Chirpy's localization and communication under realistic scenarios.

2 RELATED WORK

Since this work involves multiple domains of research, we present only the most relevant literature.

2.1 Localization

Simultaneous Localization and Mapping (SLAM):-. SLAM algorithms (such as GMapping [16], HectorSLAM [29], and KartoSLAM [30]) were originally developed for single robots. Adapting a SLAM algorithm to a multi-robot system presents challenges such as dealing with a dynamic number of robots, scaling the size of an environment, and operating in dynamic scenarios [42]. Developing a SLAM for decentralized multi-robot systems or swarms magnifies the said challenges and introduces new ones such as how should the robots share the gathered information? [26]. Given the limitations of SLAM for swarms, our investigation for relative robot localization envisions a different approach that takes advantage of audio characteristics and recent advances in Artificial Intelligence (AI).

Audio Localization:-. A large body of literature can be found on acoustic localization systems [4, 33, 35, 41]. Traditionally, one of the most used features for localization is the Time Difference of Arrival (TDOA) [21, 49]. Two components of a source position can be estimated – the bearing (azimuth) and the distance. Systems such as [38] are able to estimate both but require reference points in the environment. Majid et al. [34] mounted a set of microphones and a speaker on a robot to enable relative localization between the robots without the need of reference points. However, the presented method does not support multiple access. Khyam et al. [27] present a method to design chirp-like waveforms that can be used to implement multi-robot localization. More advanced techniques such as beamforming could also be used for localization [7, 12, 50]. However, multi-source localization becomes difficult when the number of transmitting sources exceeds the number of microphones [2, 41]. This affects the scalability and flexibility of systems such as a swarm of robots. Nemec et al. [37] propose multiplexing sound signals with frequency tones to identify swarm robots in close proximity. However, such a method can only be used for identification, and data has to be transmitted over a different medium. Wang et al. [47] present a method to localize a sound source using a microphone array. However, since it makes use of nearby wall reflections, this method is very much environment-dependent.

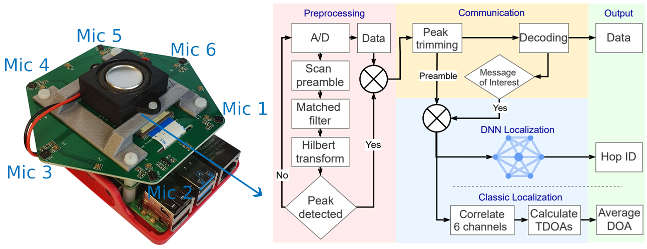
Similar to prior work our lightweight localization method depends on TDOA to estimate the Direction of Arrival (DOA) of a sound signal. Unlike prior work, however, in our study, we investigate the possibility of using orthogonal audio chirps to simultaneously localize multiple sound sources.

Deep Learning-based Audio Localization:-. Deep learning excels when the relationship between the input and output is non-linear. This makes it an effective tool for sound source localization. He et al. [20] used a Deep Neural Network (DNN) to enable a humanoid robot to localize up to two simultaneous speakers. Adavanne et al. [1] showed that their proposed DNN is capable of determining the bearings of up to three overlapping sounds from different sources. Chakrabarty and Habets [9] presented a DNN that consists of only convolutional and fully connected layers to determine the DOA of up to 3 speakers. They used a classification network to locate each of the many sources in one of 37 DOA classes spanning half a circle. Vera-Diaz et al. [46] present an end-to-end localization method that uses the raw recorded signals as inputs to the network. The authors first trained the DNN in simulation and then fine-tuned it with small real-world data set. The authors of [10, 11] present a simulated robot capable of detecting and navigating towards a sound event using a DNN that processes both visual and audio signals. The use of the W-disjoint orthogonality principle to localize two speakers using convolutional networks is conceived by Hammer et al. [18]. This allowed them to achieve a high time resolution for the localisation. Xu et al. [48] show how a DNN can use the outputs from many microphones (64 positioned on a single plane) to locate up to 25 sources in a 1 × 1 m plane that faces the array. Our work targets swarm robotics applications. Consequently, we chose the hardware with a small form factor and our focus is on environments where non-line-of-sight operation is needed; moreover, the environments could be highly reverberating.

2.2 Audio based Communication

Next, we briefly discuss the related work of Aerial Acoustic Communication (AAC). Sounds good [25] presents a primitive audio communication platform for robots, where the robots announce physical markers in the area by using the Dual Tone Multiple Frequency (DTMF), or touch-tones. Suddrey et al. [45] present a novel DTMF-based backup communication protocol for swarm robots. The system encodes messages into words, which are mapped to one of the 16 tones supported by DTMF. Nakayama et al. [36] created CyberPerformerAudio that can remotely control an animatronic doll at a distance of 2 m by encoding DTMF codes into existing audio. Drew et al. [13] present acoustic communication for inflatable robots with Frequency Shift Keying (FSK) and ALOHA, achieving a bit rate of 200 bps on physical contact. Anguelov et al. [3] discuss the difficulties regarding communication in multi-robot systems and propose a model for communications based on a hybrid implementation between light and ultrasonic AAC. Different from the aforementioned work, we use orthogonal chirps to support concurrent transmissions and simultaneous localization between swarm robots.

Our design using the orthogonal chirps is based on the methodology presented in [27]. Prior work on AAC with orthogonal chirps has mainly focused on increasing the bit rate [8, 23, 32] of the communication protocol. Our work, however, is centred around allowing concurrent transmissions and relative localization, and we present the entire communication stack. Furthermore, orthogonal audio chirps design prior to [27] do not scale well with increasing the dimensionality of orthogonal chirps. This is because these methods either split the bandwidth [8], apply quaternary on-off keying [23] or shift the symbols circularly in time [32]. The method conceived by Khyam et al. [27] simply increases the time of the orthogonal chirp linearly to support a higher number of symbols. To the best of our knowledge, we are the first to use audio for localization in a robot swarm to construct a guidance path between two points.



(a) Hardware **(b) Software architecture**

Figure 2: Chirpy: An audio-based communication and localization module for swarm robots.

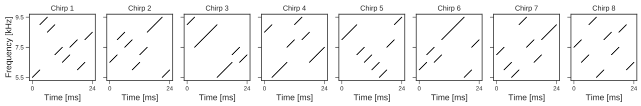


Figure 3: Eight orthogonal chirps with eight sub-chirps, $M = 8$ and $T_s = 24$ ms and and frequency range 5.5-9.5 kHz [27].

3 CHIRPY: SYSTEM OVERVIEW

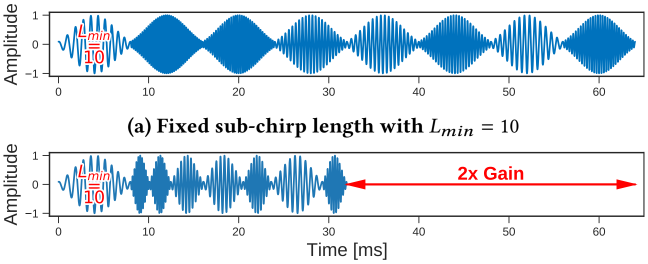
Figure 2 shows the hardware and software architecture of the proposed communication and localization system. To provide a complete solution for the problem of path finding using a swarm of robots, our software comprises a communication and localization stack.

3.1 Acoustic Communication

As audio has a limited local communication range, an acoustic communication protocol can scale well with the number of devices (e.g., a swarm) because of spatial reuse. Further, with orthogonal chirps, Chirpies achieve higher networking throughput.

