AudioLocNet

Deep Neural Network Based Audio Source Localization for Inter Robot Localization

Casper van der Horst

Master's thesis

TU Delft

AudioLocNet

Deep Neural Network Based Audio Source Localization for Inter Robot Localization

Master thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of
Master of Science in Embedded Systems
Faculty of Electrical Engineering, Mathematics and Computer Science
by
Casper van der Horst
Student number: 4472136

To be defended in public on December 9th 2022

Graduation committee

Dr. R. R. Venkatesha Prasad,	Delft University of Technoloy
Dr. R. T. Rajan,	TU Delft University of Technoloy
Dr. A. Y. Majid,	Delft University of Technoloy

TU Delft

Preface

From a young age, I had an interest in technology and there are stories of me stating that I wanted to go to the TU Delft from when I was in middle school. This thesis forms the culmination of my time there and a springboard for my future full of technology and new challenges.

I could not have reached this point without the help of the people around me. For this, I would like to thank the following people. First and foremost, my daily supervisor Dr Amjad Yousef Majid and professor Ranga Rao Venkatesha Prasad, who enabled me to do the research resulting in this thesis.

Secondly, I would like to thank Lucan de Groot, Mees Jonker and Nils van den Honert for their (continued) work with the Chirpy robots. And Barend van de Wal for proofreading my thesis.

Lastly, I would like to thank my family and friends for their support and encouragement during these last few years.

Abstract

For my Master's thesis, I developed and trained an audio-based localization system for indoor localization called AudioLocNet. AudioLocNet is based on convolutional neural networks and maps recordings from a small(10 cm diameter) microphone array to a grid of locations around said array. AudioLocNet was made to be used by swarms of small robots to locate each other using audio signals. AudioLocNet was trained using orthogonal chirp signals which have a low cross-correlation. Said signals can also be used for simultaneous communications between multiple robots. These signals were recorded in indoor environments ranging from simple line-of-sight environments to reverberant non-line-of-sight ones. Audio signals are used since they form a propagational middle class when compared to radio frequency (RF) and light-based signals for localization. Whereas light requires a line of sight, audio can bend around corners; and whereas RF signals pass through walls, reaching robots that are outside of each other's spheres of influence, audio will not.

AudioLocNet reaches high accuracies for both a coarse grid (99.96 %) and a fine grid (99.89 %) of possible locations, where only the final layer of the network architecture must be changed to account for the increased resolution of the fine grid.

Casper van der Horst
Delft, November 2022

Contents

1 Introduction

Swarm robotics is a field that focuses on methods of solving robotic problems by using large numbers of small, non-complex robots instead of large highly specialized ones. Although each individual robot is unable to complete the task efficiently or (often) at all, the interactions between them result in a behavioural pattern that can not only complete the task at hand but can also result in more efficient solutions than what a single (more complex) robot would be capable of. These so-called emergent behaviours emerge from interactions between members of a group, and can often be found in nature, which is where this field of study finds much of its inspiration. Ants are well known for showing swarming behaviours. They can use themselves to build structures like bridges [50] and scaffolding [40] in order to make resource collection more efficient, where the ants collectively find an optimum between increasing the foraging performance of the workers and the workers lost as building materials for the structures. Bees also show impressive feats of emerging behaviours. Clusters of bees are capable of regulating the internal temperature of clusters spanning more than 10,000 individual bees. This is achieved by expanding or contracting the cluster size. This way the swarm can hold a constant internal temperature over a large range of outside temperatures [47]. Not all examples of emergent behaviour come from the world of insects, as illustrated by the flocking/schooling of birds and fish [29]. These communal movements are governed by individual members balancing avoidance, alignment and attraction behaviours between themselves and nearby neighbours based on how close they are to those neighbours.

The main attributes of a robotic swarm are its scalability, flexibility and robustness, which, respectively, means that swarms can deal with various numbers of members, can adapt to changing environments and are able to operate even as individual members break down. This results in swarms acting in decentralised and local manners in regard to communication, localization and decision-making.

For a swarm to function, it is important that the members can locate each other. Popular inter-robot localization methods for swarm robots include (infrared) light [1, 4, 5, 20, 25, 42, 45, 52, 65] and radio frequency (RF) based [11, 16, 43, 46, 57, 60] methods. Although there are papers on audio localization on robots [48], these focus mainly on locating other sources around the robot or on self-localization based on locating a beacon source with a known location [6].

1.1. Motivation

This thesis investigates whether audio signals can be used for inter-robot localization for small microphone arrays. The main motivation for audio as a localization medium comes from how it has a different observation space than RF or light-based signals. The following properties show how this observation space is different:

- **Low communication range.**
  In the open air, two phenomena work on sound to decrease its intensity, firstly the geometric decrease due to the inverse square law, according to which the sound intensity is inversely proportional to the square of the distance to its source; and the attenuation from the air itself. The attenuation of the amplitude of a signal is determined by Equation 1.1, with A_0 the initial amplitude and x the distance travelled. The attenuation factor is determined using Equation 1.2, where η and η^v are the dynamic and volume viscosity, respectively, ω is the angular frequency of the sound, ρ is the density of the medium and V is the speed of sound in said medium. For RF signals under 10 GHz attenuation due to the air is negligible [30], meaning that (barring obstacles), radio frequency signals can travel further than audio

signals.
Since swarm robots work on local data, a signal which reaches too far can result in a robot using processing resources to decode messages from other robots which are outside of its sphere of influence and which it has no use for.

$$A(x) = A_0 * \exp -ax \tag{1.1}$$

$$a = \frac{2\left(\eta + \eta^{\nu}\right)\omega^2}{3\rho V^3} \tag{1.2}$$

- **High attenuation by walls.**
  Another major contributor to the differences between communication ranges between RF and audio signals in indoor environments comes from how they respond to walls. [23, 27] give partition losses for double plasterboard walls in the range of 3.4–3.8 dB, whereas research from the National Research Council Canada [26] on sound propagation through a plethora of different indoor wall materials and constructions had the following results (Figure 1.1) for a dual plasterboard wall with no insulation. This shows that after passing an indoor wall, the power of an RF signal is 10–70 dB less attenuated than a sound signal. This would make the RF signal 16–107 times stronger than an audio signal transmitted at the same power level. This means that two robots utilizing audio localization on opposite sides of a wall will have a much harder time detecting each others presence when compared to RF based localization. Since these robots are unable to interact (due to the presence of the wall), there is no need to have to use computational resources on locating said other robot.

- **Bending around obstacles.**
  The above two properties showed how the use of audio helps in reducing the observation space when compared to RF signals. However, by using audio the observation space may acctually be increased when compared to light-based signals (that behave similarly to audio in regards to the previously stated properties). Audio signals bend around the corners of walls or other obstacles. Light rays, however, do not demonstrate this behaviour. The advantage of such bending behaviour of audio signals becomes clear with the following scenario. Imagine two robots approaching each other at right angles and a wall blocking the line of sight between them. If these robots were locating each other using light-based signals, then they would only learn of each other's existence right before a possible collision when it might be too late to prevent a crash! If, on the other hand, these robots were using audio signals, they could notice each other in advance. After locating each other they could use a communication method or a set of traffic rules to determine which robot passes first.
  Another advantage that comes from bending around obstacles instead of going through them, is that when two robots locate each other. The determined sound direction points towards the path the sound took around the obstacle.

There are two more advantageous properties to the use of audio over light or RF. Firstly, unlike light-based signals, audio is not influenced by environmental lighting conditions. IR-based sensors can be influenced by other IR sources like sunlight as shown by [49]. Another lighting concern comes up when using cameras, since these require light to hit the sensor, it forces a user to illuminate the operational area or equip each robot with lighting of its own. By using audio a hypothetical swarm-powered warehouse could operate in the dark, saving on the total energy consumption of the building. Lastly, when audio signals with frequencies in the human audible range (20 Hz–20 kHz) are used, humans may use their own ears to locate (or at least determine the presence of) robots in their surroundings. Simultaneously, a robot equipped to locate things based on sound could locate an operator based on sound signals made by said operator. This human-robot awareness can aid in making systems where humans and robots work together on a task.

1.2. Problem Statement

With the stated properties of sound it becomes an interesting localization medium when the focus is on short range, indoor localization where one only wants robots to interact if the traversable distance between them is low, but when there is no line of sight between them. As such the main research question is:

- How can we leverage audio and AI to create a localization method for inter-robot localization?

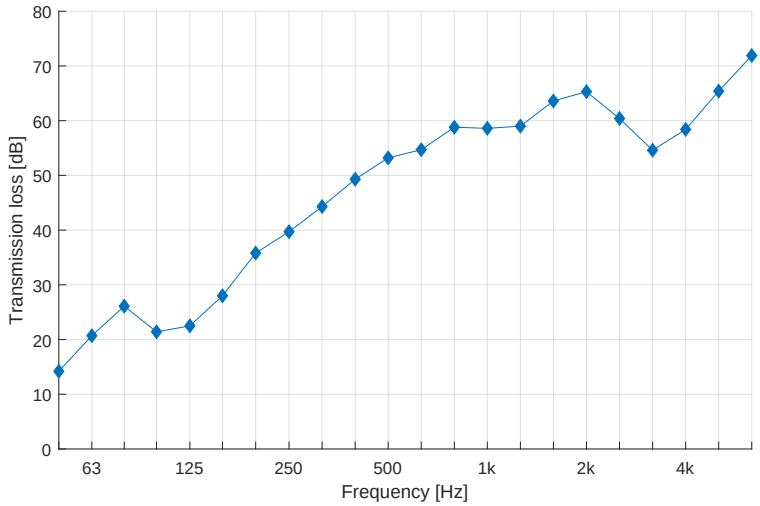


Figure 1.1: Transmission loss for sound signals with different frequencies, for a double plasterboard wall with no insulation in between the panels [26].

1.3. Key Contributions

For this thesis, I designed and trained a neural network-based audio source localizator which uses a small (10 cm diameter) microphone array consisting of six microphones to locate sources up to 250 cm away. This network, called AudioLcoNet, was also used in [68] where we made a complete system for inter-robot audio-based communication and localisation, where the communication were also used to facilitate localization. The novelties of this AudioLocNet lie in:

- the use of orientation-aware input padding to account for the circular nature of the microphone array;
- that AudioLocNet outputs both distance and direction from a small microphone array; and
- that the network is able to localize signals from sources without a line of sight between the source and the microphones.

Additionally, the recordings used to train the networks were made in the real world using actual hardware. These recordings are made available to others to use to train their own networks.

1.4. Limitations

- This study doesn't consider the computing power of the robots, as such timing and real time performance are outside of the scope of this work.
- No research was done into the effects of outside noise, Although the recordings were made in the real world with no additional noise was added to decrease the signal to noise ratio of the recordings.

1.5. Thesis Structure

The remainder of this work is structured as follows: firstly, Chapter 2 will discuss existing works and background in the fields of swarm robotics and different localization systems. Chapters 3 and 4 explain how the different parts of the system work and the data gathering and training process, respectively. Chapter 5 discusses the results of testing the network and its performances. Lastly, in Chapter 6, I will give some closing remarks on the system as a whole.

2

Background and related work

This section discusses the state of the art in terms of audio source localisation and swarm robotics

2.1. Swarm robotics

In nature social insects like ants, bees and birds show an ability to perform complex tasks which are far beyond the capabilities of individual swarm members. These tasks are performed by the group without any centralised oversight [9] and their execution is the result of each swarm member following relatively simple rules. The group behaviour following from these individual rules is called an emergent behaviour [39]. Swarm robotics is focussed on how to coordinate swarms of simple robots and the emergent behaviours resulting from the interactions between them [54].

Such swarms, whether natural or robotic, poses the following main advantageous properties [15]:

- **Robustness.** Robustness encompasses the ability of the swarm to continue operation in the face of losing members, this can be achieved in many different ways, such as having enough redundancy in the number of robots that a failed individual can be replaced by another. Another aspect of robustness comes from the decentralised nature of the swarm, this means that there isn't a specific "brain" of the swarm which would render the swarm useless when lost. Lastly, the low complexity of the individual swarm members helps in preventing errors in individual robots.

- **Flexibility.** Flexibility refers to the ability of the swarm to adapt to new, different or changing requirements of the environment. In nature, redundancy, behavioural simplicity and task allocation promote the flexibility of a swarm [7].

- **Scalability.** A scalable swarm is able to function with differing numbers of members, and their behaviours should support large groups of members. The use of local communication and sensing methods is therefore important for a scalable swarm. As global communication methods can start to struggle when large numbers of swarm robots try to use them.

In order to solve a complex task using a single robot, that robot needs to be designed with a complicated structure and control modules which results in a high cost of design, construction and maintenance [63]. This may also result in a single point of failure for the task, where if a part of the robot breaks down, the entire task completion process could be at risk. When using a robotic swarm for such a task, many of such problems can be mitigated, as illustrated by the following advantages.

- **Parallelism.** When a task comprises multiple parts distributed over an environment, the size and distributed nature of a swarm allows it to complete the multiple parts simultaneously, speeding up the process. Search tasks also greatly benefit from a swarm's parallel capabilities, as the swarm can search at multiple places at once.

- **Scalability.** The locality of the interactions within a swarm enables it to handle changes in the swarm size without disturbing the operation of the swarm as a whole. This also means that it is not required to change the hardware or software when robots join or leave the swarm.

- **Cost.** The individual swarm robots can be made simpler which helps to reduce the cost of the entire system. Additionally, since a swarm consist of large numbers of robots it can benefit from the economies of scale to reduce the cost even further.
- **Stability.** The scalability of a swarm gives it a matter of redundancy. This means that it remains able to complete tasks (albeit with reduced performance) when members malfunction or break down. This, in combination with the cost advantage, makes swarm robots well-suited for environments which are dangerous to the robots themselves, as the cost of losing a robot is low and isn't detrimental for task completion.
- **Energy efficiency.** Another advantage of their small scale and simplicity is that smaller robots are usually more energy efficient than large ones. Also, when a swarm requires fewer members its scalability principles allow it to shut down members which do not have a task, to reduce the total energy usage.

2.2. Relative localization

In order to work together, each swarm member needs to know where the other members are. This is both for collaboration (e.g. not checking the same area multiple times in a search task) and for collision avoidance. Therefore a swarm robot system requires some localization method such that each robot can localize the other swarm members. Localization techniques can be split into two types, global and relative localization techniques.

Global localization techniques are techniques where the locations of all robots are determined by a central/global entity which transmits the coordinates to the robots. Examples of such a system include having robots equipped with markers and motion capture cameras with accompanying software to communicate the locations to the robots [18, 31], and using ultra-wideband (UWB) localization systems comprising fixed anchors and tags [75]. Although these methods give good localization accuracies and speeds for smaller numbers of robots, their reliance upon global communication methods and centralised localization systems means that they do not match well with the scalability property of a swarm robotic system.

In relative localization methods, each robot localizes the other robots in its direct vicinity, these methods are also referred to as local localization methods. These methods are more suitable for scalable swarms since they do not use global communications or centralised systems. The method described in this thesis is such a relative method, as all robots make their own sounds which the surrounding robots can detect. As such there is no single entity running the calculations to locate all members, nor is this method reliant on beacons which have known locations.

