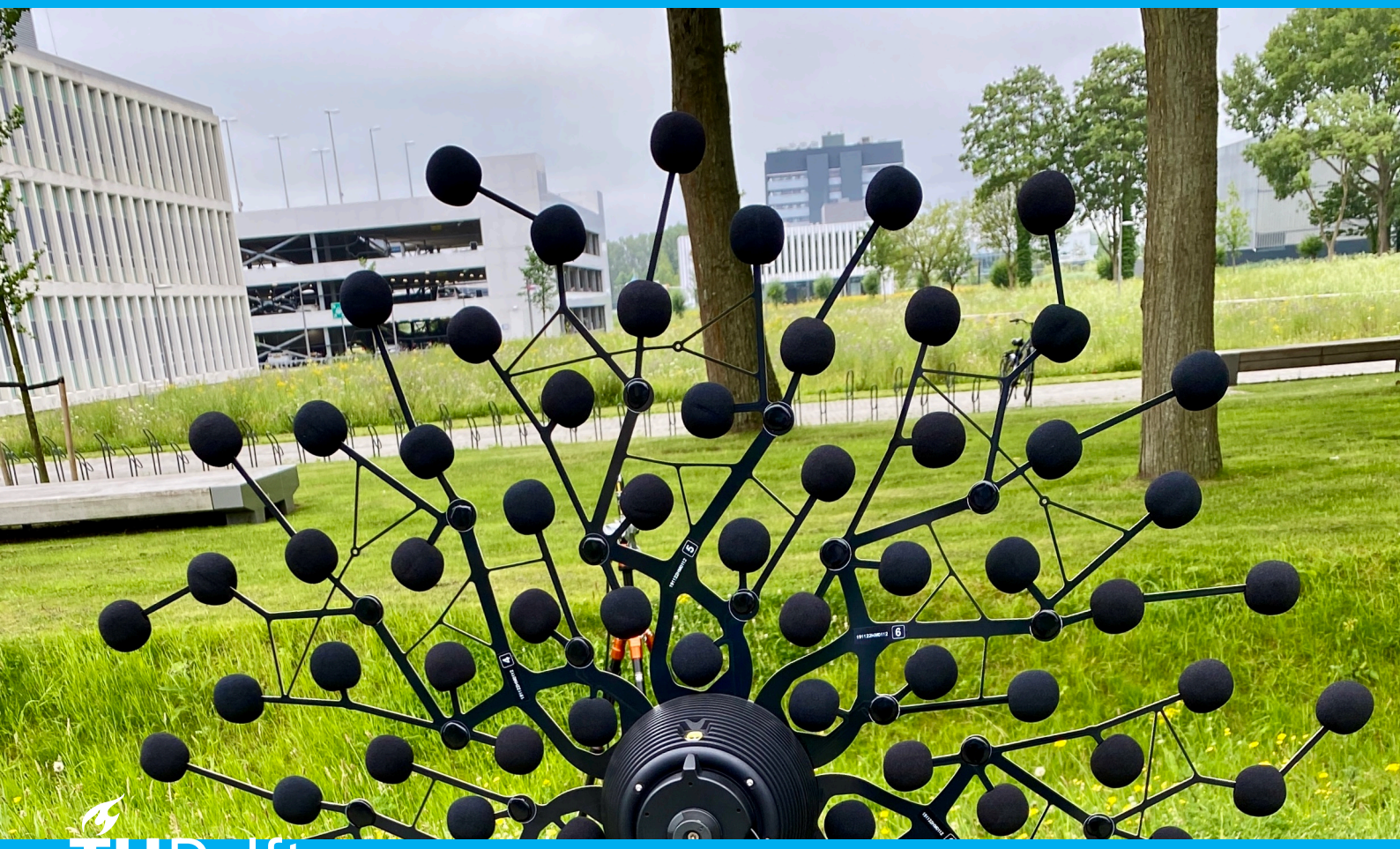
A comparison of 3-dimensional acoustic localization methods based on global optimization and neural networks

Daan Schoorl



TU Delft

A comparison of 3-dimensional acoustic localization methods based on global optimization and neural networks

by

Daan Schoorl

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on 23-12-2021.