3.1.1 Orthogonal chirps design. Narrowband audio signals suffer significantly from constructive and destructive interference [34]. This renders them useless for robot localization. Alternatively, white gaussian audio noise can be used for distance and direction estimation between swarm robots [34]. However, distinguishing it from ambient noise is hard as white noise does not possess correlation properties. Chirp audio signals, on the other hand, are wideband signals with strong correlation properties. Therefore, they suffer less significantly from multipath effects than narrowband signals and are distinguishable from ambient noise. Our system uses linear[1] orthogonal audio chips for localization and communication to mitigate the effect of concurrent transmissions in swarms. We

[1] linear chirps are chirps whose frequency changes linearly over time.



(a) Fixed sub-chirp length with $L_{min} = 10$

(b) Dynamic sub-chirp length with $L_{min} = 10$ guaranteed by (3)

Figure 4: Orthogonal chirps from 1 kHz to 5 kHz with $M = 8$. We take half the symbol time without violating the minimum required cycles L_{min} per sub-chirp.

based our initial orthogonal chirp design on [27] (Figure 3). The number of sub-chirps M and chirp duration T_s are input parameters for our system that can be set according to the users requirements (in our case, $M = 8$ and $T_s = 24$ ms). However, we observed that when the sub-chirp length is too short (i.e., containing just several or no oscillations), the orthogonality and correlation properties of the signal decrease, which negatively impacts the performance of the system. This is due to hardware limitations introduced by the speakers. To prevent transmitting chirps that are too short and to guarantee a certain number of cycles (oscillations) in the chirp, the following formula can be used [39],

$$T_b = \frac{2L}{f_0^s + f_e^s}, \tag{1}$$

where T_b is the time required to construct a chirp with L cycles, and f_0^s and f_e^s are the start and end frequency of the chirp. To ensure that the sub-chirps of an orthogonal chirp contains numbers of cycles greater than a certain minimum, i.e., $L \geq L_{min}$, the following expression is utilized,

$$T_s = M \cdot \frac{2L_{min}}{2f_0 + \frac{f_e - f_0}{M}}, \tag{2}$$

where T_s is the time needed to meet the constraint on L and f_0 and f_e are the start and end frequency of the orthogonal chirp. However, while this would guarantee L cycles for the lowest frequency sub-chirp, all other sub-chirps would have more cycles because the length of the sub-chirps is fixed (Figure 4a). To reduce the symbol time without violating $L \geq L_{min}$, we use the following,

$$T_s = \sum_{i=1}^{M} \frac{2L}{2f_0 + (2i-1)\frac{f_e - f_0}{M}}. \tag{3}$$

This reduces the symbol time by shortening every individual sub-chirp to fit exactly L cycles. However, this does result in an orthogonal chirp with sub-chirps of different lengths (Figure 4b). Additionally, regardless of whether Equation (2) or (3) is used, every sub-chirp is shaped by a Kaiser window with $\beta = 4$ to reduce the out-of-band leakage [40] and the clicking effect caused by any phase differences between the sub-chirps.

3.1.2 Frame composition. An acoustic frame consists of the following fields: `sender_id`, `message_id`, `data`, and `CRC`. For the preamble, we use 24 ms orthogonal chirps. To distinguish them from the rest of the frame, the preamble chirps use a different frequency band than the data chirps. For the proposed application a fixed data field contains the direction of arrival and a hop count is considered. To check for errors an 8-bit CRC is appended. However, the aforementioned values can easily be changed to meet application requirements.

3.1.3 Medium access control. A simple Carrier-Sense Multiple Access (CSMA) protocol is adopted. Before transmitting, a Chirpy listens for $T_{fp} + \epsilon$, where T_{fp} is the duration of a frame plus an additional preamble, and ϵ is a small random delay drawn from a uniform distribution to reduce the probability of repeated collisions. However, it should be highlighted that a Chirpy scans the medium only for orthogonal chirps that are assigned to itself, and otherwise it is permitted to transmit.If two or more robots close to each other and transmit concurrently using the same chirp symbols the medium access control (MAC) layer resolves their conflict and enables them to communicate. At this stage, we consider the error-correcting code and routing protocol to be out of scope. However, in Section 6 we elaborate more on these topics and their relation to the proposed application.

3.2 Localization

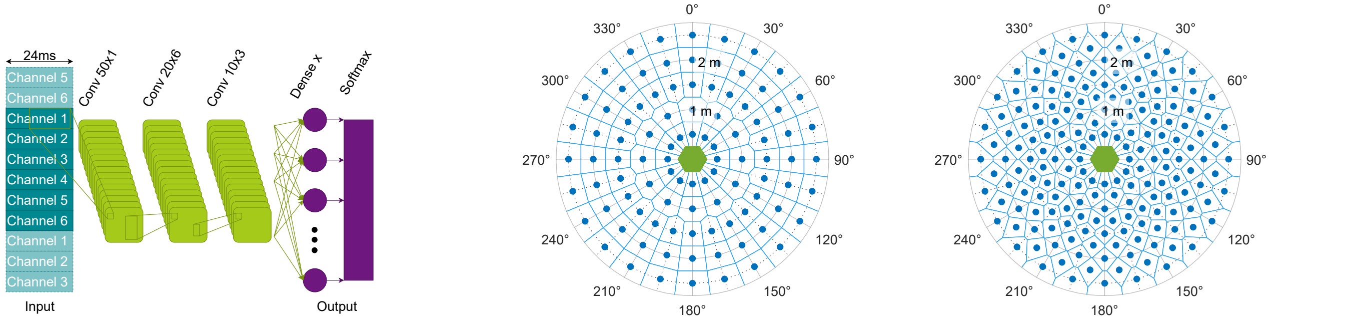
For the relative localization amongst Chirpies, two approaches are considered – a classical and DL-based one.

3.2.1 Classical approach. To estimate the Direction of Arrival (DOA) of a received audio signal, several processing steps are required, which are shown in Figure 2b. The signal is received through six microphones (channels). The preprocessing steps are used to determine when a preamble is present and record the frame. A single-chirp recording on all six channels is passed to the classical localization module. The TDOA values between each pair of microphones are then found by correlating the recording of a single chirp with a stored copy using a matched filter on all of the six channels and identifying the time differences between the peaks. The DOA is estimated with the derived TDOA values using the algorithm explained in Section 4.3.1. Additionally, the DOA estimates found for each adjacent pair of microphones are averaged to reduce the error in the estimate.

3.2.2 AudioLocNet: DNN-based localization. The second localization method uses a Deep Neural Network (DNN), referred to as AudioLocNet, to locate a sound source. It comprises an input layer, 3 hidden layers, and an output layer (Figure 5a). AudioLocNet is a classification network that maps an audio signal to a class representing the source location. Two separate localization grids are considered: a coarse and a fine one (Figure 5b) & (5c). For each grid a separate version of AudioLocNet was trained. Apart form the the output layer, both versions have the same architecture.

The input of AudioLocNet consists of a 6 × 1060 array which at a sampling frequency of 44.1 kHz portrays a 24 ms recording captured by each microphone. This input comes from the preamble of a message. The message to localize on is identified by the communication stack and its preamble is passed to the localization module (Figure 2b).