2.2.1. Light based

The most popular localization methods include infrared (IR) light transmitters and receivers. Like the Kobots [65] which use a sensing scheme to which comprises a *kin-detection* phase and a *proximity-sensing* phase. In the *proximity-sensing* phase the robot transmits directed modulated IR light at varying power levels and receives reflections using directed IR receivers. Based on the power of the recorded reflections, the robot determines that there are obstacles in certain directions and at certain distances. During this phase, other robots which are in the *kin-detection* phase would receive the IR signal from the first robot and use that to determine the location of the first robot. In order to prevent cross-talk, a carrier sense multiple access with collision avoidance (CSMA-CA) scheme is used.

The robots from the Jasmine project [1], AMiR [4], Colias [5] and R-One [42] use similar techniques for neighbour localization where a ring of IR transceivers outputs signals to be detected by the IR transceivers of the neighbouring robots. These systems determine the direction towards the transmitting robot based on which transceiver receives the signal. Therefore the accuracy of such systems depends on the number of transceivers used. Often these systems also include local communications into the transmitted IR signals, merging localization and communication. The popularity of this method is expressed further by works like [25] where an open-source localization and communication system was developed which could be added to any swarm robot to enable IR-based localization and communication.

Another implementation of IR for relative localization comes from the Kilobots [52], these robots are able to manoeuvre over a 2D plane to form shapes. In order to keep the size small (each robot is 3.3 cm in diameter) and the cost low, the designers decided to work with only the distance to and not the direction towards the other swarm members. Therefore only a single IR transceiver is required per kilobot. This transceiver is pointed downwards and the other kilobots receive the IR signal reflected from the ground (Figure 2.1). The

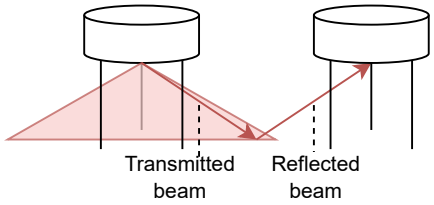


Figure 2.1: Localization via reflected IR from Kilobots[52]

signal strength (i.e. light intensity) of the received signal is used to determine the distance. Similar to the Kobots, Kilobots use CSMA-CA to prevent cross-talk.

Another light-based localization system can be seen in the SWARM-BOTS project [20]. This system comprises S-bots which are, along with other sensors, equipped with LED rings and omnidirectional cameras. The cameras are used to localize the illuminated led rings. Nouyan et al. [45] show two methods with which S-bots can use their cameras and LED rings to form paths from a *nest* location to a *prey* location, where neither location is known beforehand. In the first method, the robots first need to find the *nest* or an existing path. After which they either form a new path or move to the end of the path to join it. Robots that are part of the path have one of three colours (blue, yellow or green) and said colour is determined by the colour of the robot in front of it in the path. This creates a cyclic colour pattern which can be used by other robots to navigate towards the end of the chain. In the second method robots which are part of a path form a vector field that points along the path towards the *nest*. This pointing is done by lighting up specific sectors of the LED ring. Any robot finding a path can then use the vector field to navigate to the end and extend the path.

Apart from the popular light-based localization methods, other relative localization methods can also be observed, some of which are discussed below.

2.2.2. Ultra-wideband

UWB communication methods use radio frequency (RF) signals to communicate data between transmitters and receivers. These signals cover a large frequency bandwidth which enables the use of a large total signal power without interfering with narrowband signals. UWB-based localization techniques are gaining in popularity, especially in environments where satellite-based localization methods (like GPS) struggle. Above, it is mentioned how UWB-based localization may be used as a global localization method requiring fixed anchors which localize moving tags. However, works like those of Morón et al. [43] and Stier et al. [60] are based on different methods where robots locate themselves and others relative to movable anchors, which are also part of the robot swarm.

The most popular metric for UWB localization is called the time of flight (ToF). ToF-based techniques firstly determine the propagation time of a ranging message between a transmitting node and a receiving node, and then use the speed of light to compute the distance between them. Such ranging messages contain timestamps of when they were transmitted. These techniques fall in either of two categories, one-way ranging (OWR) or two-way ranging (TWR), based on whether clock synchronisation between the two nodes is required. In OWR the ranging message is transmitted in a single direction and the receiving node computes the ToF based on the transmission timestamp from the message and the receive time of the message at the receiving node. OWR requires that both nodes have their clocks synchronised. This makes these techniques more complex and therefore less popular than the TWR methods. In TWR a response message is returned to the transmitting node by the receiving node. Said response message carries the required processing time $T_{p,1}$ that the receiving node needed to return the message. The transmitting node can then determine the round trip $T_{r,1}$ by comparing the time of when it send the ranging message with the time it received the response. The transmitting node can then compute the propagation time T_{prop} using Equation (2.1). This technique is referred to as Single-Sided TWR (SS-TWR). If the transmitter sends another response message to the receiver node with its own processing time $T_{p,2}$, then the receiving node can compute its own round trip time $T_{r,2}$. This technique is then referred to as Double-Sided TWR (DS-TWR) and T_{prop} can then be determined via Equation (2.2). DS-TWR tends to result in more accurate estimations than SS-TWR [57]. Figure 2.2 depicts both the SS-TWR method and the DS-TWR method and illustrates when the different time measurements are taken.

$$T_{prop} = \frac{1}{2}(T_{r,1} - T_{p,1}) \tag{2.1}$$

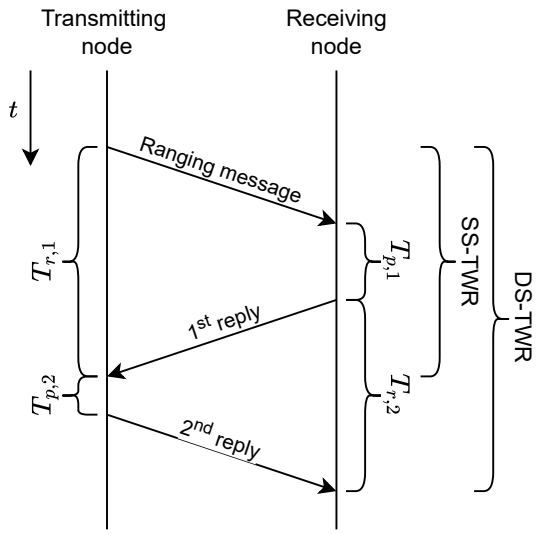


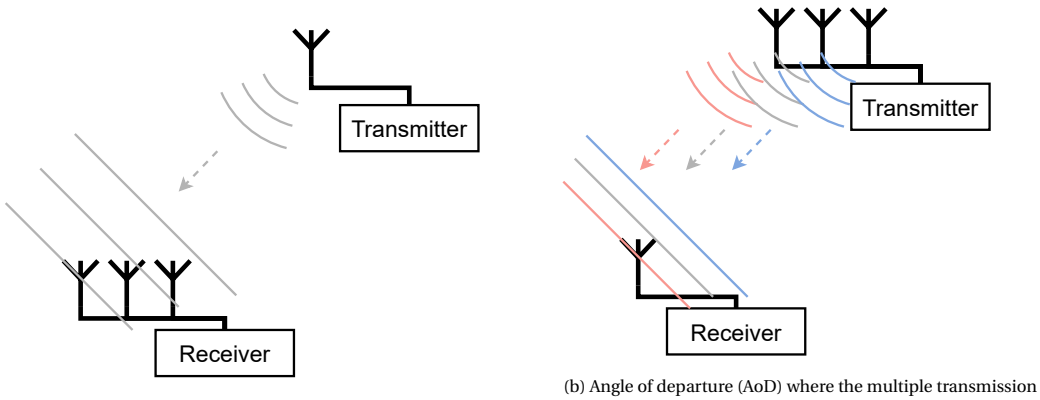
Figure 2.2: Communication process and timing definitions for SS-TWR and DS-TWR

$$T_{prop} = \frac{T_{r,1} \cdot T_{r,2} - T_{p,1} \cdot T_{p,2}}{T_{r,1} + T_{r,2} + T_{p,1} + T_{p,2}} \tag{2.2}$$

One of the main challenges of TWR methods is their lack of scaling. This is due to the time slots required to perform a TWR exchange which needs to be scheduled to prevent collisions with other transmissions, where these slots also have limited time bandwidths. This drastically decreases the communication frequency with each added node [43]. Stier et al. [60] tackle this by developing a dynamic and more localized TDMA-based slotting system to give the pairs of nodes the required time slots for TWR. The strength of their protocol lies in how it handles joining and leaving nodes and in that it handles nodes that are in motion. A different solution comes from Morón et al. [43], who propose a (dynamically allocated) split of active and listener nodes, wherein the smaller subset of active nodes comprises the nodes which form a convex envelope around the listener nodes. The active nodes use TWR to locate each other. The listener nodes, on the other hand, eavesdrop on these TWR exchanges to obtain the distances between the active nodes and to determine a time difference of arrival (TDOA) between the signals from the active nodes. The listener nodes are then able to use the TDOA to locate themselves relative to the active nodes.

2.2.3. RFID

Radio-frequency identification (RFID) technology is built around tags and readers. The RFID-tags comprise radio transponders and small memories to store identifying information. If a reader transmits an RF signal towards a tag, then the tag automatically responds by returning its identifying information. RFID tags can be divided into two categories, active and passive, based on their power source. Active tags contain batteries to supply the energy for the transmission; whereas passive tags harvest the energy received from the signal transmitted by the reader for the reply. Due to having their own power source, the signals coming from active tags are stronger than those from passive ones, resulting in larger transmission ranges going up to 100 meters. The passive tags on the other hand can be a lot smaller and lighter due to not having a battery, this makes them well suited for integration in other products like library books or identification cards [71]. The low-powered and cheap nature of RFID tags makes them an appealing option when a large number of products need to be tracked. Therefore Charléty et al. [11] utilize a network of active RFID tags distributed over a 30 × 30 meter area of an active landslide to track the movement of said landslide over a period of 11 months. Here the motion of the tags was recorded relative to a nearby antenna array. Most RFID localization methods require complex calibrations of the antennas to produce accurate localization results. Patel and Zawodniok [46] trie to remove this requirement by implementing a deep learning-based RFID localization method. Their method uses beam steering with four antenna sections connected to a single RFID reader to create "images" depicting the phases and signal strengths returned by the passive tags. A convolutional neural network then uses these images to estimate the location of the tag relative to the antenna. Although the use of low-power tags (even zero power in the case of passive tags) for localization is very appealing for many fields (like warehouses and



(a) Angle of arrival (AoA) where the time differences between the reception of the same message at different antennas are used to determine the angle of arrival.

(b) Angle of departure (AoD) where the multiple transmission antennas create time delays between copies of the same message at the receiver, these delays are used to derive the angle of departure relative to the transmitter.

Figure 2.3: Direction finding methods included since Bluetooth version 5.1

motion tracking), these techniques do not work well with a swarm robotic system. This is because all the localization calculations take place on the reader device, which should then communicate the calculateed locations to all swarm robots. This breaks the scalability aspect of swarm robots.

2.2.4. Bluetooth

Bluetooth is a well-known communication standard whose signals occupy the 2.40–2.42 GHz band. Rudimentary localization in the form of distance sensing has been commercially available since 2013 with the introduction of Apple's iBeacon and Google's Eddystone (introduced in 2015) protocols. These protocols are for locating Bluetooth low energy (BLE) beacons. BLE beacons are Bluetooth transmitters which broadcast messages which include their own identifier and a reference transmission power. This reference transmission power tells a Bluetooth receiver (like a smartphone) the expected signal power at a predefined distance from the transmitter. With this reference and the received signal strength indicator (RSSI), the receiver can estimate its distance to the beacon.

Coppola et al. [16] show how to use such RSSI measurements to avoid collisions between multiple flying robots. In their system, Bluetooth advertising messages are used to have each robot broadcast its altitude and velocity (relative to magnetic North). The RSSI of said message as recorded by a receiver is then merged with the data using an extended Kalman filter to determine the location of the transmitter relative to the reciever. This localization scheme powered a collision avoidance system which was able to drastically increase the time to collision with up to 3 drones in an enclosed space.

Since Bluetooth version 5.1 (January 2019) direction finding capabilities were added to the standard. The thusly included angle of arrival (AoA) and angle of departure (AoD) methods allow for, respectively, multi-antenna receiver or transmitter arrays to be used to have the receiver determine the direction that the transmission came from (Figure 2.3). In the AoA case, the receiver samples each antenna to determine the phases between the recorded signals. In the case of AoD, it is still the receiver that determines the phase differences resulting from the multi-antenna transmitter. However, this does require that the receiver has knowledge of the physical configuration of the transmitting antenna array. Using the dimensions of the multi-antenna array used (by the receiver for AoA or the transmitter for AoD) and the determined phase differences, the receiver can determine the direction that the received message was transmitted from. In order to facilitate direction-finding, the BLE standard includes so-called direction-finding signals. These signals contain a constant tone extension (CTE) is a pure tone that the receiver uses to determine the direction of the transmission [74]

Toasa et al. [64] tested the AoA method and showed that it resulted in RMS errors below 0.5° if the source was within 30° from the normal of a linear antenna array comprising 4 antennas. Ye et al. [77] showed that a sub-meter accuracy is possible when using a planar antenna array (12 antennas forming a square border) facing a plane of BLE transmitters. On the AoD front, Shin et al. [56] show that with only two transmitting antennas a mean angular error of 2.5° can be reached.

2.2.5. Wi-Fi

Wi-Fi signals are omnipresent in modern-day indoor environments and, as more and more devices are wirelessely connected to the internet, this presence will continue to grow. Since Wi-Fi also uses RF signals, research in localizing these signals is also conducted.

Back in 2018 Soltanaghaei et al. [58] showed that it was possible to use multi-path reflections to enable a single multi-antenna receiver to locate a transmitter without explicitly communicating with said transmitter. They firstly estimate the AoA, AoD and ToF of the line of sight path, and also of several reflected paths. The channel state information (CSI) is used for these estimations. With these parameters for multiple paths, it is possible to determine not only the location of the transmitter but also its orientation and the location of the reflecting surface of the reflected path. It is most notable that this method does not require explicit communications between the initial transmitter and receiver, meaning that the localization can be performed by a third party listening to Wi-Fi traffic between other nodes.

Soltanaghaei et al. [59] try to bypass the need for dense RFID antenna networks needed for RFID-based localization by leveraging existing Wi-Fi infrastructure. They introduce TagFi, a system comprising backscatter Wi-Fi tags. These tags use backscattering to modulate arbitrary signals coming from a common Wi-Fi access point. A receiving user device (like a phone or laptop) then receives/eavesdrops on the arbitrary signals and their modulated counterparts to identify and localize the tag.

2.3. Audio source localisation

Unlike RF and light based signals, which propagate via electromagnetic waves, sound propagates via vibrating the molecules of the medium it is in. This results in sound waves that struggle with going through solid objects and that do not propagate far. While at the same time these vibrations are more capable of bending around obstacles, meaning that the sound traverses the path without obstacles. For these reasons this thesis uses audio as a medium for localization.

Audio source localisation comprises estimating the location from which an audio signal originates, relative to the listener. The source positions are generally estimated in two parts: the Direction-of-arrival (DoA) and the distance. These two parts are handled separately [48].