Student number:	4281950
Project duration:	Start 01-02-2021 end 23-12-2021
Thesis committee:	Prof. dr. ir. M. Snellen
	Prof. dr. ir. D. Simons
	Dr. ir. M. Lourenço Baptista

TU Delft

Acknowledgements

This document contains my graduation thesis, which is the final step to complete the master Aircraft Noise and Climate Effects at the faculty of Aerospace Engineering of TU Delft. The past months I have been investigating possibilities to combine acoustic data with global optimization methods and neural networks. Prior to this research, I had little to no experience with these methods. Therefore I am grateful this project gave me the opportunity of developing interest in these topics. I believe to have found a great match, considering the fascinating thesis project I was able to work on.

Throughout the duration of the project I have been guided by Mirjam Snellen. I would like to express my gratitude towards her for patiently coaching me through the graduation process. During the project, I much enjoyed the numerous discussions we had on the fundamental basics and limitations in the field of acoustics. I really appreciated the freedom given to shape the project towards its current coverage. Thank you Mirjam for all the help during my graduation project!

Second, I would like to thank Irina Besnea and Bieke von den Hoff for assisting me with the recordings of the microphone array. Besides helping with the experimental recordings they were always accessible to answer questions, discuss theories or help out. I would like to thank professor Dick Simons for his guidance during the meetings.

During one of the latest chapters of the graduation project, the project obtained an additional subject by implementing neural networks. As I was unfamiliar with the field so far, a coached crash course was a initiated. I would like to thank Erik ten Oever and Kars Trommel for their help. With their help, each question and discussion could easily be answered. They assisted in creating a practical understanding of neural networks.

Finally, I would like to thank my family and friends for given me a stage to talk about the project or provide entertaining distractions. This was much appreciated, especially due to the fact that a significant part of the research was conducted from home.

R.D. Schoorl
Rotterdam, December 2021

Contents

List of Figures

List of Tables

List of Abbreviations

BOA	Butterfly optimization algortihm
DE	Differential evolution
MAD	Mean absolute deviation
MPT	Mean processing time
NN	Neural network
PSD	Power spectral density
ReLU	Rectified linear unit

Introduction

The past century surveillance methods have been increasing, however the field of surveillance is never completed. By developing new surveillance methods, safety can be increased at sensitive public areas. This research is focused on investigating the possibilities of locating single acoustic sources based on emitted noise.

A possible application could be drone localization. In December 2018 drones were spotted around Gatwick Airport. Due to potential collision risks, the airport was closed resulting in more than hundreds of flights to be cancelled and over a thousand flights were impacted. Commercial drones are often too small for radar equipment to be picked up [9]. Therefore the location of small commercial drones can be hard to determine. The methods proposed in this research could be of interest considering extensions of current surveillance methods.

Another application could be tracking marine life under water. Some sea animals communicate under water by means of sound. These sound waves could be captured by under water microphones. This would require additional research due to the difference in environmental aspects. By implementing the localization methods to the under water environment, it might be possible to locate sea animals based on the sound they produce. The gathered information could become of importance to monitor under water wild life populations.

The research is aimed at locating an acoustic source using different approaches. The research compares a proven global optimization method with two methods which have not yet been combined with 3-dimensional acoustic source localization. Global optimization methods and neural networks were implemented to localize acoustic sources.

The research was done as a graduation project in the section Aircraft Noise and Climate Effects of the Aerospace Engineering faculty at Delft university of Technology. The section focuses on investigating noise aspects and climate impact related to the aviation industry.

The report is divided into three main components. The first part contains a scientific paper, the second part a literature survey and the final part includes supporting information. The scientific paper presents the methods which were investigated during the research. After explaining, the methods were tested on simulated and experimental data. The results are used to quantify the performance of the methods and compare them. The second part contains a literature survey, which has been conducted at the start of the research. The literature survey was used to obtain knowledge in the field of acoustics and optimization. The final part consists of supportive information. Some of the fundamentals used in the scientific paper are explained more extensively. Besides supportive information, additional results are presented.