The output layer has 96 nodes when trained for the coarse grid (Figure 5b) and 180 nodes when trained for the fine grid

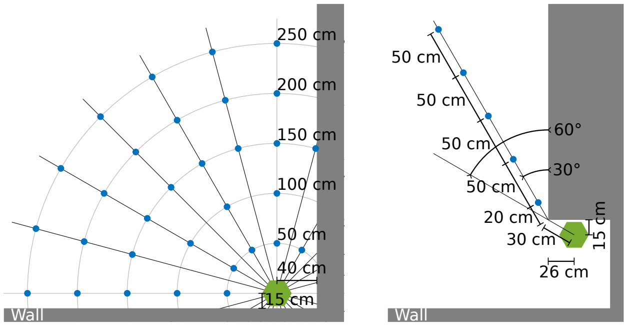


(a) AudioLocNet has three CNNs, a dense layer and a softmax layer. AudioLocNet is trained to localize a sound source on a predefined grid.

(b) Coarse location grid, 96 locations

(c) Fine location grid, 180 locations

Figure 5: AudioLocNet (5a) is a DNN that is trained to localize a sound source on a coarse (5b) or fine (5c) grid. The blue dots represent possible sound transmitter locations and the green hexagon represents the location of the receiver.



(a) Reverberating environment

(b) Non-line-of-sight

Figure 6: Setup for the recording in reverberating and non-line-of-sight scenarios for the coarse grid (Figure 5b), where the blue dots represent possible sound transmitter locations.

Ring	1	2*	3	4*	5	6*	7	8*	9
Radius [cm]	50	75	100	125	150	175	200	225	250
Angle [°]	30	30	30	15	15	15	15	15	15
Offset [°]	0	15	0	7.5	0	7.5	0	7.5	0

Table 1: Parameters for the rings which form the location grids of Figure 5b and 5c. Angle denotes the angular distance between two subsequent locations; and Offset denotes the angular offset from the 0° direction. Columns denoted with '*' are only used for constructing the fine grid.

(Figure 5c). The fine grid is constructed out of 9 rings centered around the microphone array. These rings have radii ranging from 50 to 250 cm, in steps of 25 cm. Where the three smallest rings contain 12, uniformly spaced, locations each and the remaining rings contain 24 locations per ring. All odd numbered rings (i.e. those at 50, 100, 150, 200, 250, cm) have their first location aligned with the 0° direction, whereas the ring at 75 cm has it's first location at an offset of 15° and the remaining rings have their first locations at an offset of 7.5°. The locations of the coarse grid form a subset of those of the fine grid. Specifically it contains the locations on the rings at 50, 100, 150, 200 and 250 cm from the microphone array. These rings are summarised in Table 1, where the rings marked with an asterisk (*) are only used in the fine grid.

In the input layer, the 6×1060 input array is cylindrically padded in order to account for the physical locations of the microphones. Without this padding, microphones 1 and 6 would be on opposite ends of the data array even though they are located next to each other on the microphone array. We observed difficulties in the training due to the first convolutional layer struggling with finding features between the mics at the ends of the array without this padding. The padded array is 11 channels by 1060 time samples in size. The first hidden layer is a convolutional layer with a 50×1 kernel. By going over the individual channels, this layer helps with finding the chirps. The second layer is a 20×6 kernel which shifts over all six microphone channels at a time, where the 20 time steps per kernel ensure that a signal arriving at a first microphone during the first time step will be received by a microphone farthest away from said first microphone before the end of the kernel. The final hidden layer layer comprises another convolutional layer with a 10×3 kernel. All convolutional layers consist of 64 filters and use rectified linear unit (ReLU) activation functions. It should be mentioned that we intentionally dropped the use of a max-pooling layer after the convolutional layer, which is a common practice. This is to maintain the time differences between the signals received via different microphones. Lastly, the output layer is a fully connected layer with 96 or 108 nodes, depending on which localization grid was used. A softmax activation function then normalises the values of the output nodes into a probability distribution. The output node with the highest value is then taken as the predicted source location.

3.2.3 Localization dataset. In order to train the network for different scenarios, data was collected from known sound source locations around the microphone array. Three different indoor environments were selected:

(1) **Less reverberating, line-of-sight.** In this environment the microphone array was placed away from any walls and with a direct line of sight to the source.

Table 2: A database summary of orthogonal audio chirps.

Parameter	Value
Total sample size	1324800
Environments	LOS, reverberant and NLOS
Source locations	180 (Figure 5)
Orthogonal chirp types	8
Symbol duration	24 and 48 ms *
Samples per location	200

* The 48 ms samples are only available for locations on the coarse grid of Figure 5b

Table 3: Training parameters

Parameter	Value
Learn rate	0.005
Learn rate schedule	constant
β_1	0.9
β_2	0.999
ϵ	10^{-8}
L2 regularization factor	0.0005
Mini batch size	256

(2) **Highly reverberating, line-of-sight.** In this environment the microphone array was placed in a corner in between two walls, this creates an environment where reverberations are more prevalent. There is, as with the previous environment, a line of sight between the array and the source. Figure 6a.

(3) **Non-line-of-sight.** In this environment the microphone array and source are placed in such a way that the corner of a wall breaks the line of sight between them. Figure 6b.

Throughout this paper these environments are refferenced with the labels LOS, reverberant and NLOS respectively. If the walls prevented access to all of the source locations, then the microphone array is rotated and the newly available source locations are used for recordings. This process is repeated until recordings from all locations were gathered. Per source location 200 individual orthogonal chirps were recorded for each of the 8 different chirp-waveforms. For the coarse grid locations, chirps with a T_s of 24 and 48 ms were recorded, for the additional rings of the fine grid, only chirps with a T_s of 24 ms were recorded. In this way, 1.3 million samples were collected (Table 2).

4 CHIRPY: IMPLEMENTATION

4.1 Hardware

Chirpy comprises a Raspberry Pi 4 with a microphone array and a speaker. The array consists of six microphones equally spaced on a circle with a diameter of 10 cm [44]. The speaker placed on top of the array is 6 Ω 2 W [43] (Figure 2a).

Table 4: Communication Parameters.

Parameter	Value
Preamble Duration	24 ms
Preamble Freq. Band	5.5-9.5 kHz
Symbol Duration	24 ms/18.8 ms*
Symbol L	17.5
Data Freq. Band	9.5-13.5 kHz
Frame length	40 bits
Frame Duration	984 ms/776 ms*

*We test both with dynamic and fixed-length sub-chirps (Eq. (2) and (3)), for this reason, we have two different symbol times.

4.2 Acoustic Communication

4.2.1 Frame composition and symbol distribution. One of the key benefits of AAC in our application is that it allows for synergy between localization and communication. However, typically, optimizing the bit rate for communication often conflicts with the need for high SNR in classical localization for accurate positioning. To address this challenge, we propose a novel solution of constructing the preamble in such a way that localization can still be achieved, while also allowing for greater flexibility in the design of the data chirps. This approach has the potential to increase the bit rate while maintaining high localization accuracy.

Symbol distribution. As mentioned previously, creating orthogonal chirps with $M = 8$ results in a set of eight chirps. These chirps are divided into four sets of two symbols, (s_0^i, s_1^i), representing logical digits 0 and 1 for robot i. Each Chirpy module in a swarm is assigned a set such that a uniform distribution is approached. Consequently, the AAC allows four simultaneous transmissions without (or with minimal) interference.