2.3.1. Classical localization

Most classical methods work in two phases: first they extract specific features from the recordings and then they apply a feature-to-location mapping in order to estimate the location of the audio source based on the extracted features. This mapping relation relies on a sound propagation model which models what the recorded features look like based on the audio source location. Techniques which use such a structure will be referred to as classical techniques as these do not use machine learning techniques in their localisation.

Feature Extraction

The first step of audio source localisation comprises the extraction of features. The following are some of the more popular features:

- Time difference of arrival (TDOA). This denotes the time difference between the arrivals of the same signal at different microphones, where those microphones located closer to the source receive the signal earlier than those located further away. Based on the signal and the microphone setup this can also be called the inter-aural time difference (ITD), for systems with 2 microphones using pinnae [1], or the inter phase difference (IPD) for when a narrowband signal is recorded. A popular way of determining the TDOA is via cross-correlation, where the GCC-PHAT technique [36] is the most popular.

- Inter-microphone intensity difference (IID). The IID is the energy difference between two signals at the same point in time. This feature uses the attenuation difference between microphones at different distances to aid in the localization. If the energy differences at specific frequency components are extracted, then it is referred to as the Inter-microphone level difference (ILD), in which case the energy spectra are compared.

- Spectral notches. Microphones which use pinnae will have certain frequencies amplified or attenuated due to reflections of audio signal against the auricle. The locations of these spectral notches, on the frequency spectrum, depend on the location of the audio source [32].

[1] microphones with a synthetic auricle to mimic an ear

It is also possible to use multiple features in conjunction with each other and when the feature set is a combination of the IPD and the ILD then it is referred to as binaural or spectral cues. This combination is mostly used when an artificial head with ears (simulating a human head) is used, where (due to interference and obstruction of the head) high and low-frequency sounds are primarily localized using the ILD and ITD features respectively [72].

Feature-to-Location Mapping

The propagation models used in the feature-to-location mapping depend highly on the extracted features, the geography of the microphone array in relation to the source locations and environmental characteristics like reverberance and obstructions. The following 3 types of propagation models are used often:

- **Free-field/far-field model.** This is the most popular model since it works from two simple assumptions. The free-field assumption states that there are no reverberations or obstacles between each source and microphone. Therefore every possible source has a single, straight path to each microphone. By using the far-field assumption the model assumes that the distance between the source and the microphone array is sufficiently large relative to the diameter of the microphone array that the sound waves can be modelled as being planar instead of spherical.
- **The Woodworth-Schlosberg spherical head model** [73] models sound waves propagating over a spherical head and is often used when microphones are placed on robotic heads.
- **The near-field model** is used when the sound source is expected to be near the microphone array relative to the diameter of the microphone array. At such distances, the waves propagating from the source must be modelled as being circular. The circular waves make the computation more complex. This model is not used too often and there are even works which attempt to modify the far-field model to be usable in the near field [3, 66]. However, using this model also allows for the simultaneous determination of both the distance and DOA of a source, as shown by Chen et al. [14]

For some mapping procedures, it is possible to directly map the features to specific locations. However, for some features, it is required to test for different locations to see whether the expected results match the recorded features. Such mapping procedures make use of a grid-search to find the source locations and are especially prevalent when multiple sources are to be located at the same time.

Two other popular sound source localization techniques are MUSIC [55] and the delay-and-sum (DAS) beamformer [19]. These do not extract specific features as mentioned before but use subspace orthogonality and steered power response respectively to search for source locations. These methods will be briefly explained due to their popularity and differing structure when compared to techniques mentioned before, though it should be noted that these techniques are also used in non-audio-based techniques, as they work on the recorded signals coming from multiple sensors.

MUSIC

Multiple signal classification (MUSIC) is a subspace method for DOA determination. Instead of extracting specific features from the recordings, MUSIC works directly from the frequency domain representations of the recording. MUSIC represents the received signal as Equation 2.3, where $\boldsymbol{X}$ contains the frequency domain representations (in F frequency bins) of the recordings from the M microphones in an $M \times F$ matrix; $\boldsymbol{S}$ represents the frequency domain representations of the D transmitted signals in a $D \times F$ matrix and $\boldsymbol{W_s}$ represents the TDOAs from each signal to each microphone in an $M \times D$ matrix. Lastly, the $M \times F$ matrix $\boldsymbol{V}$ represents the noise received on each microphone.

$$\boldsymbol{X} = \boldsymbol{W}_s\boldsymbol{S} + \boldsymbol{V} \tag{2.3}$$

MUSIC tries to estimate the TDOAs of the D signals with the following steps. First, eigendecomposition is performed on the sample covariance matrix $\hat{\boldsymbol{R}}$ of $\boldsymbol{X}$. Resulting in the decomposition of $\hat{\boldsymbol{R}} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1}$. Here $\boldsymbol{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\hat{\boldsymbol{R}}$ (in descending order) and $\boldsymbol{Q}$ contains the eigenvectors.

Based on the sizes of the eigenvalues, $\boldsymbol{Q}$ is then split into the matrices $\boldsymbol{Q}_s$ and $\boldsymbol{Q}_v$ containing the first λ_s and the remaining $M - \lambda_s$ columns, and representing the signal and noise subspaces respectively, wherein the index value λ_s is chosen as an index of the smallest value of a group of large, non-zero eigenvectors. Here MUSIC assumes that the large eigenvalues comprise the signal subspace and the small eigenvalues comprise the noise subspace.

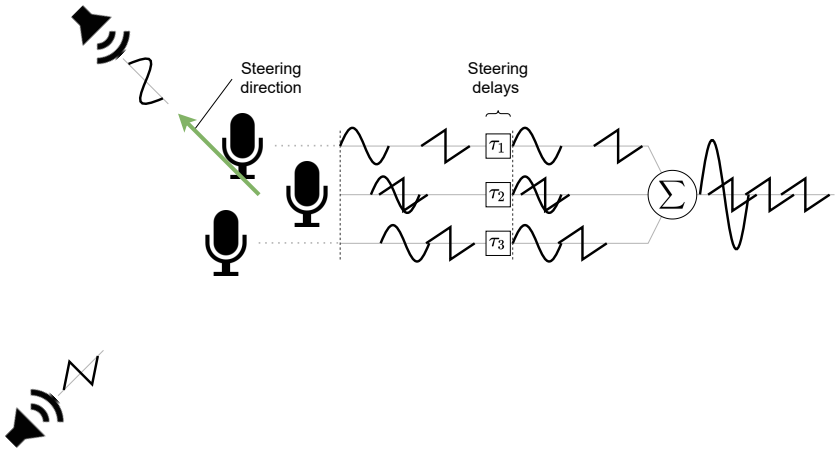


Figure 2.4: Overview of a delay and sum beamformer, where the beam is steered towards one of the sources.

MUSIC finds the DOA by performing a grid-search with different direction candidates for orthogonality with the subspace of noise eigenvectors $\boldsymbol{Q}_v$. Each direction candidate θ comprises an $M \times 1$ vector $\boldsymbol{b}(\theta)$ with expected TDOAs for each of the M microphones given the candidate θ, wherein the TDOAs of $\boldsymbol{b}(\theta)$ are computed based on a propagation model like those described above. If $\boldsymbol{b}(\theta)$ points into the direction of a source in the source subspace, then, due to the orthogonality between the eigenvectors of the subspaces, $\boldsymbol{b}(\theta)$ will be orthogonal to $\boldsymbol{Q}_v$. This orthogonality is tested with Equation 2.4. When plotted over multiple candidates $P(\theta)$ will form peaks at the directions which correspond to source locations.

$$P(\theta) = \frac{1}{\boldsymbol{b}(\theta)^H \boldsymbol{Q}_v \boldsymbol{Q}_v^H \boldsymbol{b}(\theta)} \tag{2.4}$$

Beamforming

The steered response power localization method from [19] is an example of a beamforming-based localization method. In beamforming, a sensor array is electronically steered in a specific direction by using spatial filters on the signal lines of the array elements. For audio localization, these sensors are the microphones, but beamforming is also used in other fields like signal transmission and reception of RF signals using antennas [69]. The delay and sum (DAS) beamforming technique is the simplest beamforming technique [48] and has been shown to work for audio signals [38].

An overview of DAS beamforming is depicted in Figure 2.4. Reciption of a transmitted signal $s(t)$ at microphone m of a microphone array can be modelled as $x_m(t)$ in Equation 2.5, where a_m represents the attenuation of the signal, $v_m(t)$ the noise or other signals received at the microphone, t_s the propagation delay from the sound source to a predetermined reference microphone of the microphone array and τ_m the TDOA between microphone m and the reference microphone (which can be either negative or positive).

$$x_m(t) = a_m s(t - t_s - \tau_m) + v_m(t) \tag{2.5}$$

In DAS beamforming the microphone array is steered by delaying or advancing the recorded signals based on a steering direction θ. Based on a selected propagation model (like those discussed before), the array dimensions and a steering direction θ the TDOAs for each of the M microphones, relative to the reference microphone, are computed as $\hat{\tau}_m$ (where $\hat{\tau}_m = 0$ for the reference microphone). The steered recordings are then as follows:

$$x_{m,steered}(t) = a_m s(t - t_S - \tau_m + \hat{\tau}_m) + v_(t + \hat{\tau}_m) \tag{2.6}$$

Note that if the steering direction points towards the source the computed TDOAs and the TDOAs of the recordings are the same, i.e. $\hat{\tau}_m \approx \tau_m$. Therefore the signal part of Equation 2.6 becomes $s(t - t_s)$ which is the same for each microphone m. Then the DAS beamformer sums the steered recordings from all microphones together and normalises the output to the number of microphones M (Equation 2.7).

$$p(t) = \frac{1}{M} \sum_{m=1}^{M} x_{m,steered}(t) \tag{2.7}$$

If the steering direction is pointing towards the source then the signal part $s(t)$ becomes M times larger after summing than the noise parts. In order to use DAS for sound source localization, the beamformer grid-searches through different steering directions until it finds a peak in the energy of the summed recording $p(t)$.

Because DAS-beamforming searches in specific directions it is possible to localize multiple sources at once. However, the power peaks of a DAS search tend to be quite wide, which results in a low resolution of the search grid [21]. Valin et al. [67] tackle this by whitening the signal and processing the delay in the frequency domain, which results in narrower peaks. They then introduce spectral weights to aid in the detection of narrow-band signals and to increase the robustness against noise.

2.3.2. Machine learning-based localization

Machine learning excels when the relationship between the input and output is non-linear. This makes it an effective tool for sound source localization. Machine learning can be applied in the sound source localization process in different ways. First, machine learning can be used in the feature mapping stage of the classical localisation methods mentioned above. In such a case a network is trained on extracted features as inputs with known source locations, thereby removing the need for using explicit propagation models and the assumptions they bring. This is especially advantageous when the features are difficult to map, like when using spectral notches resulting from pinnae [44]. This use of learned mapping is also popular when using multiple features, like the binaural or spectral cues used by Saffari et al. [53] and Deleforge et al. [17]. Additionally, Rodemann et al. [51] and Youssef et al. [78] show this mapping's robustness against reverberations. He et al. [28] used a Deep Neural Network (DNN) to enable a humanoid robot to localize up to two simultaneous speakers. As an input feature, they used the generalized cross-correlation with phase transform (GCC-PHAT) [37]. Usually, this feature is used to determine the TDOAs between microphones. However, He et al. [28] used it as the direct input of their network. The robot that they work with does not fit within existing propagation models, therefor deep learning was used to learn the required mapping for localization.

Apart from the aforementioned methods of learning the mapping relationships, there are also machine learning methods which try to handle the entire localization process with the network. These so called end-to-end methods do not use preprocessed features, but the direct recordings from the microphones (though in some cases the recordings are first converted to the frequency domain). This allows the neural network to utilise an input which is much richer in features than a method which uses only a set of extracted features. Vera-Diaz et al. [70] present an end-to-end localization method that uses the raw recorded signals as inputs to the network. Firstly, they trained the DNN in simulation and then fine-tuned it with small real-world data set. Adavanne et al. [2] showed that their proposed DNN is capable of determining the bearings of up to three overlapping sounds from different sources. They mix both classification and regression to identify the source for each location. Chakrabarty and Habets [10] 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.

Another promising application of machine learning comes from the field of sensor fusion. In sensor fusion data from different sensors is combined to create a more complete picture than could be done with the sensors individually [22]. A good example of this comes from Chen et al. [12, 13], who work on the SoundSpaces project. They present a simulated robot capable of detecting and navigating towards a sound event using a DNN that processes both visual and audio signals.

Lastly, deep neural networks are very capable to process large amounts of data, as shown by Xu et al. [76], which uses an array consisting of 64 microphones in order to locate up to 25 simultaneous sources.

This thesis uses machine learning to localize other robots because of it's ability to handle non-linear relationships and because it's able to learn from challenging, environments (like those where there is no line of sight between the transmitter and the receiver), to be more applicable in real world situations.

3

System Overview

Below the specifics of the recording hardware, sound signals and the architecture of AudioLocNet will be described.

3.1. Hardware

In my work, I used Raspberry PI 4s to handle any of the audio transmission and recording tasks. These Raspberry Pis were equipped with a ReSpeaker 6-Mic Circular Array from Seeed [61]. This off-the-shelve microphone array contains six microphones equally distributed around a 10 cm diameter circle. The microphone array is connected to an accompanying PI shield which directly hooks into the Raspberry Pi. This shield also has a speaker output, to which a 6 Ω 2 W speaker [62] is connected. This speaker is attached to the microphone array using a 3D-printed bracket (self-designed). A fully equipped Raspberry Pi can be seen in Figure 3.1. A full hardware stack containing the Raspberry PI, the microphone array and the speaker is called a Chirpy. For each Chirpy, a line coming from the centre of the microphone array, through the centre point between the first and second microphones is defined as the 0° (or forward) direction where the angle increases when turning clockwise.

3.2. Sound signals

3.2.1. Chirp Signals

The AudioLocNet method focuses on inter-device localization, where the robots use audio for communication. Such devices can use different types of signals for their communication, such as narrowband signals like pure sine waves or wideband signals like white noise. However, narrowband signals struggle with constructive and destructive interference which is especially problematic for energy-based distance measurements. Wideband signals have fewer problems with this as the constructively interfering frequencies balance out the destructively interfering frequencies [41]. However, noise signals can be difficult to detect as they have, by definition, bad convolutional qualities. Chirp signals give the best of both worlds, as they are both wideband, but also show strong correlation properties, which makes them easy to detect.

Chirp signals (sometimes called sweep signals) are signals whose frequency continuously increases or decreases over time, wherein the way that the frequency changes (e.g. linear or exponential) differs between different chirp types. Figure 3.2a depicts the spectrogram of a linear *chirp 1* which starts at $f_s = 50\,\text{Hz}$ and ends at $f_e = 300\,\text{Hz}$ in $T = 5$ seconds. Figure 3.3 shows correlation plots of the correlations between the aforementioned *chirp 1* and, respectively, itself, another *chirp 2* ($f_s = 10$, $f_e = 250$, $T = 5$), a pure sine wave of 175 Hz, and white noise, where all signals have the same mean power. The spectrograms of the three new signals are depicted in Figures 3.2b to 3.2d. Note the high correlation peak when chirp 1 correlates with itself and how different it is when compared to the other signals. This peak makes the signal easy to detect.