I

Scientific paper

A comparison of 3-dimensional acoustic localization methods based on global optimization and neural networks

Daan Schoorl, *

Delft University of Technology, Delft, The Netherlands

Abstract

This paper presents a research on the localization of an acoustic source in a 3-dimensional space. The application could be an addition to current surveillance methods by localizing acoustic sources. The investigated localization methods are global optimization and neural networks. The evaluated global optimization methods consist of differential evolution, i.e a variation of the well-known genetic algorithm and the butterfly optimization algorithm. The neural network investigated during the research was a multi-layer perceptron network, which was trained using synthetic data. The three approaches are compared using both simulated and experimental data. Performance of the three methods was assessed by determining their accuracy in locating the acoustic source and the corresponding computational demand. Although the neural network showed low computational demands (neglecting the training phase) compared to differential evolution and the butterfly optimization algorithm. The accuracy of the localization was better for the approaches based on the global optimization methods.

1 Introduction

In the past years, the commercial availability and popularity of drones has been increasing. There are few limitations when it comes to buying drones and in many cases there is no operating knowledge required. The absence of these requirements can lead to threats for society or public facilities. Possible threats could be violating peoples privacy, attacks on public facilities or public annoyance due to noise. The incident at Gatwick Airport in December 2018 showed that a drone is capable of disrupting an entire airport. As current surveillance methods fall short of resolving this issue, new methods should be developed to increase safety [1].

One of the possibilities to localize drones is based on the noise produced by drones. In addition, there are many other application where source localization is of importance. The noise can be captured by a microphone array. A proven method to localize and quantify acoustic sources using arrays is beamforming. This method uses the phase differences of the received signal over the microphones. Beamforming can be used to estimate source levels at locations in a scan plane. A beamform plot presents these source levels in a 2-dimensional image. Low levels indicate no source and high levels present sources. Calculating the source level at each of the scan points could be considered an exhaustive search. When applying an exhaustive search, the method is restricted to a limited amount of unknowns which prevent the method of being suitable for more general applications. Beamforming can be extended from the second dimension to the third dimension by adding multiple grid planes. This will increase the exhaustive search to become more computational demanding. Therefore new methods are required to solve this issue. Global optimization methods are one of the possibilities. Global optimization methods are capable of searching large solution spaces based on few assumptions. A disadvantage of global optimization methods is the possibility of finding a local optimum instead of the global optimum. Beamform plots arguably show higher energy levels not only at the location of the source, but also at locations where there is no source, called sidelobes. Sidelobes could be interpretend as local optima which can set global optimizations methods off track. However global optimization methods do have the ability of escaping local optima while searching the global optimum [2].

The method of Malgoezar et al. has proven to be capable of locating one or multiple acoustic sources in a 3-dimensional space. To achieve this result he combined differential evolution with beamforming energy functions to efficiently search through acoustic environments. Beneficial of the work of Malgoezar et al. is the reduction in required forward calculations and high accuracy while maintaining a high probability of locating the global optimum [3][4].

In the work of Arora and Singh, they introduce the butterfly optimization algorithm. This global optimization method is inspired by the movement of butterflies. In their work they compare the butterfly optimization algorithm with other global optimization methods, including differential evolution. The comparison is made by applying optimization techniques

*Msc Student, Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology

to a set of 30 benchmark functions. They search for the optimal solutions of the benchmark functions to evaluate the performance of the tested methods. The performance of the global optimization methods are compared using mean deviation, standard deviation and speed. The results indicate that the butterfly optimization algorithm is capable of finding global optima with a fast convergence speed [5].

In recent years deep learning has made rapid development in a wide variety of research fields. In the work of Kujawski et al. a convolutional neural network was used to locate acoustic sources using simulated microphone array data. The convolutional neural network was trained by feeding 2-dimensional beamform images. The investigated square scan plane had x and y dimensions ranging from -0.343 to 0.343 [m] at a fixed distance of 0.343 [m] from the microphone array. The amount of grid points at which beamforming was applied consisted of 51 by 51 points. The frequencies investigated ranged from 500 to 6500 [Hz]. The goal of the research was to locate acoustic sources and determine their source strength. The method appeared to be effective at locating a single acoustic point source with sub-grid accuracy. When investigating acoustic sources at higher frequencies, more side lobes arise resulting in a more challenging environment to locate acoustic sources. During the evaluation of the performance, the estimated location accuracy of the convolutional neural network appears to be independent of the frequency. However combining convolutional neural networks with beamforming does come with shortcomings. To estimate 2-dimensional locations and source strengths of a single acoustic source, a beamform image is required. Therefore the method is subjected to an exhaustive search prior to application [6].