Preamble design. The preamble is a 24 ms orthogonal chirp with the same sub-chirp distribution as symbol s_0^i. To simplify frame detection and decoding, the preamble is transmitted on a different frequency band than the rest of the frame (Figure 7a). Equation (2) is used for creating the preamble to ensure fixed-length sub-chirps. Beyond frame detection, the preamble is used for localization.

Frame layout. Each frame starts with a preamble. The body of the frame contains the following fields: `sender_id`, `message_id`, `data` and `CRC`. Each field is one byte long, except for `data` which is of two bytes — encoding distance and angle. Table 4 gives an overview of the communication parameters used.

4.2.2 Transmission. The signal is encoded as follows. Starting with the first data bit, we map every bit to a symbol (s_0^i, s_1^i). Then, every symbol is designed with either Equation (2) or (3), depending on whether dynamic sub-chirps are desired. These symbols are then concatenated, appended to the preamble, and played over the speaker.

4.2.3 Reception. Algorithm 1 outlines the reception and decoding process of our AAC. First, the receiver scans the acoustic medium for a potential preamble signal. Once a preamble symbol is detected, the receiver records the maximum possible frame.

Decoding. As there is a one-to-one mapping between the preamble symbols and data symbols, the receiver directly convolutes the recorded signal with the appropriate pair of the orthogonal chirps

Algorithm 1 Decode Acoustic Messages

```
 1: $x(t) \leftarrow$ Record( )
 2: if ContainsPreamble($x$) then
 3:     $\hat{c}^i_{0,1}(t) \leftarrow \textsc{conv}(x(t), s^i_{0,1})$
 4:     $c^i_{0,1}(t) \leftarrow \textsc{abs}(H(\hat{c}^i_{0,1}))$
 5:     $c(t) \leftarrow \textsc{max}(c^i_0(t), c^i_1(t))$
 6:     $p_{max} \leftarrow \textsc{max}(c(t))$
 7:     $peaks \leftarrow \textsc{FindPeaks}(p_{max}, p_{step}, c(t))$
 8:     $peaks \leftarrow peaks \cup \textsc{FindPeaks}(p_{max}, -p_{step}, c(t))$
 9:     $peaks \leftarrow \textsc{sort}(peaks)$
10:     $peaks \leftarrow \textsc{TrimPeaks}(peaks)$        ▷ remove peaks on edges
11:     $bits \leftarrow \textsc{GetBits}(peaks, c^i_{0,1})$
12: end if

13: function FindPeaks($p_{max}, p_{step}, data$)
14:     $peaks \leftarrow \emptyset$
15:     $p \leftarrow p_{max}$
16:     while $0 < p < len(c(t))$ do
17:         $p \leftarrow \textsc{FindLocalMax}(p, data)$        ▷ return max value around the given peak
18:         $peaks \leftarrow peaks \cup p$
19:         $p \leftarrow p + \Delta$
20:     end while
21:     return $peaks$
22: end function
```

(Line 3). Moreover, to reduce the time complexity of the convolution process (which is $O(N^2)$) we implemented the *overlap-add* method [6]. This method is optimized for convolution between a large time series and a set of smaller ones. Then we use the **Hilbert transform** to get the envelope of the correlated signals (Line 4). Taking advantage of the orthogonality, we zip the two outputs of the previous step and select the maximum points. Then, by selecting the point with the highest value, a peak is detected (Line 5-6). Starting from this peak we search bidirectionally for other peaks to extract the digital data. Given that these peaks are spaced $T_s \cdot f_{sample}$ samples apart, the algorithm searches only around these potential points. More specifically, we specify a search area of 10 % around a potential peak point to account for multipath, rounding, and timing inaccuracies. The highest point in this range is assumed to be the actual peak (Line 7). In a low-SNR scenario, the start and end of transmission might be difficult to discern accurately. Therefore, we simply create a window with size b, which is the number of bits in a single transmission. Then, we slide this window over the found peaks and select the offset which gives us the highest average peak value and select these peaks as the transmission (Figure 7b).

Then we convert the found peaks to data. For every peak $p_i \in c(t)$ we select the symbol with the highest amplitude in $c_{0,1}(t)$, since the other is cross-correlation. Figure 7a shows the output of the matched filter and peak detection for 8 bits of data. Moreover, Figure 7b zooms in on the first three symbols of Figure 7a and appends some zeros to the sides to better visualize the peak detection algorithm.

4.3 Localization

4.3.1 Implementation of the classical approach. Our method for estimating the Direction of Arrival (DOA) of a received acoustic signal is inspired by [24]. First, we calculate the Time Difference of Arrival (TDOA) between each pair of the received six audio signals (from adjacent microphones). This is done by cross-correlating them with a stored copy of the transmitted chirp (matched filter), detecting the times τ_{m_i} of the peaks, and computing the difference between them. Then we apply the following expression to compute an average estimate of the Direction of Arrival (DOA) of the received signal,

$$\text{DOA} = \frac{1}{M}\left(\sum_{i=0}^{M} \arcsin\left(\frac{\tau_{m_i m_{i-1}} \cdot c}{l_{m_i m_{i-1}}}\right) - (i-1)\alpha,\right) \tag{4}$$

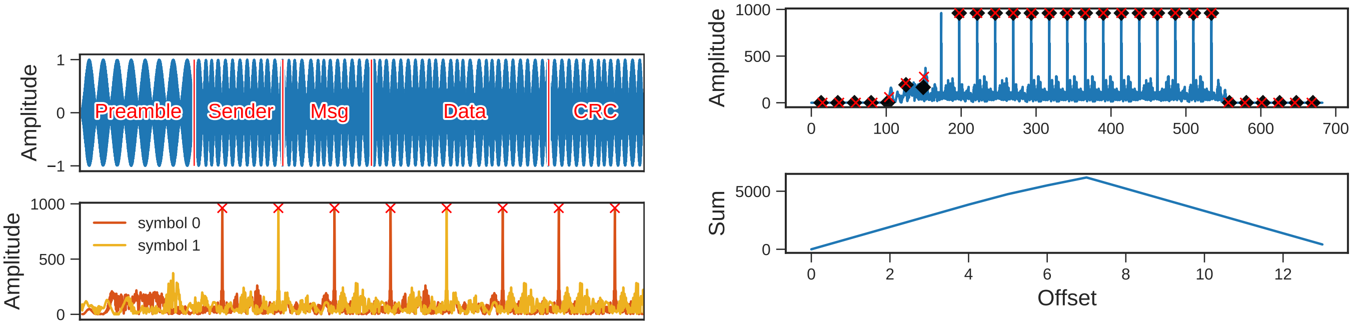
where $\tau_{m_i m_{i-1}}$ and $l_{m_i m_{i-1}}$ are the TDOA and the distance between microphone m_i and m_{i-1}, M is the number of microphones, α is the angle between the microphones on the array relative to the center, which is in this case 60°, and c is the speed of sound. Further, θ is observed relative to the line perpendicular to $\overline{M_1 M_2}$, as shown in Figure 2a.

4.3.2 Implementation of AudioLocNet. The deep learning (DL) method of sound source localization works as follows: the microphone array records a 24 ms frame for all six microphones. This recording contains an orthogonal chirp as discussed in Section 3.1 and the recording is done with a sampling frequency of 44.1 kHz. This recording is then loaded into the deep neural network (DNN) to estimate the source location. The network does this by finding the most likely source locations among the location grid from Figure 5.