3.2.2. Orthogonal Chirps

During my work on this thesis, I worked together with another student who was researching Aerial Acoustic Communication (AAC) [68], where robots communicate using audio signals. In order to facilitate multiple simultaneous transmissions, we based our signals on orthogonal chirp (Ochirps) signals as introduced by

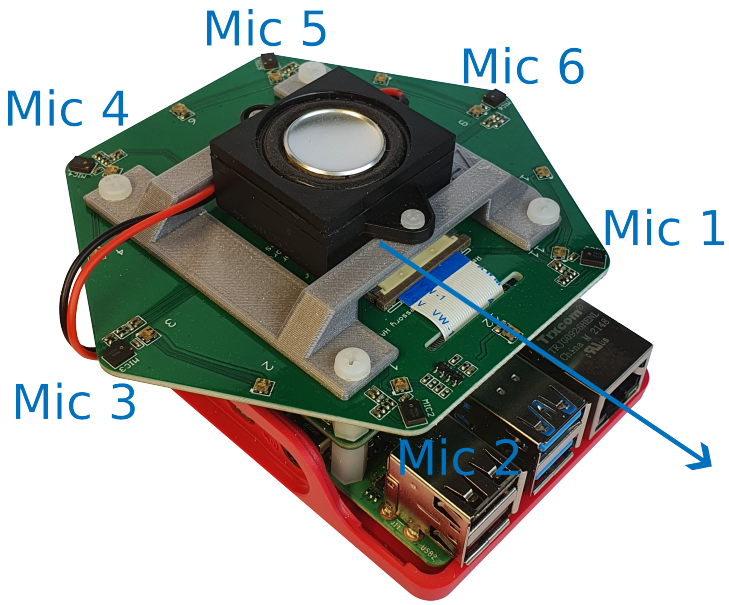
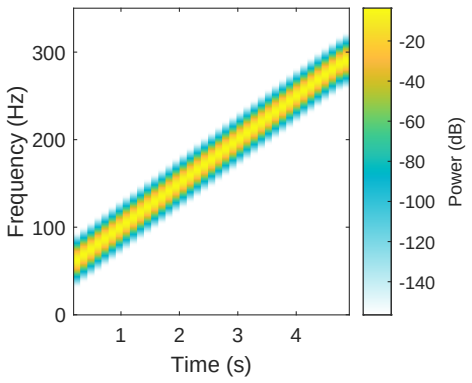
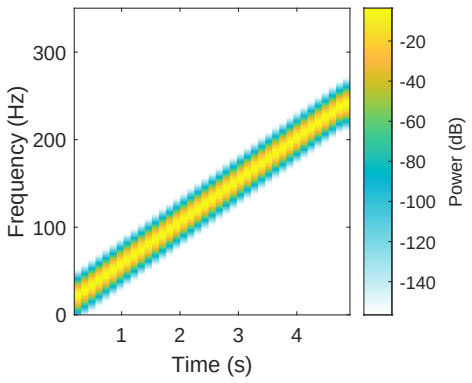


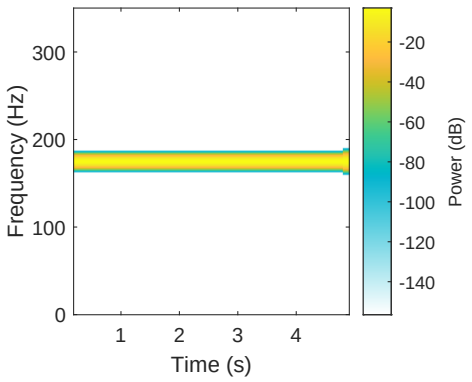
Figure 3.1: A Chirpy consisting of a Raspberry Pi 4 with the microphone array, accompanying shield and speaker stacked on top. Where the forward (or 0°) direction is denoted by the blue arrow



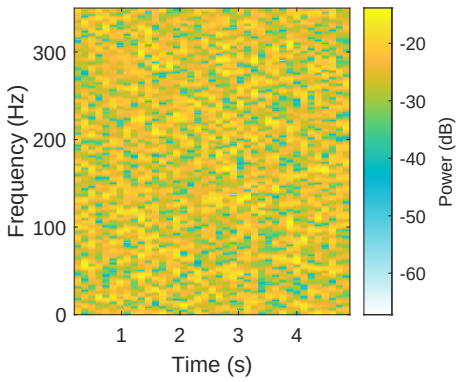
(a) Chirp 1, linearly increasing from 50 to 300 Hz.



(b) Chirp 2, linearly increasing from 10 to 250 Hz.



(c) Pure sine 175 Hz



(d) Gaussian white noise

Figure 3.2: Spectograms for four different signals

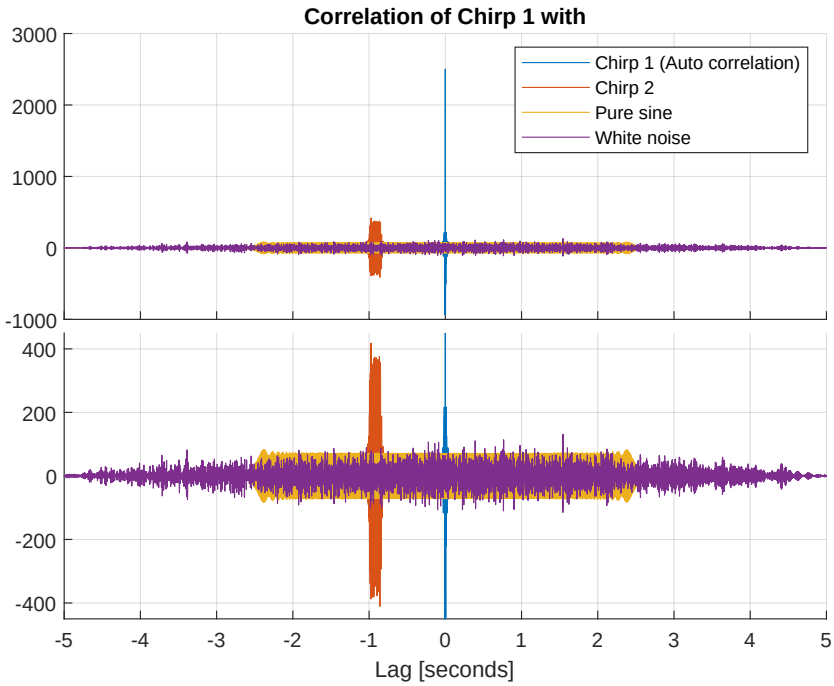
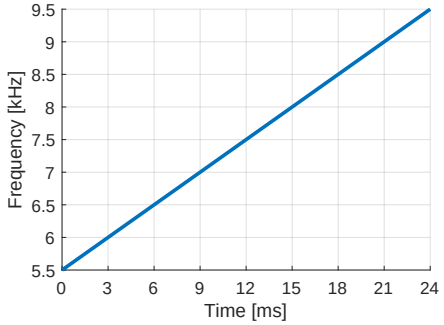


Figure 3.3: Auto and cross-correlations of chirp 1 with 4 signals, split into a full scale and a zoomed-in version. Note the high autocorrelation peak compared to the crosscorrelation peaks with the other signals.

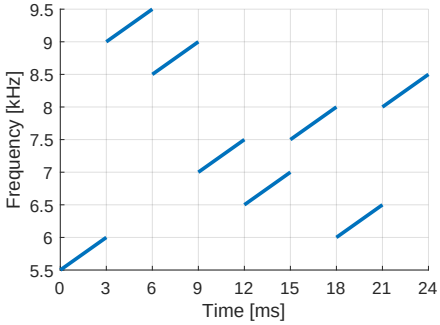
[34]. Ochirps allow for multiple simultaneous transmissions where each transmission uses the same time-frequency resources. This is achieved by using a technique similar to frequency hopping. The chirp is divided into M sub chirps, where each sub chirp comprises a chirp signal spanning $1/M$-th of the bandwidth of the full chirp in $1/M$-th of the total time, where none of the sub-chirps have overlapping frequencies. Sub chirps $\{1,2,\ldots,M\}$ are then reordered to create a signal that spans the entire bandwidth and time as the original chirp. This process is depicted in Figure 3.4 for $M = 8$. It is possible to make groupings of M sets of orders where none of the corresponding signals have overlapping frequencies at the same point in time (Figure 3.5). Ochirps are generated by selecting M such that the individual Ochirps have low crosscorrelation but relatively high autocorrelation peaks. This behaviour is depicted in Figure 3.6 for the set of Ochirps from Figure 3.5. This means that the different Ochirps can be detected, even if they are not neatly aligned (i.e. all starting at the same time).

AudioLocNet was trained using the same orthogonal chirps that were used as part of the AAC protocol and are depicted in Figure 3.5, meaning that the robots could locate each other while communicating.

We also added an envelope over each sub-chirp to reduce the amplitude of the signal at the endpoints of each sub-chirp. This prevents popping noises and distortions which can occur when the audio signal changes faster than the loudspeaker can handle. This can be observed in the time series plot of the signal for chirp 1 (Figure 3.7).

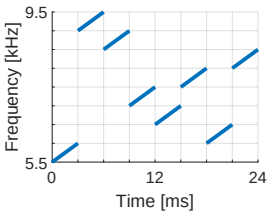


(a) Base chirp, linearly increasing from 5.5 to 9.5 kHz in 24 ms, divided into 8 sub chirps each with a bandwidth of 500 Hz and a duration of 3 ms

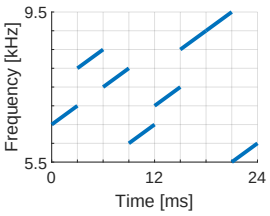


(b) Ochirp 1, constructed of the 8 sub chirps from (a).

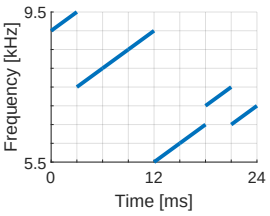
Figure 3.4: Process of generating a set or Ochirps with the frequency range of 5.5 to 9.5 kHz with a duration of 24 ms. The full chirp in (a) is divided into 8 parts. These parts are then reshuffled to form an Ochirp (b). Ochirp 1 from (b) is part of the set of Ochirps depicted in Figure 3.5.



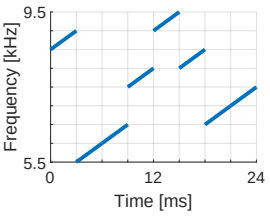
(a) Ochirp 1



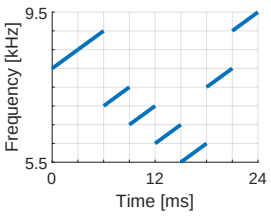
(b) Ochirp 2



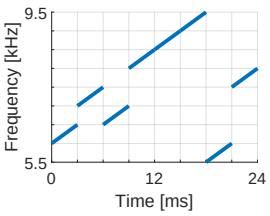
(c) Ochirp 3



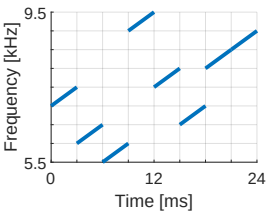
(d) Ochirp 4



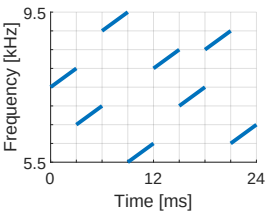
(e) Ochirp 5



(f) Ochirp 6

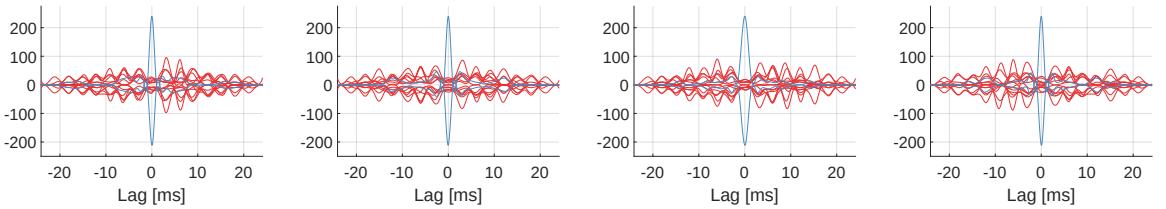


(g) Ochirp 7



(h) Ochirp 8

Figure 3.5: Set of 8 Ochirps generated from Figure 3.4

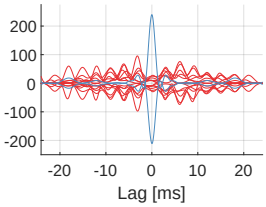


(a) Correlation envelopes with as base Ochirp 1 (3.5a)

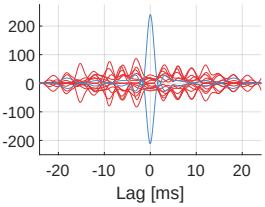
(b) Correlation envelopes with as base Ochirp 2 (3.5b)

(c) Correlation envelopes with as base Ochirp 3 (3.5c)

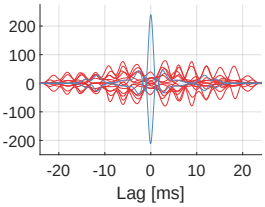
(d) Correlation envelopes with as base Ochirp 4 (3.5d)



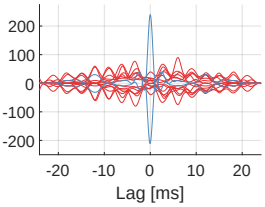
(e) Correlation envelopes with as base Ochirp 5 (3.5e)



(f) Correlation envelopes with as base Ochirp 6 (3.5f)



(g) Correlation envelopes with as base Ochirp 7 (3.5g)



(h) Correlation envelopes with as base Ochirp 8 (3.5h)

Figure 3.6: Correlation envelopes for the set of Ochirps from Figure 3.5. Where each Ochirp is correlated with itself (autocorrelation, in blue) and other Ochirps of the set (cross-correlation, in red). The cross-correlation graphs are superimposed onto each other to show that none of the cross-correlations gets close to the autocorrelation peak. Note that for all Ochirps, the autocorrelation peak is at least twice the highest cross-correlation

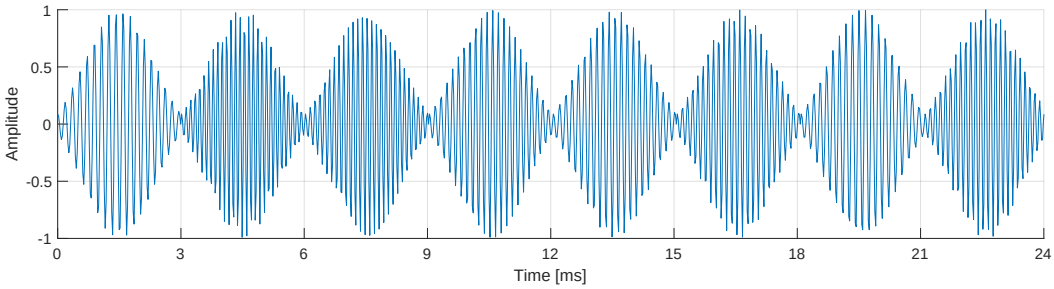


Figure 3.7: Time series for Ochirp 1

3.3. Localization

3.3.1. Location Grid

The localization of the Ochirp sources is done by the deep neural network AudioLocNet. AudioLocNet relates a 24 ms, 6 microphone recording (made with the microphone array from Figure 3.1) to one of 96 possible locations around the microphone array. These locations are depicted as the dots in Figure 3.8a where the microphone array is illustrated as the green hexagon in the centre. The grid locations are constructed out of 5 concentric circles (with the microphone array at the centre) with radii of 50 cm, 100 cm, 150 cm, 200 cm and 250 cm, where the two smallest circles are evenly divided into 12 tiles each, and the remaining circles are divided into 24 tiles per circle. For this grid the distances between locations range from 25.9 to 65.3 cm, with a mean distance 50.6 cm.