At the moment deep learning is an attractive tool to process data. By training algorithms to recognise patterns in data, a wide variety of problems can be solved at high speeds. Ma and Liu [7] compared beamforming, DAMAS and convolutional neural networks to locate simulated acoustic sources. To test each of the three methods, simulated data was fed to the methods to compare the localization accuracy at different frequencies. The convolutional neural network was trained by presenting cross spectral matrices with corresponding locations. The research is aimed at locating multiple acoustic sources in a 2-dimensional x and y plane with dimensions between -0.8 [m] and 0.8 [m]. The acoustic sources were placed at a fixed radial distance of 2 [m] from the microphone array. The work concludes that convolutional neural networks are capable of locating multiple acoustic sources. Specifically at higher frequencies the accuracy of the estimates for source position appears to be increasing.

Castellini et al. used a different approach to combine neural networks with a phased microphone array. In the research a multi-perceptron neural network was used to analyze measurements taken by a microphone array. The aim was to locate acoustic sources in a 2-dimensional plane. Castellini applied a few mathematical operations to reshape the cross spectral matrix for the neural network. By using the cross spectral matrix almost directly, most of the steps used with beamforming are not required. The locations of the acoustic sources considered by Castellini varied between -0.5 [m] and 0.5 [m] in the x and y plane. The sources were located at a fixed radial distance of 2 [m] from the microphone array. The network was trained with a synthetic data set consisting out of 1000000 cross spectral matrices each corresponding to a different source location. The network has proven to be capable of locating acoustic sources and finding the corresponding source strengths [8].

The aim of the research presented in this paper is to compare acoustic localization methods based on different approaches. Often research done with a phased microphone array is focused on a 2-dimensional search in the x and y directions with fixed radial distance z. This is likely a result of all microphones being positioned in the x and y plane and therefore obtain less accurate measurements in the radial direction [4]. This research will include the radial direction and therefore search for acoustic sources in a 3-dimensional search space.

In section 2 the methodology is described. The principles of the optimization methods and artificial neural networks are explained. Section 3 presents the applications of all three methods. This section connects theory to practice by explaining how the methods are combined with locating acoustic sources. The localization methods are compared based on simulated and experimental data. The results will evaluate the performance of each method in section 4. After the results are presented, possible impacting factors are discussed in section 5. Finally section 6 will state the findings of the research in the conclusion.

2 Methodology

In this study acoustic localization is investigated using three methods. The first method considered is differential evolution, the second method the butterfly optimization algorithm and the third method makes use of neural networks. Differential evolution and the butterfly optimization algorithm can be classified as metaheuristic methods. A metaheuristic method has the capability of searching through large solution spaces making no or few assumptions. All three methods were set to use cross spectral matrices of the acoustic data measurement by the arrays. Based on the acoustic data the locations of single acoustic sources are estimated in a predefined search area. The input data consisted of simulated and experimental data.

2.1 Differential evolution

Differential evolution is based on an iterative process to optimize candidate solutions to an optimization

problem. Each iteration is denoted as a generation and the number of iterations is set to give a high probability of finding the solution to the optimization problem. The method combines existing populations with new populations and keeps the improved solutions at the next generation. Besides keeping the most suitable solution, differential evolution keeps some of the less attractive solutions, thereby creating possibilities to escape local optima. The population consists of q members. The population members are presented by $\mathbf{m_{k,u}}$ in which k specifies the number of the generation and u denotes an individual population member. $\mathbf{m_{k,u}}$ is a vector containing the elements of a possible solution to the optimization problem. In case of a 3-dimensional search the three elements of $\mathbf{m_{k,u}}$ are x, y and z. Considering the acoustic solution space $\mathbf{m_{k,u}}$ will start at a randomly chosen location within the limits of the search area. Partner population $\mathbf{b_{k,u}}$ is created from population $\mathbf{m_{k,u}}$, presented in formula 1.

$$\mathbf{b}_{k,u} = \mathbf{m}_{k,u_1} + F(\mathbf{m}_{k,u_2} - \mathbf{m}_{k,u_3}) \tag{1}$$

The variable of u_1,u_2,$u_3 \in \{1, 2, , q\}$ denote specific members of population $\mathbf{m_{k,u}}$. The subscript u values in equation 1 are chosen at random, denoting that partner populations are generated by combining random chosen population members from the current population. F denotes a scalar multiplication factor with a preset value between 0 and 1. Increasing the value of F will result in larger differences between population members $\mathbf{m_{k,u}}$ and partner populations $\mathbf{b_{k,u}}$.