Training. AudioLocNet was trained using a data set sampled from the full data set summarised in Table 2. This sampled data set comprises 150000 samples with orthogonal chirps with a duration of 24 ms, sampled randomly from the different locations. The sampled set is split into a training, validation, and testing set, each containing 75%, 15%, and 15% of the samples from the sampled data set, respectively.

The network was trained using the Adam training algorithm [28], with cross-entropy loss as the loss function. The used parameters can be found in the Table 3, which generally match the suggestions of the original paper [28]. Two mechanisms were used to prevent overfitting, L2 regularisation and dropout. During the training, a dropout layer with a dropout probability of 0.2 was added after the input layer to increase the network localization robustness. L2 regularization comprises adding a term to the loss function to penalize high network weights. This incentivizes the trainer to make a simple network over a complex one, thereby reducing overfitting [5].

After each epoch, the validation set is run through the current network. The results from this set are used to monitor the training progress and fitting characteristics and not to update the weights of the network. As long as the validation loss follows the loss in the training samples we can conclude that the network is not overfitting. Once the validation loss stops improving the training is stopped and the network with the lowest validation loss is selected. Figure 8 depicts the training process both in terms of the classification accuracy in the top graph and the loss (which is used by the



(a) The output of the matched filter and peak detection. The red crosses are the found peaks. We only show the sender ID (8 bits) in the bottom plot.

(b) The output of the peak detection algorithm. The peaks are the same as from Figure 7a. The diamonds are the initial predicted peaks, the red crosses are the actual peaks. The peak without a diamond or cross is the initially found peak. The triangle is the windowed-sum, which gives the correct offset.

Figure 7: Visualization of the decoding method.

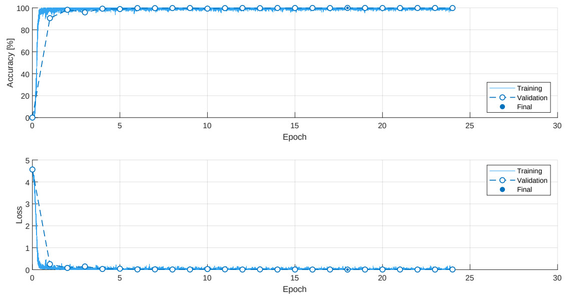


Figure 8: Plots showing the training process in terms of both the accuracy and loss.

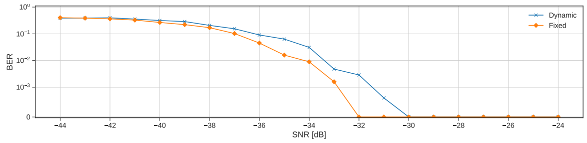


Figure 9: Chirpy's communication performance under different SNR values. It should be highlighted that while fixed-length chirps achieve higher BER, the dynamic ones haver a higher bit rate (BR) as they are shorter.

trainer to improve the network) in the bottom graph. The accuracy is the mean of the number of correctly identified samples over the total number of samples in a mini-batch. The plot also shows how the validation accuracy and loss follow the training accuracy and loss respectively, implying that the DNN is not overfitting.

5 EVALUATION

In this section, we examine the performance of communication and localization methods in Chirpy under increasingly challenging scenarios.

5.1 Experimental setup

All the communication and localization experiments were done in an indoor environment. As we imagine that Chirpy will be mounted on robots, we designed the experiments while taking common swarm robotic platforms. For example, robots in a warehouse often meet while crossing an intersection-like configuration. This scenario is captured by the non-line-of-sight (NLOS) setting. For the line-of-sight (LOS) experiments there were no obstacles in the proximity of the receivers and transmitters. Figure 11 shows one of our experiment scenarios where Chirpies are mounted on robots. Moreover, for the NLOS experiments, the transmitters and receivers were positioned around a corner as shown in Figure 6b. Further, to examine the effects of echos on the performance of Chirpy, a Chirpy receiver was positioned in a corner between two walls as shown in Figure 6a. Finally, for all the experiments the transmitters and receivers were positioned at the same height on a flat surface.

5.2 Communication

Table 4 gives an overview of the communication parameters. All tests refer to these parameters, unless stated otherwise.

5.2.1 Signal to Noise Ratio.

We used a common source of noise, namely, Babble noise from [22], to analyze the performance of communication capability of Chirpy under different levels of signal-to-noise ratio (SNR). Most importantly, the babble noise provides a very realistic scenario where these robots with Chirpy should be working. Babble noise creates a challenging scenario for Chirpy, since they overlap in the frequency spectrum. For each SNR setting, a noise signal was sampled out of the recording at random and mixed with the audio signal. The superimposed signal was then fed to the receiver of Chirpy and the bit error rate (BER) is calculated. Each presented data point is the average of 30 iterations, and we varied the SNR from -44 dB to -24 dB with an interval of 1 dB (Figure 9). We notice that the system reaches zero BER at -32 dB for the fixed-length chirps and -30 dB for the dynamic ones. It should be highlighted that while the fixed-length chirps have better BER, the dynamic ones have a higher bit rate (BR) as they are shorter in

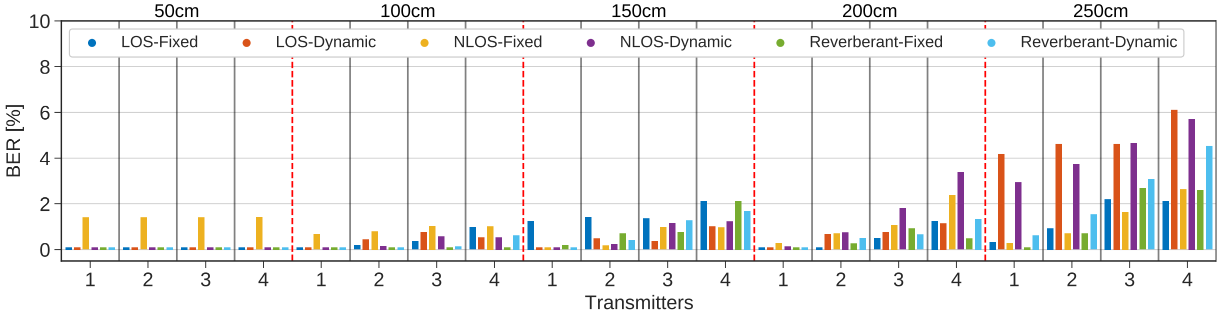


Figure 10: The mean BER for line-of-sight (LOS), non-line-of-sight (NLOS) and reverberant scenarios from one to four concurrent transmissions. Two types of sub-chirps are tested: Fixed (Equation (2)) and Dynamic (Equation (3)).