For improved accuracy, a finer grid was made by adding locations in between said 96 locations, as depicted in Figure 3.8b. These points inhabit the intermediate rings at 75 cm, 125 cm, 175 cm and 225 cm. The positions on the ring at 75 cm are spaced 30° apart, but at an offset of 15° from the forward direction. The other rings have locations spaced 15° apart and at an offset of 7.5°. This finer grid consists of 180 locations and reduces the maximal distance between two locations to 50 cm and makes the spread of neighbour distances more equal, as shown by the boxplots in Figure 3.9.

The parameters for the rings which construct the location grids are summarised in Table 3.1.

3.3.2. Deep Neural Network

The architecture of the final version of AudioLocNet is depicted in this subsection. During the research multiple different architectures were trained and evaluated. This version was selected as it has the best performance and uses a smaller network than the different networks that were trained.

AudioLocNet comprises an input layer, three convolutional layers, and a dense (or fully connected) output layer (Figure 3.10). The input of AudioLocNet consists of a 1060×6 array which, at a sampling frequency of 44.1 kHz, portrays a 24 ms recording window captured by each microphone, matching the length of a single Ochirp. This length was chosen with the ACC system from [68] in mind. The ACC system crops a recorded Ochirp of which it wants to know the location and sends this recording to AudioLocNet to perform sound source localization.

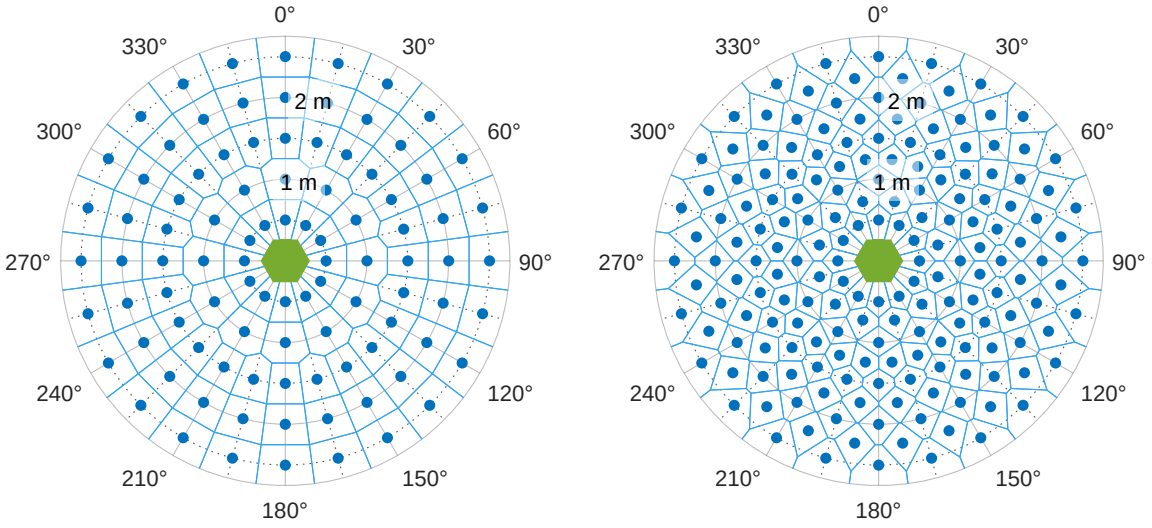
The output layer has, respectively, 96 or 180 nodes depending on whether the coarse or the fine grid is used. Each output node uniquely relates to one of the, respectively, 96 or 180 possible locations around the microphone array (Figure 3.8). Therefore AudioLocNet is a classification network which determines both the DOA and the distance of a sound source as one of 96 or 180 classes.

In the input layer, the 1060×6 input array is toroidally padded around the time axis 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. In Figure 3.10, the padding is illustrated by the copies of the channels at the input. Without this location-aware padding, the network was more likely to overfit. The padded array is 11 channels by 1060 time samples in size.

The first convolutional layer has a 50×1 kernel. By going over the individual channels, this layer helps with finding the Ochirps. The second convolutional layer has a 20×6 kernel. The 20×6 kernel shifts over all six microphones, where the 20 time steps per kernel ensure that a signal arriving at a first microphone during

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

Table 3.1: Parameters for the rings of locations which form the location grids. Where the distance denotes the radius of the ring (which is centred at the microphone array), Angle denotes the angular distance between two subsequent locations on the same ring and Offset denotes the angular offset from the 0° direction. Note that the even rings (denoted with "(f)") are only used for the fine grid



(a) Coarse location grid, 96 locations

(b) Fine location grid, 180 locations

Figure 3.8: Two different location grids, with the Chirpy which records the sound at each centre, coarse grid with 96 locations (a) and a fine grid with 180 locations (b).

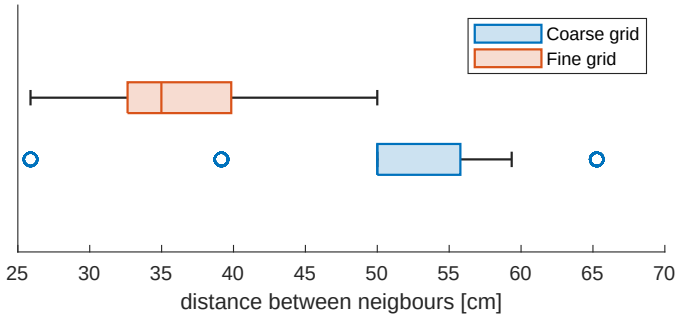


Figure 3.9: Boxplots depicting the different distances between neighbouring cells for the course and fine location grids from Figures 3.8a and 3.8b respectively.

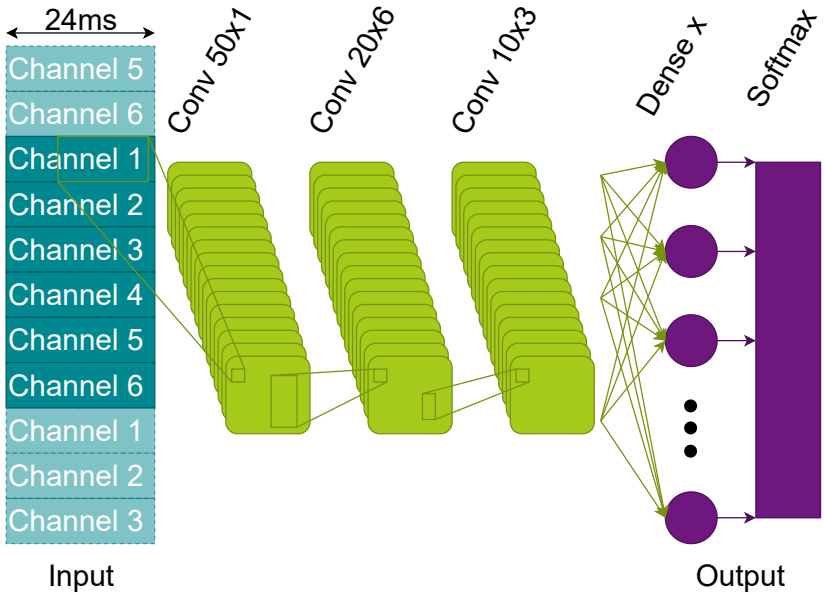


Figure 3.10: AudioLocNet architecture, where the size of the final layer ("Dense x") is different depending on the number of classification locations. For the coarse grid of Figure 3.8a x = 96, and for the fine grid of Figure 3.8b x = 180.

the first time step will be received by a microphone farthest away from said first microphone before the end of the kernel.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. The last convolutional layer comprises a 10×3 kernel. All convolutional layers consist of 64 filters and use rectified linear unit (ReLU) activation functions.

For the output layer, a softmax activation function converts the values of each node to a probability distribution over all output nodes of the network. The node with the highest value is then taken as the predicted location by the network.

4

AudioLocNet

This chapter describes the process of producing a trained version of AudioLocNet. Starting from the data collection, going through the training process and the validation of the network.

4.1. Data collection

In order to train the network to determine the sound location in different indoor environments a set of training data had to be collected. In order to cover different auditory landscapes, three different environments were chosen to record sounds from:

- Free space environment;
- Reverberant environment, illustrated in Figure 4.1; and
- Non-line-of-sight (NLOS) environment, illustrated in Figure 4.2.

The recordings were made using two Chirpies: one listener Chirpy, which would be placed on a fixed location; and one talking Chirpy, which would be placed on locations surrounding the listener, where the locations surrounding the listener match the locations of the localization grids from Figure 3.8. For the free space environment, the locations match the grid one-to-one. However, due to the presence of walls in the reverberant and NLOS environments, partial patterns were used for those locations. After recording signals from each location of such a partial pattern, the listener was rotated and new recordings were gathered. This way recordings from all the locations of the full localization grid were gathered. The maps depicting these partial patter for the talker locations for these environments are depicted in Figure 4.1 (reverberant environment) and Figure 4.2 (NLOS environment).

The free space environment has no obstacles between the talker and listener and there are no walls near the listener to cause echoes. The microphones are 3 cm above the floor, this creates a second path for the sound to take while travelling from the talker to the listener. This is to be expected for small robots moving near the floor and was therefore kept constant between all environments.

In the reverberant environment (Figure 4.1) the listener was placed in the corner of two walls, with the backside of the Chirpy facing a wall at 15 cm and the wall on the right side being at a distance of 40 cm, where the distances are measured from the centre of the microphone array.

For the NLOS environment (Figure 4.2), the listener is placed 15 cm from its facing wall and 26 cm along the wall, from the corner of said wall. From the perspective of the listener, the corner behind which the listener locations are located is 30 cm away at an angle of 30°. With this listener location and the talker locations from Figure 4.2, the geodesic distance and the initial direction towards the talker match locations on the location grid of Figure 3.8.

For each recording, the talking Chirpy is placed at one of the locations of one of the environments. At this location, the talker transmits eight sequences of Ochirps to be recorded by the listener. Each sequence consists of 200 copies of one of the eight Ochirps of Figure 3.5.

The talking and listening were synchronised using MQTT, where a single message would trigger both the talker and listener to transmit and record a sequence of Ochirps.

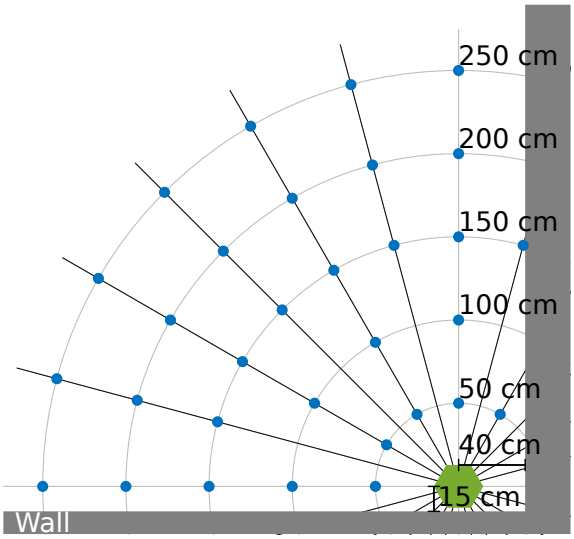


Figure 4.1: Coarse location grid for the reverberant environment

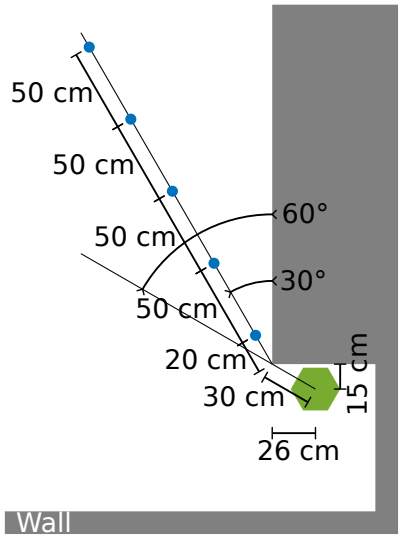


Figure 4.2: Coarse location grid used for the NLOS environment

Table 4.1: AudioLocNet data set, where each recording is from a single source location and contains 200 Ochirps. The entire dataset consists of

Set	Environment	Ochirp length	Grid	Recordings
1	Free space	24 ms	Fine	1,632
2	Free space	48 ms	Fine	1,632
3	Free space	24 ms	Off grid	2,312
4	Reverberant	24 ms	Fine	1,568
5	Reverberant	48 ms	Fine	1,568
6	Reverberant	24 ms	Off grid	2,312
7	NLOS	24 ms	Fine	1,440
8	NLOS	48 ms	Fine	1,440
9	NLOS	24 ms	Off grid	2,312

Data set repository

In addition to the data set mentioned above, two more sets of recordings were collected. One set using the fine grid with longer Ochirps of 48 ms in length, these were used to train a previous version of AudioLocNet. In order to support future work with these microphones a third set of recordings with 24 ms Ochirps was recorded. Unlike the first set, this set uses randomly determined locations for the talker. These locations are depicted in Figure 4.3.

In order to facilitate further research, these labeled recordings are made available on the 4TU.ResearchData repository[1]. In the repository the audio files are grouped in 9 zip files, with per environment one for the 24 ms recordings, one for the 48 ms recordings (both are on the fine grid), and one for the 24 ms off grid recordings. The numbers of recordings per set are depicted in Table 4.1. Recordings are made for each of the Ochirp configurations of Figure 3.5, For the 48 ms Ochirps, the time axis of this plot is stretched to reach 48 ms. Each recording comprises 200 Ochirps, resulting in a dataset consisting of 3.2 milion samples.

[1] `https://data.4tu.nl/` Full URL to be provided on publication

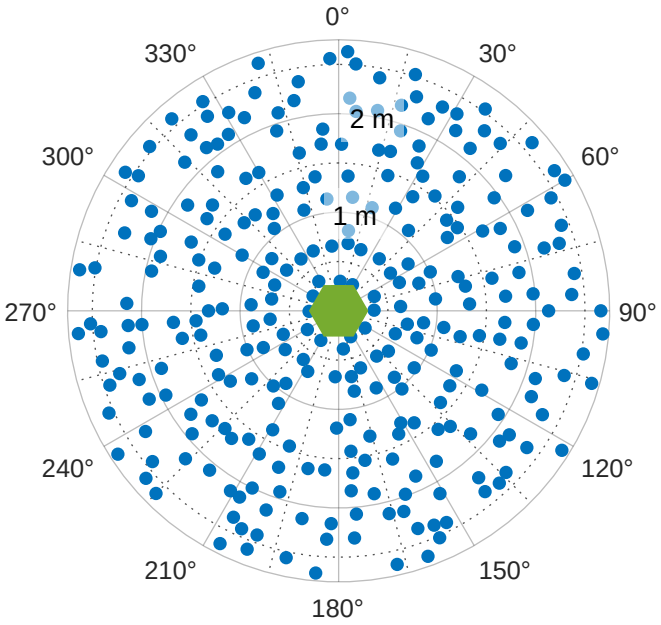


Figure 4.3: Speaker locations for the off grid recordings.