In formula 2 is determined whether an original population member or a member from the partner population is selected to create new descendants $\mathbf{d_{k,u}}$.

$$\mathbf{d_{k,u,v}} = \begin{cases} \mathbf{m_{k,u,v}} & \text{if} \quad r \geq p_c \\ \mathbf{b_{k,u,v}} & \text{if} \quad r < p_c \end{cases} \tag{2}$$

The crossover probability p_c defines the threshold on which population members are selected to proceed towards the next generation. The values of F and p_c can be altered to tune the algorithm. Parameter r defines a uniform distributed random value between 0 and 1. In formula 3 a comparison is made between energy values E from descendants $\mathbf{d_{k,u}}$ and current generation $\mathbf{m_{k,u}}$.

$$\mathbf{m}_{k+1,u} = \begin{cases} \mathbf{d}_{k,u} & \text{if} \quad E(\mathbf{d}_{k,u}) < E(\mathbf{m}_{k,u}) \\ \mathbf{m}_{k,u} & \text{if} \quad E(\mathbf{d}_{k,u}) \geq E(\mathbf{m}_{k,u}) \end{cases} \tag{3}$$

Based on the values obtained for the energy function, the current population or the descendants are selected for the next generation. This process is repeated until the preset amount of generations is completed. The acoustic source location corresponding to the population member with the least overall energy value is saved during each generation. The differential evolution algorithm used during this research is a minimization problem. The energy values of the population members $\mathbf{m_{k,u}}$ and descendants $\mathbf{d_{k,u}}$ can be determined by the following formulas. First $r_{n,j}$ is created, containing the distance between each of the microphones and the locations selected by the algorithm. Distance $r_{n,j}$ is presented in formula 4, in which n denotes a specific microphone and j the location of a potential acoustic source specified by the algorithm.

$$r_{n,j} = \sqrt{(x_n - x_j)^2 + (y_n - y_j)^2 + (z_n - z_j)^2} \tag{4}$$

In formula 5, the so called steering vector $g_{n,j}$ is presented.

$$g_{n,j} = e^{-2\pi i f(\frac{r_{n,j}}{c})} \tag{5}$$

The steering vector contains the phase difference between the input signals obtained from the simulated microphone array. When the steering vector components $g_{n,j}$ between each of the microphones and one specific location are combined vector $\mathbf{g}$ can be constructed. Steering vector $\mathbf{g}$ can be used to calculate beamform output $B(x, y, z, f)$ according to formula 6.

$$B(x, y, z, f) = \frac{\mathbf{g}^* \mathbf{C} \mathbf{g}}{\|\mathbf{g}\|^4} \tag{6}$$

The beamform output presents the acoustic resemblances between the phase differences at different grid locations. When the beamform output is determined at a single point, this can be denoted as the energy value. Therefore the method estimates acoustic source levels at locations chosen by the algorithm, without using a grid. The method strives to find the location within the search area with the most optimal energy value. The final generation will present the overall best estimated source location [4] [9] [10]. A flowchart of the method is presented in figure 1.

2.2 Butterfly optimization algorithm

The butterfly optimization algorithm originates from butterflies communicating by scent. The method is based on an iterative process in which butterflies move through the search area, searching for the highest possible energy value. The movement of the butterflies is based on the positions of other butterflies. The algorithm is set to maximize the solution. The initial step at iteration $t = 1$ is to allocate all butterflies $\mathbf{x_i^t}$ to a random chosen location within the search area. The vector $\mathbf{x_i^t}$ denotes the x, y and z coordinates of butterfly i at iteration t. The energy function used to evaluate the fitness is determined by formula 4, 5 and 6, equivalent to the energy function used by differential evolution. The values obtained from the energy function will be used to determine the stimulus intensity I. The stimulus intensity is defined as the normalized energy value. By normalizing the energy value, the impact of source strength is deducted from the localization process. In formula 7 is presented how the perceived magnitude of fragrance h is calculated.