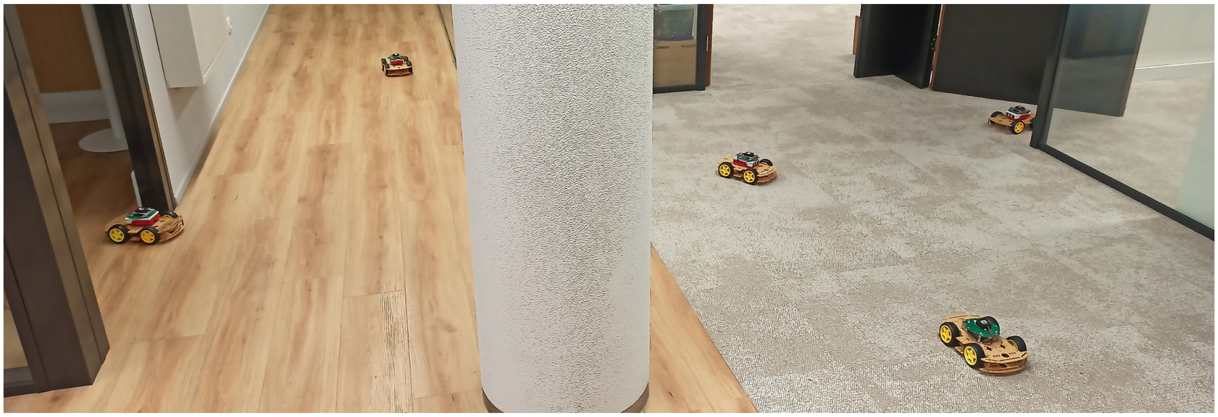


Figure 11: Robots mounted with Chirpy

length. Overall, Chirpy can communicate reliably under severely low SNR values.

5.2.2 Concurrent transmissions. To test the ability of a receiver to decode signals of interest in a concurrent transmissions scenario, we superimposed signals from other transmitters (i.e., transmitters with different pairs of orthogonal chirps) on the signals of the original transmitter. Then the superimposed signals are transmitted and the BER is observed at the receiver. Figure 10 shows the mean BER over various distances (50 cm - 250 cm), scenarios (LOS, NLOS and reverberating) and concurrent transmissions (1-4). The data is generated with 400 repetitions per sample.

Figure 10 clearly shows that communication with all transmitters up to 150 cm is fully functional. Within the aforementioned range the *BER* did not exceed 2.1%. There is an interesting small spike of $\approx 1.4\%$ in the BER of the NLOS Fixed configuration at 50 cm. This is also the case for the LOS scenario at 150 cm. We hypothesize that these are due to multipath caused by the testing environment or coincidental outside interference. Furthermore, at 200 cm, the performance of the NLOS scenario degrades to $BER \leq 3\%$, while the rest maintains a $BER \approx 1\%$. From 200 cm onward, there is a clear distinction between the performance of the fixed and dynamic sub-chirps. The dynamic sub-chirps degrade to $BER \leq 6.1\%$ while the fixed sub-chirps maintain $BER \leq 3\%$. The combination of a shorter symbol time and a low SNR is the main reason behind the lower performance of the dynamic sub-chirps. Since this combination leads to small correlation peaks that are hard to distinguish from noise during the decoding process. However, our evaluation shows that the communication system of Chirpy supports concurrent transmissions with minimum interference between the signals.

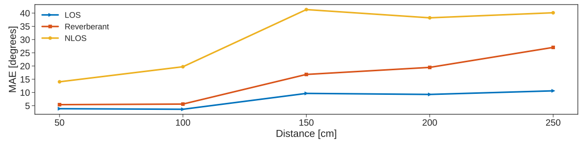
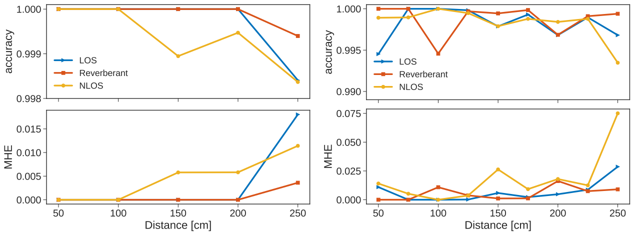


Figure 12: DOA estimation error of Chirpy's classical localization method.

5.3 Localization: Classical Approach

The signals used for the classical evaluation are 24 ms orthogonal chirps transmitted by a single transmitter. Figure 12 shows the mean absolute error (MAE) of the estimated DOA for classical localization in three different scenarios. The MAE is computed such that it is in the range $[-180, 180]$; this is to ensure that the MAE of a signal with a DOA of 0° and an estimated DOA of 350° is normalized to 10°.



(a) Coarse network. **(b) Fine network.**

Figure 13: Mean hop error and accuracy of for two implementations of AudioLocNet—Chirpy's DNN-based localization method.

In the LOS environment, the MAE only slightly increases over distance with a maximum value of ≈ 10°. Meanwhile, in the reverberant environment (shown in Figure 6a), the error increases more strongly over distance and reaches a maximum value of ≈ 27° at the farthest testing point. The NLOS environment (Figure 6b) yields the worst accuracy. At distances beyond 1.5 m the MAE fluctuates around 40°. From these results, we can conclude that despite being a computationally light approach (compared to the DNN-based approach), this localization method cannot generalize across different environments and should be used only when the transmitter and receiver have a direct line-of-sight connection and the level of reverberation is low.

5.4 Localization: Deep Learning Approach

For the testing of the performance of AudioLocNet, the testing data set was used since none of the sound samples from this set have been seen by the network before.

Hop error. Usually, the performance of a classification network like AudioLocNet is measured using metrics like the accuracy and the F1-score [15]. However, these metrics only look at the results from a pure classification perspective and do not take into account that the classes correspond to physical locations. Meaning that they would penalize being one class next to the correct class the same as when the prediction is on the opposite end of the location grid. Therefore, we first introduce the hop error to more closely reflect the relations between (miss) classifications and the true classes. Finally, we discuss the classification performance of the network.

The hop error is defined as the number of classes from the true class to the predicted class. It is determined by drawing a straight line between the true and predicted classes and counting the number of classes the line passes through. For a correct prediction, the hop error is set to 0. This metric is chosen over the distance error between the corresponding physical locations of true and predicted classes because the distances between adjacent class are not constant due to the circular nature of the location grid.

Network performance. The performance for the two versions (coarse and fine) of AudioLocNet can be observed in Figure 13. The results show that AudioLocNet is able to find a Chripy of interest around itself, as depicted by the low mean hop errors. For the coarse grid (Figure 13a) the performance seems to decrease as the distance increases, as indicated by the decreased accuracy and increased MHE. The version trained on the fine grid (Figure 13b) trades some of the performance for the finer localization grid. This version also seems to be more resilient to changes in the distance. AudioLocNet shows less differences in its performance for the different environments when compaired with the classical method. For the coarse network the accuracies for the LOS, reverberant and NLOS environments are 99.96%, 99.99% and 99.92%, respectively, and for the fine network they are 99.88%, 99.93% and 99.86%, respectively. Over all environments, the coarse network reaches an accuracy of 99.96% and the fine network has an accuracy of 99.89%.

6 DISCUSSIONS

Performance under concurrent transmissions: We notice that in low SNR scenarios concurrent transmissions lead to a higher BER. This degradation in performance becomes severe when chirps with shorter duration are considered. We hypothesize that the orthogonal chirps from [27] gradually lose orthogonality due to two factors: i) Signal degradation due to frequency selectivity of the hardware & environment and ii) Low number of cycles L compared to the reference [27]. Namely, Chirpy has 17.5 cycles in a 24 ms chirp, while [27] has 77 in a 12 ms chirp (ultrasound band). However, while our results show that performance degrades with concurrent transmissions, we strongly believe that the performance of non-orthogonal chirps would be significantly worse due to constructive and destructive interference.

Furthermore, a rake receiver could be employed to improve performance of the AAC [31, 32], especially since Chirpy already has multiple microphones. Further, the J-shape-dection from [23] and normalization method and rate adaption scheme from [8] could further improve the performance of the system, which is not covered in this work.