4.2. Training

4.2.1. Split Data Sets

AudioLocNet was trained using a data set sampled from the aforementioned samples. This data set comprises 920,000 samples with orthogonal chirps with a duration of 24 ms, sampled randomly from the different locations. The total sample size was reduced to speed up the training process. This sampled set was split into a training, validation, and testing set, each containing 70 %, 15 % and 15 % of the samples from the sampled data set, respectively. Each of these data sets has different a purpose.

The training set contains the recordings which are actually used by the training algorithm to tweak the network parameters. The training algorithm loads the training data into the network and computes, per batch of training data, how well the network is predicting the source locations. This is done based on a loss function, which compares the true source location of each sample in a batch with the corresponding predicted location. A loss function is a positive function that gets closer to zero as the network improves. Based on the loss, the training algorithm changes the weights of the connections between nodes in order to reduce the loss function further.

The validation set is used to check the progress of the network on a set of data it has not seen before. After each epoch[2] the training is paused and the validation data set is run through the current network. The performance of this validation set is then compared to the performance of the training set. If both reach a similar performance, then the training process is generalising well, but if the validation performance flattens below the training performance, then the network is overfitting instead of generalising.

Figure 4.4 depicts the training process on a network that overfits. Where in the beginning the network is learning and generalising (as seen by the validation and training performances being similar) but after 4 epochs the validation results split apart from the training results (where the validation results level out, but the training is still improving). This shows that the network is learning how to recognise the specific training samples, instead of generalising in order to handle new samples. These samples are also used as a way to determine when to stop training, when the validation results do not improve for 6 validation steps in a row, the network is deemed done with its training. At this point, the network with the best validation results is picked as the trained network. The training algorithm never sees this validation data set, nor the results of the validation step. This way it is prevented that the trainer trains on the validation data set.

Lastly, the testing data set, this data set is held separate from the training process. Only after the trained

[2] An epoch is one complete pass of the training data set

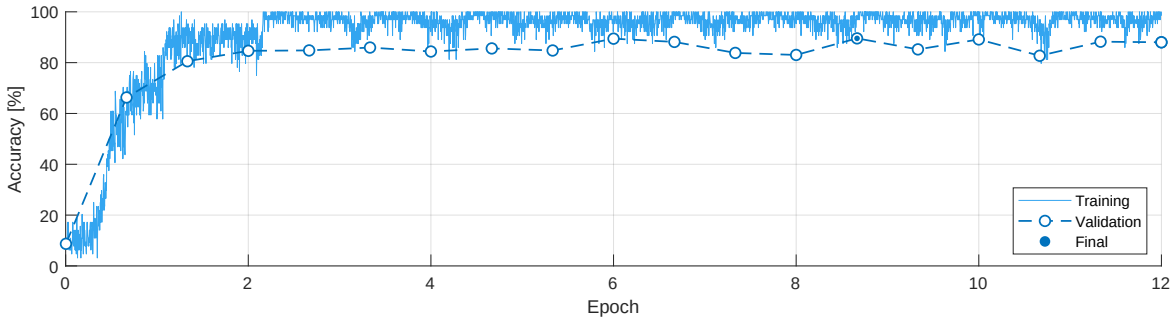


Figure 4.4: Training progress of a network that overfits. Note that from epoch 2 onward the validation accuracy starts to flatten out whilst the training accuracy is improving

Table 4.2: 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
Batch size	256

network shows good results for both its training and validation data, will it encounter the testing data. This data, which has not been fed to the network during training of the network, is then used to determine the performance of the network on unknown data. All the claims on the performance of AudioLocNet are done based on the results from this testing set.

4.2.2. Single Source Training

The network was trained using the Adam training algorithm [35], with cross-entropy loss as its loss function. The final parameters can be found in the Table 4.2, which generally match the suggestions of the original paper [35]. 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 [8].

The process of training went as follows: First, a new network architecture was trained on a reduced training set comprising 30% of the samples from the normal training set. During this step the dropout layer and L2 regularization were disabled. The goal of this training process is to see if the network is capable of overfitting on a small data set. If this is not the case then it could be that the network is not complex enough to capture all the features.

Then the network is trained on the full training set with the periodic validation checks. Based on the results the training L2 regularization and dropout parameters are tweaked to train a network that does not overfit and gets a good localization accuracy. If a network reaches an insufficient validation accuracy while not overfitting, then the network architecture is altered.

Figure 4.5 depicts the training process both in terms of the classification accuracy in the top graph and the loss (which is used by the trainer to improve the network) in the bottom graph, where the accuracy is the mean of the number of correctly identified samples over the total number of samples in a 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.

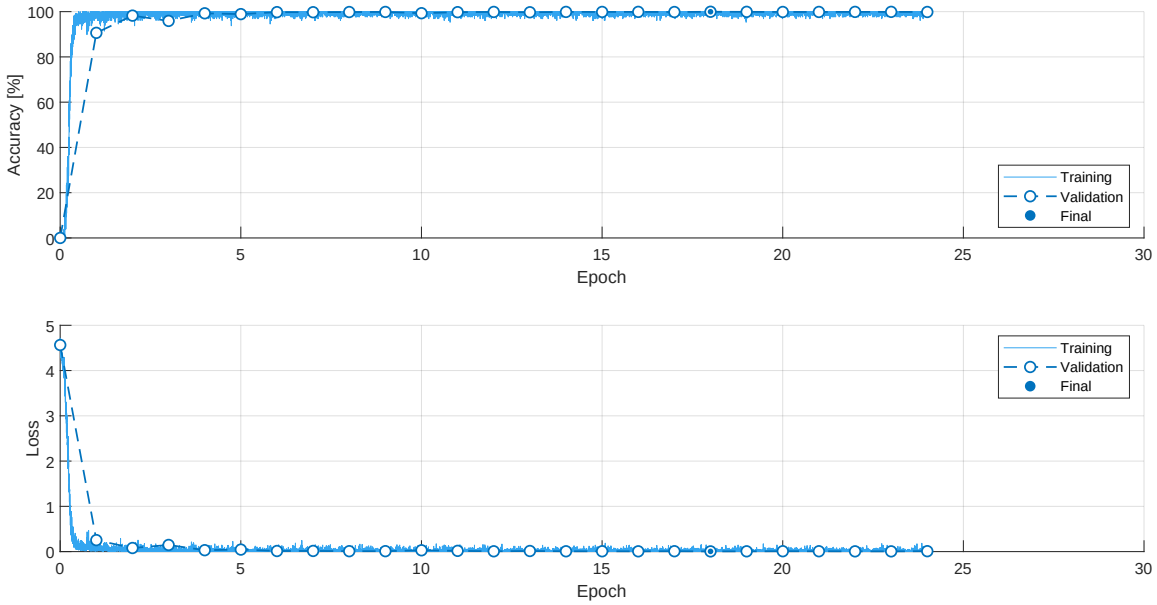


Figure 4.5: Plots showing the training process in terms of both the accuracy and loss for the coarse grid

4.2.3. Toroidal padding

Figure 4.6 depicts the training process for when the training for the coarse network was repeated but while omitting the toroidal padding. I.e. the architecture and data sets are the same. When compared to Figure 4.5, Figure 4.6 shows that although the performance on the training data is not influenced much, the performance of the validation data is dramatically worsened. Whereas the network with padded inputs reached an accuracy above 90 % after the first epoch, the non-padded network only reached this after 9 epochs (after which it first dropped down to 80 % before improving further).

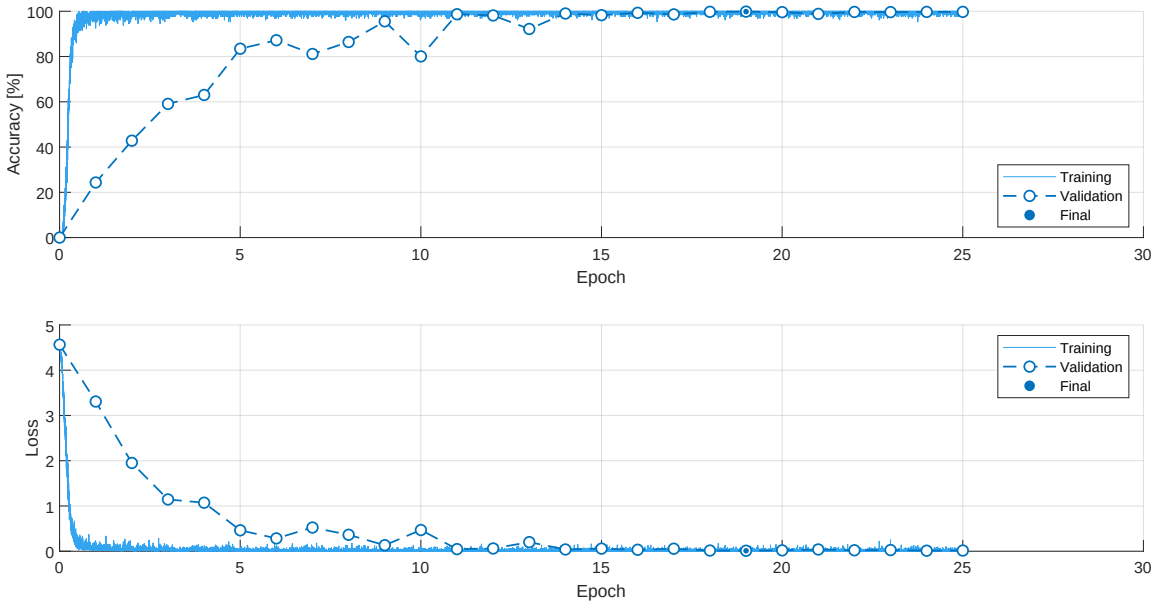


Figure 4.6: Plots showing the training process in terms of both the accuracy and loss for the coarse grid without padding

5
Evaluation

In order to determine how well the trained AudioLocNet functions, it was tested using the aforementioned testing set (subsection 4.2.1) of recordings that had never been fed to the network before. This chapter goes over the performances of the different trained networks.

5.1. Hop Error

Usually, the performance of a classification network like AudioLocNet is measured using metrics like the accuracy and the F1-score [24]. 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. This means that such metrics would penalize being one class next to the correct class the same as when the correct class and the wrong prediction are on opposite ends of the location grid. On the other hand, methods like the mean absolute error between the predicted and true locations are unable fully deal with the coarseness of the possible outputs. If the network outputs the right class, then the error would be zero, resulting in a lot of zero-valued errors, which pulls the mean down towards values smaller than the distance between two classes. It should be noted, however, that other works, like [28], do use the angular equivalent of the mean absolute error (i.e. the mean angle error) even though their networks only localize in classes 5° in width. Therefore, I introduce the hop error to more closely reflect the relations between (mis)classifications and the true classes.

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 (Figure 5.1). The hop error of a correct prediction is zero. Another reason why this metric works better than the distance error is that the distances between adjacent classes are not constant due to the circular nature of the location grid. The length equivalences of different numbers of hops are, for the different grids from Figure 3.8, depicted in Figure 5.2.

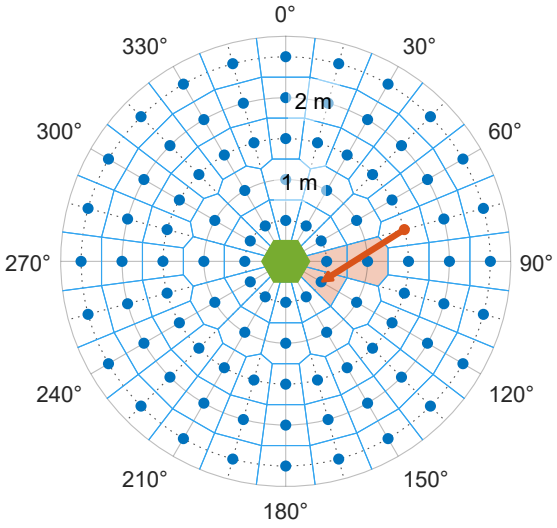


Figure 5.1: An example of a hop error of 3 hops.

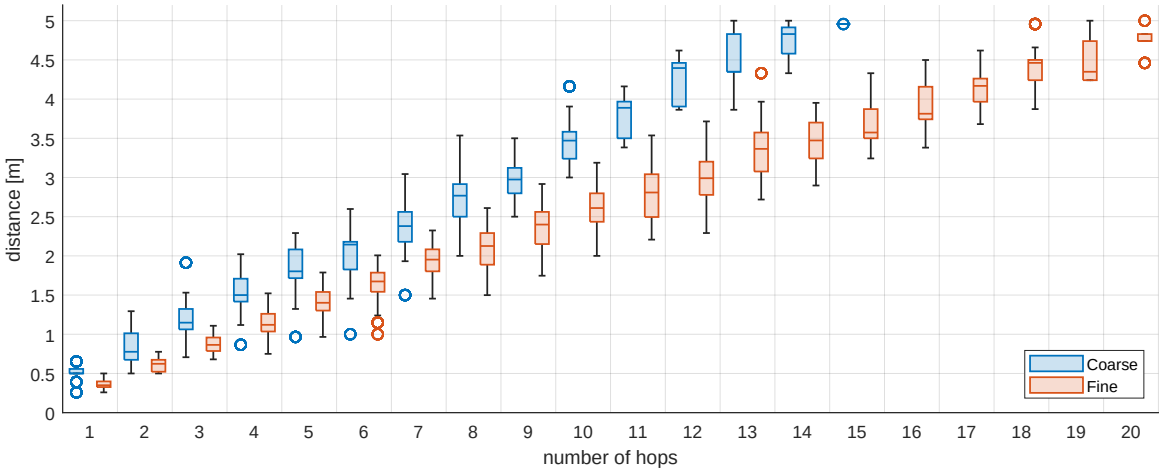


Figure 5.2: Distributions of absolute errors for different hop errors in the coarse and fine grids

5.2. Coarse Grid

For the course grid, AudioLocNet reached a classification accuracy of 99.96 %. With not much difference between the environments, which reach accuracies of 99.96 %, 99.99 % and 99.92 % for the free space, reverberant and NLOS environments respectively. For this localization grid the mean hop error is 0.0029 hops or (as the mean distance error) 1.18 mm

Firstly, Figure 5.3 shows how well AudioLocNet performs in the different environments in terms of the cumulative distribution functions (CDF) of the hop errors. Interestingly the reverberant environment is the best performing.The NLOS environment has the worst performance, which was expected as the amount of energy in each recording would be the lowest and closer to that of the first incoming reflected signals. Though it should be mentioned that at these CDF values, Figure 5.3 shows individual mispredictions. Hence there is not much difference between the different environments.

Figure 5.4 depicts the accuracy and mean hop error for different distances from the microphone array. This shows that the accuracy of the prediction decreases for further distances and that this decrease in accuracy starts sooner for the non-line-of-sight environment.

In order to determine if AudioLocNet had certain preferences or dislikes for specific locations, the mean hop errors for sources from each location were investigated. Depicted in the form of the heat maps in Figure 5.5. These figures show that there are no areas where AudioLocNet is constantly making false predictions.