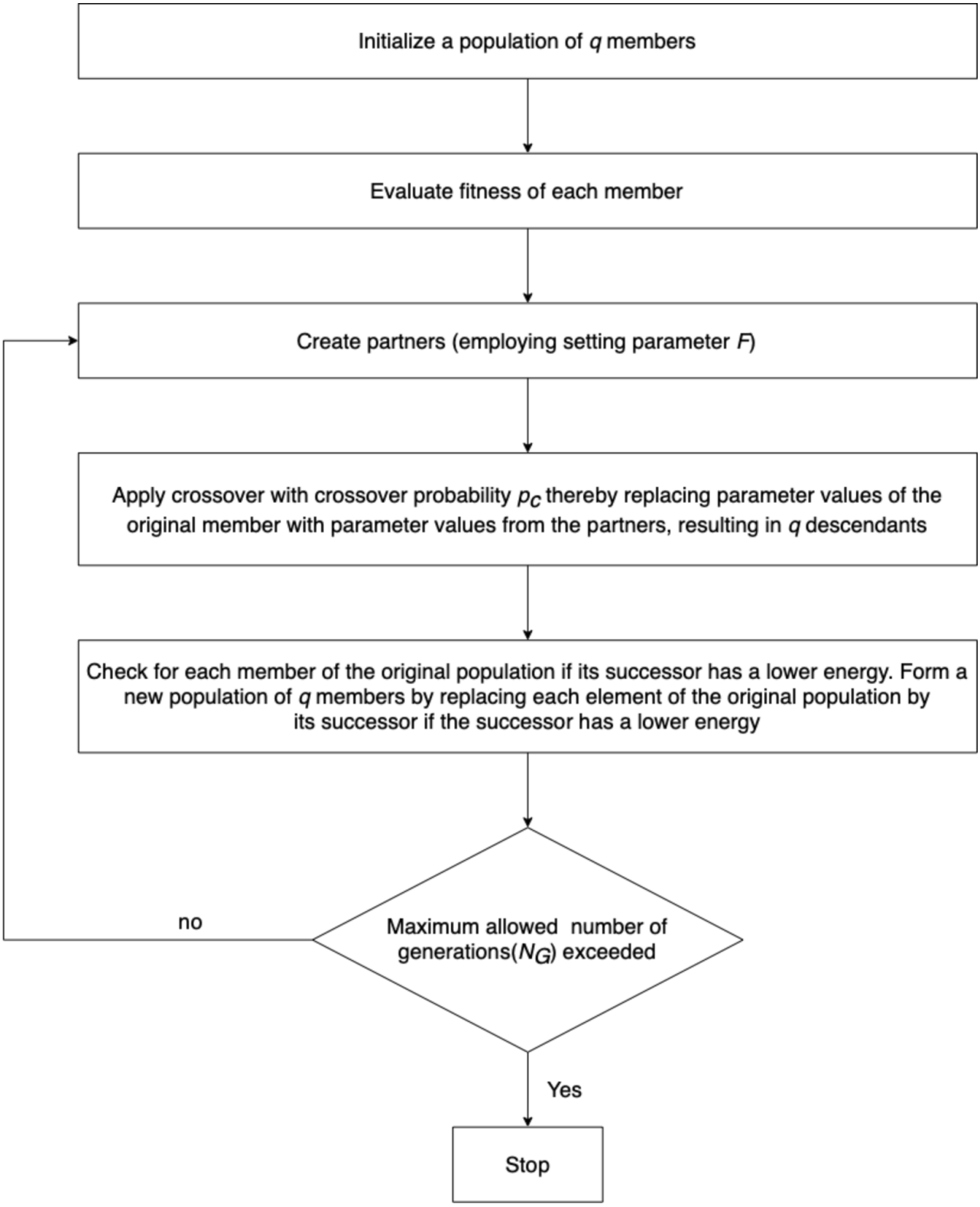


Figure 1: Flowchart differential evolution [10]