Usage of orthogonal chirps: Our current implementation of the orthogonal chirp allows up to four concurrent transmissions. This is under the assumption that each Chirpy uses two orthogonal chirps to represent data bit 0 and 1. However, to support more concurrent transmissions, one can assign a single orthogonal chirp per chirpy and use the frequency slope of the sub-chirps (up/down) to encode the bits. Other options include color-coding chirpy's network, such that chirp sets can be re-used, or encoding the data in the length or presence of a symbol.

Performance under varying channel conditions: Swarm robots are often moving and searching. This means that the communication channels between the robots will not stay consistent. While our concurrency tests and simulations do evaluate this to a certain extent, they do not take Doppler shift and multipath effects due to movement into account. However, this is pronounced only if the speed of the robots is comparable with audio frequencies used [8]. In general, we think mobility deserves its own separate study as when it is considered we must not only consider localization but tacking also. Therefore, it is left as future work.

Error-correcting code: Messages from a Chirpy have a relatively long duration (100s of ms); therefore, repeat requests are expensive and can quickly delay the network. For this reason, we envision the use of forward correcting codes to make the communication more robust.

Analysis of our results shows that most errors occur in bursts, either because, (a) a transmission overlaps with another and the cross-correlation lowers the SNR too much or (b) the channel conditions change during the transmission of a frame. For this reason, interleaving should be applied, since this is more robust against burst errors. Moreover, since the communication is already quite slow, the extra latency induced by buffering is not significant compared to a retransmission.

Routing: Audio Swarm Potential Field (ASPF) acts as a guiding mechanism; therefore, the routing scheme propagating the message constructing the ASPF must maintain a notion of directionality. In other words, circularly or randomly routing the message of interest may make constructing the ASPF challenging. Designing routing scheme for ASPF is an open question that still needs an answer.

Hop versus absolute error: We formulated the sound localization problem as a classification and not a regression problem because we wanted to have the potential to localize multiple sound sources at once. Due to this design decision, our accuracy results are reported in hop error rate and not in absolute error, which without our objective is more natural for measuring localization accuracy.

Real-world scenario: As mentioned earlier, we consider swarms of ground wheeled robots in an indoor environment. Therefore, we only considered localization in two dimensions. However, in real-world scenario, wherein the terrain can be of different heights, such as in rescue operations, Chirpy can perform well when the height difference between the robots is comparatively negligible with respect to the distances between them. As the terrain height increases, the FoV of the microphones need to be wider so that the neighboring robots can communicate and localize. If the robots have significantly high height differences, then three dimensional localization is required. Three-dimensional spaces is not in the scope of the our work.

Another point that should be highlighted is the performance of AudioLocNet in a completely different environment. While it is logical to expect the performance to degrade in such settings—as with any DL model—the solution to this challenge is well know: collect more data and retrain the model.

Classical approach: The deep learning approach estimates both the distance and DOA of a signal, while our classical approach only estimates the DOA. Given the hardware, a classical approach was unlikely to handle a ranging problem well [7]. For estimating the DOA, there are numerous approaches that could have been used, some of which might have outperformed our classical approach. However, our approach was chosen since it was lightweight, simple, and not relying on reference points in the environment.

7 CONCLUSION

We presented *Chirpy*, an audio-based communication and localization device. Chirpy can estimate the direction of arrivals of audio signals. This feature enables a swarm of robots equipped with Chirpies to construct a guiding mechanism that can guide specialized robots, rescuers, or explorers through unknown terrains to a location of interest. Chirpies use orthogonal audio chirps to communicate concurrently (our implementation supports up to four simultaneous talkers). Despite the concurrent transmissions, the bit error rate is 1.4 % for fixed-length chirps at a distance of 250 cm between the transmitters and the receiver. To enable Chirpies to tackle novel environments we developed AudioLocNet: a Deep Learning-based audio source localization model. With AudioLocNet, Chirpies localize each other in a variety of environments such as a non-line-of-sight and reverberant environment. Our results show that AudioLocNet has ≈ 99% classification accuracy.

ACKNOWLEDGMENTS

This work has been undertaken in the Internet of Swarms project sponsored by Cognizant Technology Solutions and Rijksdienst voor Ondernemend Nederland under PPS O&I.

REFERENCES

[1] Sharath Adavanne, Archontis Politis, Joonas Nikunen, and Tuomas Virtanen. 2018. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. *IEEE Journal of Selected Topics in Signal Processing* 13, 1 (2018), 34–48.

[2] Hidri Adel, Meddeb Souad, Abdulqadir Alaqeeli, and Amiri Hamid. 2012. Beamforming techniques for multichannel audio signal separation. *arXiv preprint arXiv:1212.6080* (2012).

[3] Antouan Anguelov, Roumen Trifonov, and Ognian Nakov. 2019. Emerging and secured mobile ad-hoc wireless network (manet) for swarm applications. In *Proceedings of the 9th Balkan Conference on Informatics*. 1–4.

[4] Sylvain Argentieri, Patrick Danès, and Philippe Souères. 2015. A Survey on Sound Source Localization in Robotics: from Binaural to Array Processing Methods. *Computer Speech & Language* 34 (03 2015). https://doi.org/10.1016/j.csl.2015.03.003

[5] Christopher M Bishop. 2006. Pattern recognition. *Machine learning* 128, 9 (2006).

[6] C Sidney Burrus and TW Parks. 1985. Convolution Algorithms. *Citeseer: New York, NY, USA* (1985).

[7] Chao Cai, Henglin Pu, Peng Wang, Zhe Chen, and Jun Luo. 2021. We Hear Your PACE: Passive Acoustic Localization of Multiple Walking Persons. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 55 (jun 2021), 24 pages. https://doi.org/10.1145/3463510

[8] Chao Cai, Chen Zhe, Jun Luo, Henglin Pu, Menglan Hu, and Rong Zheng. 2021. Boosting Chirp Signal Based Aerial Acoustic Communication under Dynamic Channel Conditions. *IEEE Transactions on Mobile Computing* (2021).

[9] Soumitro Chakrabarty and Emanuël AP Habets. 2019. Multi-speaker DOA estimation using deep convolutional networks trained with noise signals. *IEEE Journal of Selected Topics in Signal Processing* 13, 1 (2019), 8–21.

[10] Changan Chen, Ziad Al-Halah, and Kristen Grauman. 2021. Semantic Audio-Visual Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15516–15525.

[11] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. 2020. SoundSpaces: Audio-Visual Navigation in 3D Environments. In *ECCV*.

[12] J.C. Chen, Kung Yao, and R.E. Hudson. 2002. Source localization and beamforming. *IEEE Signal Processing Magazine* 19, 2 (2002), 30–39. https://doi.org/10.1109/79.985676

[13] Daniel S Drew, Matthew Devlin, Elliot Hawkes, and Sean Follmer. 2021. Acoustic Communication and Sensing for Inflatable Modular Soft Robots. *arXiv preprint arXiv:2101.11817* (2021).

[14] Andrea Goldsmith. 2005. *Wireless communications*. Cambridge university press.

[15] Margherita Grandini, Enrico Bagli, and Giorgio Visani. 2020. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756* (2020).

[16] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. 2007. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics* 23, 1 (2007), 34–46.

[17] RE Halliwell, TRT Nightingale, ACC Warnock, and JA Birta. 1998. Gypsum board walls: Transmission loss data. *National Research Council of Canada, Internal Report No* 761 (1998).