5.3. Fine Grid

For the version of AudioLocNet which was trained on the fine grid, the same analysis was performed. For this network the total accuracy is 99.89 % with, again, not much difference between the different environments (99.88 %, 99.92 % and 99.86 % for the free space, reverberant and NLOS respectively). The mean hop error is 0.0079 hops equating to a mean distance error of 2.05 mm.

The CDF plots form Figure 5.6 show that locations from the free space and reverberant environments are predicted equally well. Again the NLOS environment performs slightly worse.

The accuracy over distance plots from Figure 5.7 show that the fine network seems to have less trouble with increasing distances. However, as shown by the mean hop error, the size of the errors does increase. Mostly for the NLOS environment. This increase in the mean hop error for the NLOS environment can mostly be attributed to a single location at the 0°,250 cm mark as seen in Figure 5.8c. This comes however from three larger misclassifications with hop errors of 20, 18 and 18 hops out of a total of 74 predictions (where two mispredictions had hop errors of 2 hops and the remainder was correctly predicted).

The heat maps of Figure 5.8 show that the network does not have dead spots where it is consistently mispredicting the source location.

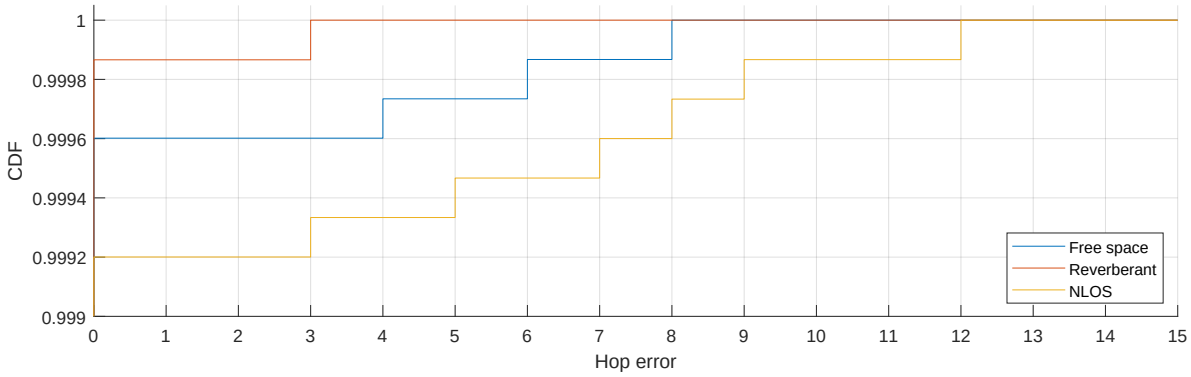


Figure 5.3: Cumulative distribution function for the hop error for different environments for the coarse grid

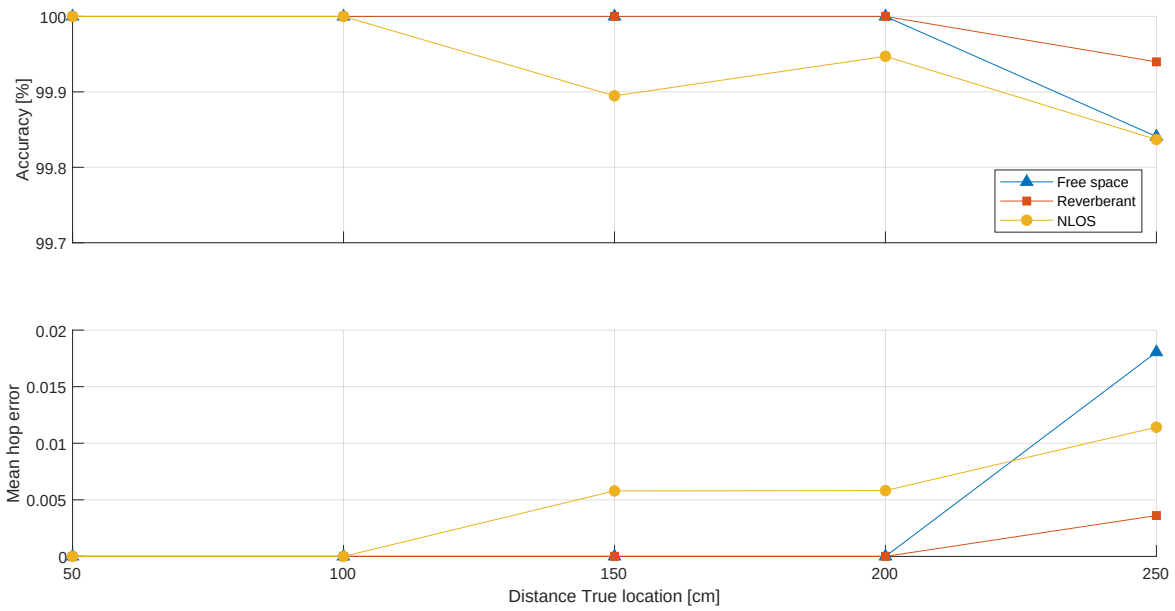
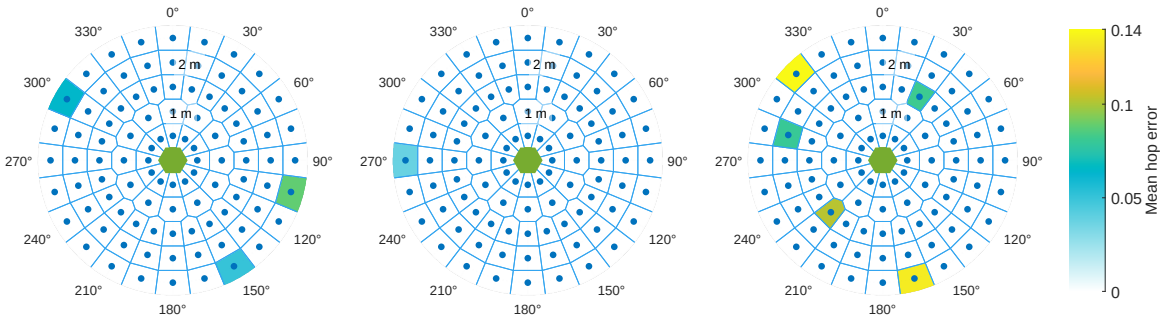


Figure 5.4: Accuracy and mean hop error for the coarse grid for different distances in different environments



(a) Free space (b) Reverberant (c) NLOS

Figure 5.5: Heat maps of the mean hop error for different locations and different environments for the coarse grid.

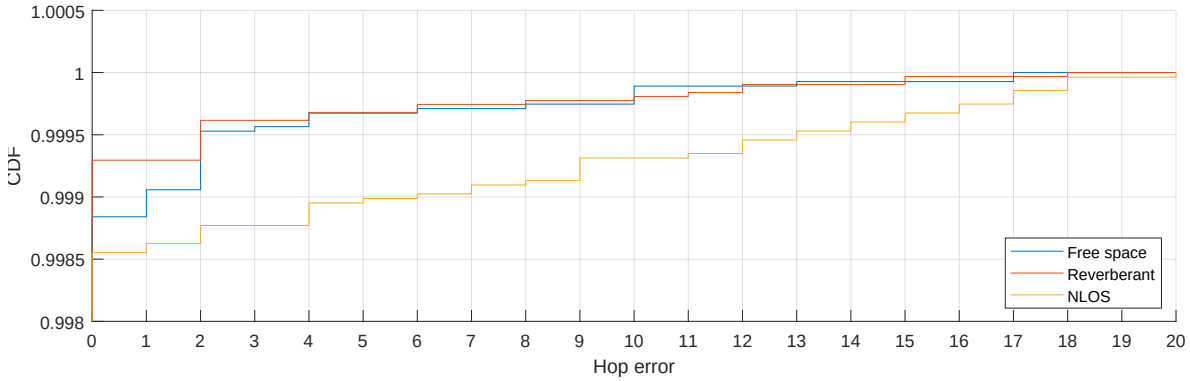


Figure 5.6: Cumulative distribution function for the hop error for different environments for the fine grid

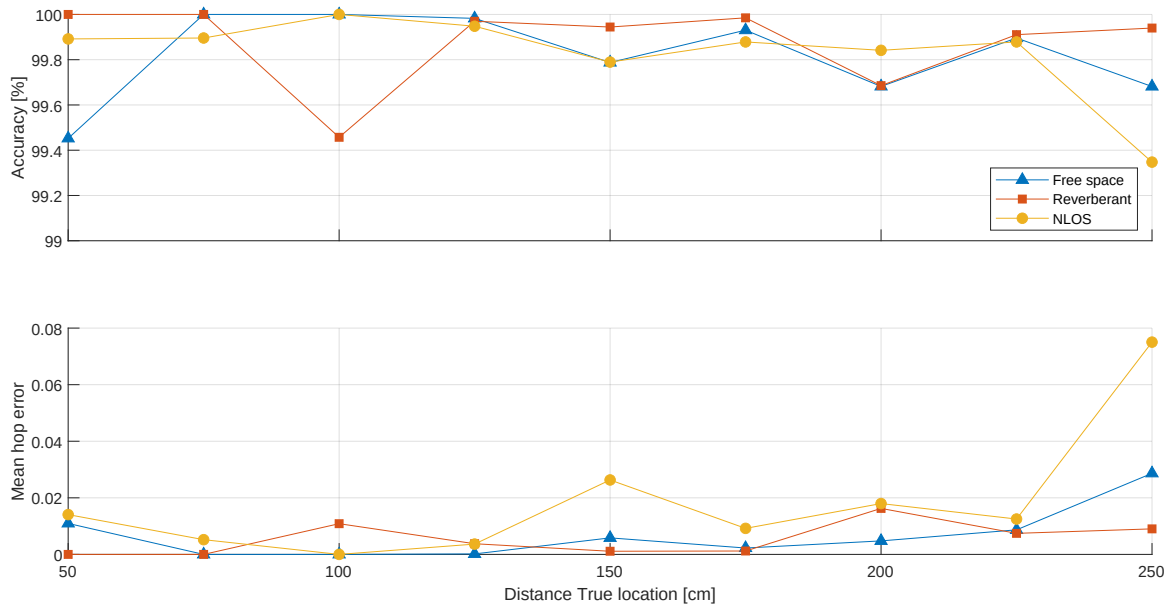
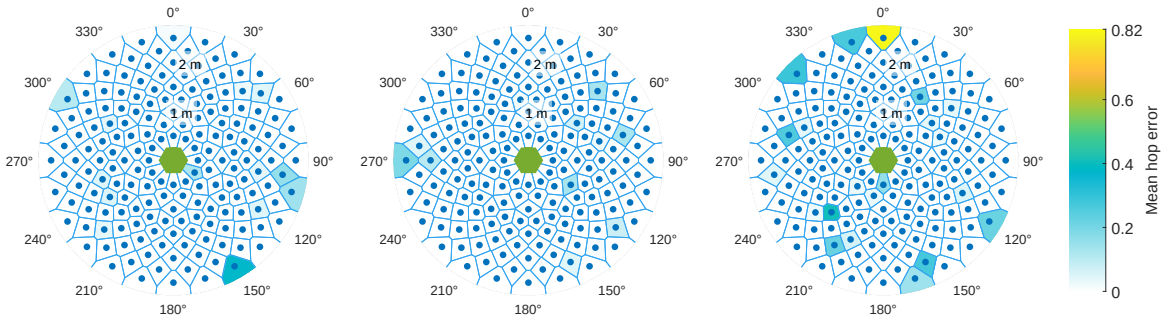


Figure 5.7: Accuracy and mean hop error for the fine grid for different distances in different environments



(a) Free space (b) Reverberant (c) NLOS

Figure 5.8: Heat maps of the mean hop error for different locations and different environments for the fine grid.

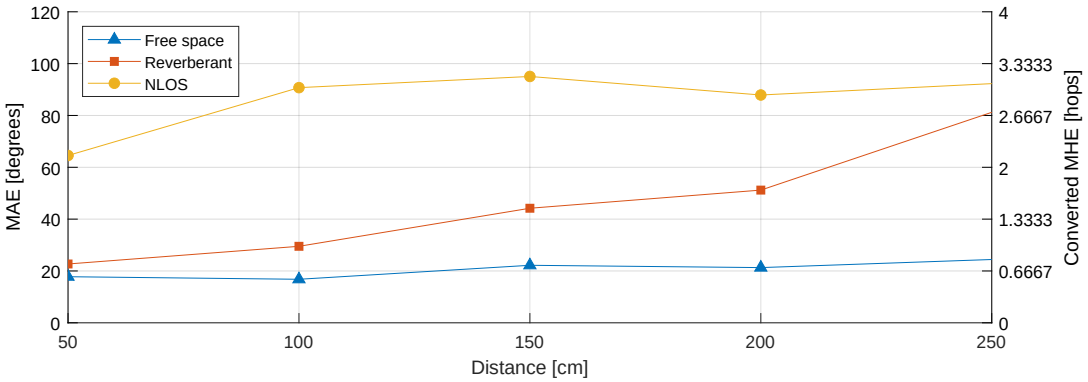


Figure 5.9: Performance of the Classic DOA method from [33] using the testing set. With the mean angle error (MAE) on the left axis and the equivalent mean hop error (MHE), where the conversion between angle to hops is done by setting 30° equal to one hop.

5.4. Comparison with Classic localization

In my work on [68], we compared the performance of AudioLocNet to that of a classical audio localization method, inspired by Karbasi and Sugiyama [33]. This method was made to be used with circular microphone arrays and uses the TDOA of the Ochirp at the different microphones to determine the DOA. In order to compare both methods, the classical method was fed the same testing set of recordings as AudioLocNet. Since the classical method only delivers the DOA and not the distance, the comparison with AudioLocNet is made using presumed hops of 30°. This is advantageous for the Classical method as this results in the largest hop steps. The performance of the classical method is depicted in Figure 5.9, with the mean angle error (MAE) on the left axis and the equivalent mean hop error (MHE) on the right.

By comparing Figure 5.9 with Figure 5.4 the performance improvements that AudioLocNet delivers become clear. AudioLocNet consistently delivers a better localization accuracy and shows a lower decrease in accuracy as the distance increases. This shows how a network trained to utilize the characteristics of a certain microphone array can perform better than a general solution.

6

Conclusions and Future Work

6.1. Conclusion

This work focused on developing a DNN-based method of audio source localization to allow small robots to locate each other. To this end, AudioLocNet was developed. AudioLocNet is a convolutional classification network which is capable of locating the sources of communication chirps which are 24 ms in length and are up to 250 cm away. As a classification network, AudioLocNet returns the predicted source location as one point (comprising distance and direction) on a localization grid (Figure 3.8). These chirps are recorded using a small (10 cm diameter) six-microphone array. The input of AudioLocNet comprises these 6 channel recordings which are toroidally padded to account for the circular nature of the microphone array. The remainder of AudioLocNet consists of three convolutional layers and a final fully connected layer whose size depends on the resolution of the localization grid.

AudioLocNet was trained for two different localization grids, a coarse grid of 96 locations and a finer grid with 180 locations, while only changing the number of nodes of the output layer, leaving the rest of the architecture unchanged. This shows that the base of AudioLocNet can easily be adapted for different localization grids. Since there are no assumptions made on the shape of the localization grid, it is likely that AudioLocNet could also be trained to handle non-uniform localization grids. This would allow a robot designer to gain a higher localization resolution in important areas (like in front of the robot) while reducing the resolution for areas which are of low importance, possibly reducing the processing power required. Further work would be required to determine the impact of such non-uniform grids.

AudioLocNet reaches high accuracies for different indoor environments including environments without a line of sight between the source and the microphone array. For each environment and grid, the accuracy of the predictions is above 99.85 %. Additionally AudioLocNet shows a large improvement a classical method designed to be used on the same type of microphone array. AudioLocNet shows that by training a network for a specific microphone array, the small size of the microphone array is not a problem in locating sources in both the DoA and distance. Because sound travels around obstacles rather than trough them sound can be used as a method of finding the locations of other robots along a traversable path. This gives more information to the robots than learning the locations of other robots through obstacles (as one would get when using RF based localization), as the robot doesn't require a map of the environment to reach another robot. It can simply follow the sound.

Another contribution of this thesis is the introduction of toroidal padding. In toroidal padding (or cylindrical padding when only done around one axis) the input matrix is wrapped around its axes to ensure that data which is spatially close together is not separated due to the borders of the matrix. When using toroidal padding, the training and validation performances stayed closer together, indicating that the network is not overfitting. For AudioLocNet this was used to inform the network that microphones 1 and 6 are spatially next to each other. Toroidal padding can also be useful in other fields where the data is best represented cylindrically and/or is periodic.

6.2. Future Work

The main next step for AudioLocNet is to have it run on the actual, moving, robots and to determine their performance related to the required resources in terms of (among others) time and computing power.

Another improvement of AudioLocNet can be made in the direction of multiple source detection, where the network is able to output multiple grid locations where it predicts a source to be present. The use of Ochirps already allows for simultaneous communications. Possible multi-localization methods could include preprocessing steps where the non-interesting Ochirps are first filtered out, and each Ochirp is then located individually. But it would be more interesting to see if a future version of AudioLocNet can locate the different sources simultaneously.

Bibliography

[1] Jasmine. URL `http://www.swarmrobot.org/index.html`.

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

[3] Sylvain Argentieri, Patrick Danes, and Philippe Soueres. Modal analysis based beamforming for nearfield or farfield speaker localization in robotics. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 866–871. IEEE, 2006.

[4] Farshad Arvin, Khairulmizam Samsudin, Abdul Rahman Ramli, et al. Development of a miniature robot for swarm robotic application. *International Journal of Computer and Electrical Engineering*, 1(4):436–442, 2009.

[5] Farshad Arvin, John Murray, Chun Zhang, and Shigang Yue. Colias: An autonomous micro robot for swarm robotic applications. *International Journal of Advanced Robotic Systems*, 11(7):113, 2014.

[6] Meysam Basiri, Felix Schill, Dario Floreano, and Pedro U Lima. Audio-based localization for swarms of micro air vehicles. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 4729–4734. IEEE, 2014.

[7] Levent Bayindir and Erol Şahin. A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering & Computer Sciences*, 15(2):115–147, 2007.

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

[9] Scott Camazine, Jean-Louis Deneubourg, Nigel R Franks, James Sneyd, Guy Theraula, and Eric Bonabeau. *Self-organization in biological systems*. Princeton university press, 2020.

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

[11] Arthur Charléty, Eric Larose, Mathieu Le Breton, Laurent Baillet, and Agnès Helmstetter. 2d phase-based rfid localization for on-site landslide monitoring. 2022.

[12] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *ECCV*, 2020.

[13] Changan Chen, Ziad Al-Halah, and Kristen Grauman. Semantic audio-visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15516–15525, 2021.

[14] Joe C Chen, Kung Yao, and Ralph E Hudson. Acoustic source localization and beamforming: theory and practice. *EURASIP journal on advances in signal processing*, 2003(4):1–12, 2003.

[15] Ahmad Reza Cheraghi, Sahdia Shahzad, and Kalman Graffi. Past, present, and future of swarm robotics. In *Proceedings of SAI Intelligent Systems Conference*, pages 190–233. Springer, 2021.

[16] Mario Coppola, Kimberly McGuire, KY Scheper, and GC de Croon. On-board bluetooth-based relative localization for collision avoidance in micro air vehicle swarms. *arXiv preprint arXiv*, 1609, 2016.

[17] Antoine Deleforge, Florence Forbes, and Radu Horaud. Acoustic space learning for sound-source separation and localization on binaural manifolds. *International journal of neural systems*, 25(01):1440003, 2015.

[18] Neel Dhanaraj, Nathan Hewitt, Casey Edmonds-Estes, Rachel Jarman, Jeongwoo Seo, Henry Gunner, Alexandra Hatfield, Tucker Johnson, Lunet Yifru, Julietta Maffeo, et al. Adaptable platform for interactive swarm robotics (apis): a human-swarm interaction research testbed. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 720–726. IEEE, 2019.

[19] Jacek P. Dmochowski, Jacob Benesty, and SofiÈne Affes. A generalized steered response power method for computationally viable source localization. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2510–2526, 2007. doi: 10.1109/TASL.2007.906694.

[20] Marco Dorigo. Swarm-bot: An experiment in swarm robotics. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 192–200. IEEE, 2005.

[21] Ramani Duraiswami, Dmitry Zotkin, and Larry S Davis. Active speech source localization by a dual coarse-to-fine search. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 5, pages 3309–3312. IEEE, 2001.

[22] Wilfried Elmenreich. An introduction to sensor fusion. *Vienna University of Technology, Austria*, 502: 1–28, 2002.

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

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

[25] Alvaro Gutiérrez, Alexandre Campo, Marco Dorigo, Daniel Amor, Luis Magdalena, and Félix Monasterio-Huelin. An open localization and local communication embodied sensor. *Sensors*, 8(11):7545–7563, 2008.

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

[27] Homayoun Hashemi. The indoor radio propagation channel. *Proceedings of the IEEE*, 81(7):943–968, 1993.

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

[29] Charlotte K Hemelrijk and Hanno Hildenbrandt. Schools of fish and flocks of birds: their shape and internal structure by self-organization. *Interface focus*, 2(6):726–737, 2012.

[30] Alfred O Hero. Radio transmission, 1998. URL `https://www.britannica.com/topic/telecommunications-media/Radio-transmission`.

[31] Junyan Hu, Parijat Bhowmick, Inmo Jang, Farshad Arvin, and Alexander Lanzon. A decentralized cluster formation containment framework for multirobot systems. *IEEE Transactions on Robotics*, 37(6):1936–1955, 2021.

[32] Sungmok Hwang, Ki-Hoon Shin, and Youngjin Park. Artificial ear for robots. In *SENSORS, 2006 IEEE*, pages 1460–1463, 2006. doi: 10.1109/ICSENS.2007.355909.

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

[34] Mohammad Omar Khyam, Md Noor-A-Rahim, Xinde Li, Christian Ritz, Yong Liang Guan, and Shuzhi Sam Ge. Design of chirp waveforms for multiple-access ultrasonic indoor positioning. *IEEE Sensors Journal*, 18(15):6375–6390, 2018.

[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[36] C. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, 1976. doi: 10.1109/TASSP.1976.1162830.

[37] Charles Knapp and Glifford Carter. The generalized correlation method for estimation of time delay. *IEEE transactions on acoustics, speech, and signal processing*, 24(4):320–327, 1976.

[38] David Kurc, Vaclav Mach, Kristian Orlovsky, and Hassan Khaddour. Sound source localization with das beamforming method using small number of microphones. In *2013 36th International Conference on Telecommunications and Signal Processing (TSP)*, pages 526–532. IEEE, 2013.

[39] Zhengping Li, Cheng Hwee Sim, and Malcolm Yoke Hean Low. A survey of emergent behavior and its impacts in agent-based systems. In *2006 4th IEEE International Conference on Industrial Informatics*, pages 1295–1300. IEEE, 2006.

[40] Matthew J Lutz, Chris R Reid, Christopher J Lustri, Albert B Kao, Simon Garnier, and Iain D Couzin. Individual error correction drives responsive self-assembly of army ant scaffolds. *Proceedings of the National Academy of Sciences*, 118(17), 2021.

[41] Amjad Yousef Majid, Casper van der Horst, Tomas van Rietbergen, David JohannesZwart, and R Venkatesha Prasad. Lightweight audio source localization for swarm robots. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2021.

[42] James McLurkin, Andrew J Lynch, Scott Rixner, Thomas W Barr, Alvin Chou, Kathleen Foster, and Siegfried Bilstein. A low-cost multi-robot system for research, teaching, and outreach. In *Distributed autonomous robotic systems*, pages 597–609. Springer, 2013.

[43] Paola Torrico Morón, Jorge Pena Queralta, and Tomi Westerlund. Towards large-scale relative localization in multi-robot systems with dynamic uwb role allocation. *arXiv preprint arXiv:2203.03893*, 2022.

[44] John C. Murray and Harry R. Erwin. A neural network classifier for notch filter classification of sound-source elevation in a mobile robot. In *The 2011 International Joint Conference on Neural Networks*, pages 763–769, 2011. doi: 10.1109/IJCNN.2011.6033298.

[45] Shervin Nouyan, Alexandre Campo, and Marco Dorigo. Path formation in a robot swarm. *Swarm Intelligence*, 2(1):1–23, 2008.

[46] Sohel J Patel and Maciej J Zawodniok. 3d localization of rfid antenna tags using convolutional neural networks. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11, 2022.

[47] Jacob M Peters, Orit Peleg, and L Mahadevan. Thermoregulatory morphodynamics of honeybee swarm clusters. *Journal of Experimental Biology*, 225(5):jeb242234, 2022.

[48] Caleb Rascon and Ivan Meza. Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems*, 96:184–210, 2017. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2017.07.011. URL `https://www.sciencedirect.com/science/article/pii/S0921889016304742`.

[49] A Rashid and A Ali. Performance analysis of low-cost infrared sensors for multi-robot localization and communication. *Int. J. Comput. Appl*, 182:23–29, 2018.

[50] Chris R Reid, Matthew J Lutz, Scott Powell, Albert B Kao, Iain D Couzin, and Simon Garnier. Army ants dynamically adjust living bridges in response to a cost–benefit trade-off. *Proceedings of the National Academy of Sciences*, 112(49):15113–15118, 2015.

[51] Tobias Rodemann, Gokhan Ince, Frank Joublin, and Christian Goerick. Using binaural and spectral cues for azimuth and elevation localization. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2185–2190. IEEE, 2008.

[52] Michael Rubenstein, Christian Ahler, Nick Hoff, Adrian Cabrera, and Radhika Nagpal. Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robotics and Autonomous Systems*, 62 (7):966–975, 2014.

[53] Ehsan Saffari, Ali Meghdari, Bahram Vazirnezhad, and Minoo Alemi. Ava (a social robot): Design and performance of a robotic hearing apparatus. In *International Conference on Social Robotics*, pages 440–450. Springer, 2015.

[54] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics*, pages 10–20. Springer, 2004.

[55] Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.

[56] Jae-Hyun Shin, Byung-Jun Jang, et al. Implementation of a bluetooth 5.1 angle of departure (aod) direction finding system using an software-defined radio (sdr). *The Journal of Korean Institute of Electromagnetic Engineering and Science*, 32(7):644–650, 2021.

[57] Wang Shule, Carmen Martínez Almansa, Jorge Peña Queralta, Zhuo Zou, and Tomi Westerlund. Uwb-based localization for multi-uav systems and collaborative heterogeneous multi-robot systems. *Procedia Computer Science*, 175:357–364, 2020.

[58] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver. In *Proceedings of the 16th annual international conference on mobile systems, applications, and services*, pages 376–388, 2018.

[59] Elahe Soltanaghaei, Adwait Dongare, Akarsh Prabhakara, Swarun Kumar, Anthony Rowe, and Kamin Whitehouse. Tagfi: Locating ultra-low power wifi tags using unmodified wifi infrastructure. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(1):1–29, 2021.

[60] David Stier, Asta Wu, Adyasha Mohanty, and Grace Gao. A test platform for uwb-based localization of dynamic multi-agent systems. *IEEE Robotics and Automation Letters*, 2022.

[61] Seeed studio. Respeaker 6-mic circular array kit for raspberry pi, . URL `https://wiki.seeedstudio.com/ReSpeaker_6-Mic_Circular_Array_kit_for_Raspberry_Pi/`.

[62] Seeed studio. Grove - speaker plus, . URL `https://wiki.seeedstudio.com/Grove-Speaker-Plus/`.

[63] Ying Tan and Zhong-yang Zheng. Research advance in swarm robotics. *Defence Technology*, 9(1):18–39, 2013.

[64] Fabricio A Toasa, Luis Tello-Oquendo, Carlos R Peńafiel-Ojeda, and Giovanny Cuzco. Experimental demonstration for indoor localization based on aoa of bluetooth 5.1 using software defined radio. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–4. IEEE, 2021.

[65] Ali E Turgut, F Gokce, Hande Celikkanat, L Bayindir, and Erol Sahin. Kobot: A mobile robot designed specifically for swarm robotics research. *Middle East Technical University, Ankara, Turkey, METU-CENG-TR Tech. Rep*, 5(2007), 2007.

[66] J-M Valin, François Michaud, Jean Rouat, and Dominic Létourneau. Robust sound source localization using a microphone array on a mobile robot. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 2, pages 1228–1233. IEEE, 2003.

[67] J-M Valin, François Michaud, Brahim Hadjou, and Jean Rouat. Localization of simultaneous moving sound sources for mobile robot using a frequency-domain steered beamformer approach. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 1, pages 1033–1038. IEEE, 2004.

[68] Casper van der Horst, Mees Jonker, Amjad Yousef Majid, Lucan de Groot, R Venkatesha Prasad, Koen Langendoen, and Chris Verhoeven. Ai-based simultaneous audio localization and communication for swarms. Under submission.

[69] Barry D Van Veen and Kevin M Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE assp magazine*, 5(2):4–24, 1988.

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

[71] Ron Weinstein. Rfid: a technical overview and its application to the enterprise. *IT professional*, 7(3): 27–33, 2005.

[72] V. Willert, J. Eggert, J. Adamy, R. Stahl, and E. Korner. A probabilistic model for binaural sound localization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(5):982–994, 2006. doi: 10.1109/TSMCB.2006.872263.

[73] Robert Sessions Woodworth, Bernard Barber, and Harold Schlosberg. *Experimental psychology*. Oxford and IBH Publishing, 1954.

[74] Martin Woolley. Bluetooth direction finding: A technical overview. *Bluetooth Resources*, 2019.

[75] Yu Xianjia, Li Qingqing, Jorge Peña Queralta, Jukka Heikkonen, and Tomi Westerlund. Applications of uwb networks and positioning to autonomous robots and industrial systems. In *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–6. IEEE, 2021.

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

[77] Hongyun Ye, Biao Yang, Zhiqiang Long, and Chunhui Dai. A method of indoor positioning by signal fitting and pdda algorithm using ble aoa device. *IEEE Sensors Journal*, 2022.

[78] Karim Youssef, Sylvain Argentieri, and Jean-Luc Zarader. A learning-based approach to robust binaural sound localization. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2927–2932. IEEE, 2013.