[18] Hodaya Hammer, Shlomo E Chazan, Jacob Goldberger, and Sharon Gannot. 2021. Dynamically localizing multiple speakers based on the time-frequency domain. *EURASIP Journal on Audio, Speech, and Music Processing* 2021, 1 (2021), 1–10.

[19] Homayoun Hashemi. 1993. The indoor radio propagation channel. *Proc. IEEE* 81, 7 (1993), 943–968.

[20] Weipeng He, Petr Motlicek, and Jean-Marc Odobez. 2018. Deep neural networks for multiple speaker detection and localization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 74–79.

[21] K. C. Ho. 2012. Bias Reduction for an Explicit Solution of Source Localization Using TDOA. *IEEE Transactions on Signal Processing* 60, 5 (2012), 2101–2114. https://doi.org/10.1109/TSP.2012.2187283

[22] InspectorJ. 2017. Ambience, Large Crowd, A.wav. https://freesound.org/people/InspectorJ/sounds/403180/

[23] Soonwon Ka, Tae Hyun Kim, Jae Yeol Ha, Sun Hong Lim, Su Cheol Shin, Jun Won Choi, Chulyoung Kwak, and Sunghyun Choi. 2016. Near-ultrasound communication for tv's 2nd screen services. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 42–54.

[24] Amin Karbasi and Akihiko Sugiyama. 2007. A new DOA estimation method using a circular microphone array. In *2007 15th European Signal Processing Conference*. 778–782.

[25] Pooya Karimian, Richard Vaughan, and Sarah Brown. 2006. Sounds good: Simulation and evaluation of audio communication for multi-robot exploration. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2711–2716.

[26] Miquel Kegeleirs, Giorgio Grisetti, and Mauro Birattari. 2021. Swarm slam: Challenges and perspectives. *Frontiers in Robotics and AI* 8 (2021), 618268.

[27] Mohammad Omar Khyam, Md. Noor-A-Rahim, Xinde Li, Christian Ritz, Yong Liang Guan, and Shuzhi Sam Ge. 2018. Design of Chirp Waveforms for Multiple-Access Ultrasonic Indoor Positioning. *IEEE Sensors Journal* 18, 15 (2018), 6375–6390. https://doi.org/10.1109/JSEN.2018.2846481

[28] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[29] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf. 2011. A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE.

[30] Kurt Konolige, Giorgio Grisetti, Rainer Kümmerle, Wolfram Burgard, Benson Limketkai, and Regis Vincent. 2010. Efficient sparse pose adjustment for 2D mapping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 22–29.

[31] Hyewon Lee, Tae Hyun Kim, Jun Won Choi, and Sunghyun Choi. 2015. Chirp signal-based aerial acoustic communication for smart devices. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2407–2415. https://doi.org/10.1109/INFOCOM.2015.7218629

[32] Jihwan Lee, Chulyoung Kwak, Seongwon Kim, and Saewoong Bahk. 2020. Reliable and Low-Complexity Chirp Spread Spectrum-Based Aerial Acoustic Communication. *IEEE Access* 8 (2020), 151589–151601. https://doi.org/10.1109/ACCESS.2020.3017097

[33] Muhammad Usman Liaquat, Hafiz Suliman Munawar, Amna Rahman, Zakria Qadir, Abbas Z. Kouzani, and M. A. Parvez Mahmud. 2021. Localization of Sound Sources: A Systematic Review. *Energies* 14, 13 (2021). https://doi.org/10.3390/en14133910

[34] Amjad Yousef Majid, Casper van der Horst, Tomas van Rietbergen, David JohannesZwart, and R Venkatesha Prasad. 2021. Lightweight Audio Source Localization for Swarm Robots. In *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*. 1–6. https://doi.org/10.1109/CCNC49032.2021.9369572

[35] J.N. Moutinho, R.E. Araújo, and D. Freitas. 2016. Indoor Localization with Audible Sound - Towards Practical Implementation. *Pervasive Mob. Comput.* 29, C (jul 2016), 1–16.

[36] Akira Nakayama, Tamotsu Machino, Ikuo Kitagishi, Satoshi Iwaki, and Masashi Okudaira. 2002. Rich communication with audio-controlled network robot. Proposal of" Audio-MotionMedia". In *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*. IEEE, 548–553.

[37] Dušan Nemec, Aleš Janota, Marián Hruboš, Michal Gregor, and Rastislav Pirník. 2017. Mutual acoustic identification in the swarm of e-puck robots. *International Journal of Advanced Robotic Systems* 14, 3 (2017), 1729881417710794.

[38] Satoki Ogiso, Takuji Kawagishi, Koichi Mizutani, Naoto Wakatsuki, and Keiichi Zempo. 2015. Self-localization method for mobile robot using acoustic beacons. *ROBOMECH Journal* 2 (12 2015). https://doi.org/10.1186/s40648-015-0034-y

[39] Toivo Paavle, Mart Min, Jaan Ojarand, and Toomas Parve. 2010. Short-time chirp excitations for using in wideband characterization of objects: An overview. 253 – 256. https://doi.org/10.1109/BEC.2010.5631149

[40] KM Muraleedhara Prabhu. 2014. *Window functions and their applications in signal processing*. Taylor & Francis.

[41] Caleb Rascón and Ivan Vladimir Meza Ruiz. 2017. Localization of sound sources in robotics: A review. *Robotics Auton. Syst.* 96 (2017), 184–210.

[42] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. 2016. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics* 33, 1 (2016), 3–46.

[43] Seeed studio. [n. d.]. Grove - Speaker Plus. https://wiki.seeedstudio.com/Grove-Speaker-Plus/

[44] Seeed studio. [n. d.]. Respeaker 6-mic circular array kit for raspberry pi. https://wiki.seeedstudio.com/ReSpeaker_6-Mic_Circular_Array_kit_for_Raspberry_Pi/

[45] Gavin Suddrey, Pearl Gariano, Sam Cunningham-Nelson, Daniel Richards, and Frederic Maire. 2014. Audio signalling as a backup communication channel for multi-robot systems. In *Proceedings of the 16th Australasian Conference on Robotics and Automation 2014*. Australian Robotics and Automation Association Inc., 1–9.

[46] Juan Manuel Vera-Diaz, Daniel Pizarro, and Javier Macias-Guarasa. 2018. Towards end-to-end acoustic localization using deep learning: From audio signals to source position coordinates. *Sensors* 18, 10 (2018), 3418.

[47] Weiguo Wang, Jinming Li, Yuan He, and Yunhao Liu. 2020. Symphony: localizing multiple acoustic sources with a single microphone array. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 82–94.

[48] Pengwei Xu, Elias JG Arcondoulis, and Yu Liu. 2021. Acoustic source imaging using densely connected convolutional networks. *Mechanical Systems and Signal Processing* 151 (2021), 107370.

[49] Le Yang and K. C. Ho. 2009. An Approximately Efficient TDOA Localization Algorithm in Closed-Form for Locating Multiple Disjoint Sources With Erroneous Sensor Positions. *IEEE Transactions on Signal Processing* 57, 12 (2009), 4598–4615. https://doi.org/10.1109/TSP.2009.2027765

[50] Cha Zhang, Dinei Florencio, Demba E. Ba, and Zhengyou Zhang. 2008. Maximum Likelihood Sound Source Localization and Beamforming for Directional Microphone Arrays in Distributed Meetings. *IEEE Transactions on Multimedia* 10, 3 (2008), 538–548. https://doi.org/10.1109/TMM.2008.917406