$$h = b_{sm} I^a \tag{7}$$

The perceived magnitude of fragrance is used to determine the location of the butterflies at the next iteration. Constant b_{sm} denotes the sensor modality and constant a the power exponent. Because the value of h will impact the step size, the values of b_{sm} and a can be tuned to impact the convergence speed due to their impact the perceived magnitude of fragrance h. After the perceived magnitude of fragrance h is determined for each butterfly, the butterfly with the highest value of h is selected to be the fittest butterfly and labeled k. Therefore vector $\mathbf{x}_k$ contains the coordinates corresponding to the fittest butterfly. The location of the fittest butterfly at each iteration is automatically saved to the next iteration, preserving the best results throughout all iterations. To proceed towards the next iteration a choice is made whether to proceed in a global search or in a local search. For each butterfly at each iteration a uniform random value r is chosen and compared with the preset switch probability p_s. The search type for each butterfly in each iteration is described by formula 8.

$$\mathbf{x}_i^{t+1} = \begin{cases} \text{global search} & \text{if} \quad r < p_s \\ \text{local search} & \text{if} \quad r \geq p_s \end{cases} \tag{8}$$

When a butterfly is selected to take a step within the global search, it will move towards the fittest butterfly according to formula 9.

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + (r^2 \mathbf{x}_\mathbf{k} - \mathbf{x}_i^t) h_i \tag{9}$$

Otherwise a butterfly is selected to proceed in a local search and moves towards a random chosen combination of two other butterflies described by formula 10.

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + (r^2 \mathbf{x}_j^t - \mathbf{x}_l^t) h_i \tag{10}$$

The vectors containing the locations of $\mathbf{x}_j$ and $\mathbf{x}_l$ are randomly chosen location vectors of other butterflies during the same iteration. The iterations are repeated until the preset number of iterations are completed [5] [11]. A flowchart of the butterfly optimization algorithm is presented in figure 2.

2.3 Artificial neural network

The past years artificial intelligence has developed rapidly. Inspired by the sophisticated functionality of the brain, the method became a popular data processing tool. One of the branches of artificial intelligence is neural networks. A neural network consists of layers, links and nodes. Each layer contains of a number of nodes and the nodes of different layers are connected by weighted links. This research will make use of a multi-layer perceptron network. A multi-layer perceptron network has the capability of mapping non-linear relations between input and output data. During the training process, each of the training cycles executed by a neural network is called an epoch. Training is meant to determine the value of the weights. Weights are found using optimization methods. The output of each node depends on its input and its activation function. The activation function combines the node inputs to determine its output. The input and output layer of a neural network are bounded by the parameters of the problem. The size of the input layer is equal to the number of elements in the data. The number of output nodes is bounded by the desired output. During this research the number of input nodes is determined by the number of elements in the cross spectral matrix. The number of output nodes is 3, which is defined by the estimates containing x, y and z coordinates. A neural network is trained by feeding samples and corresponding solutions to the network. Based on the difference between the estimates of the neural network and the actual solutions the weights of the links are adapted. By training a neural network with sufficient training data, the network has the ability to recognize patterns within the data. Once the training has been completed, the neural network has the ability of estimating outputs at a rapid pace [12] [13].

Researchers have applied different types of neural networks to process simulated or recorded data obtained from phased microphone arrays. The application is subjected to a wide variety of design choices depending on the purpose of the neural network. Neural networks have a completely different approach compared to global optimization methods. One of the key architecture choices in designing a neural network is the optimizer. The optimizer determines the weight changes and strives to minimize the error of the loss function. The loss function is a measure to quantify the error between the trained network estimates and the actual source locations. Different loss functions can be used depending on the application of the network.

The learning rate of the neural network controls the step size at which the weights are adapted. By using large steps a neural network is able to learn fast. However, choosing a large learning rate can negatively impact the performance of the model by finishing with a sub optimal set of weights. A smaller learning rate may lead to finding a more optimal set of weights at the cost of longer training time. Figure 3 visualizes possible differences between step sizes for a 2-dimensional situation. The situation presented on the left side of figure 3 describes large steps, creating the possibility of stepping over the global optimum. The situation on the right side of figure 3 presents small steps, resulting in longer training times with a more optimal solution. When using a learning rate too small there is a possibility of ending up in a local optimum.

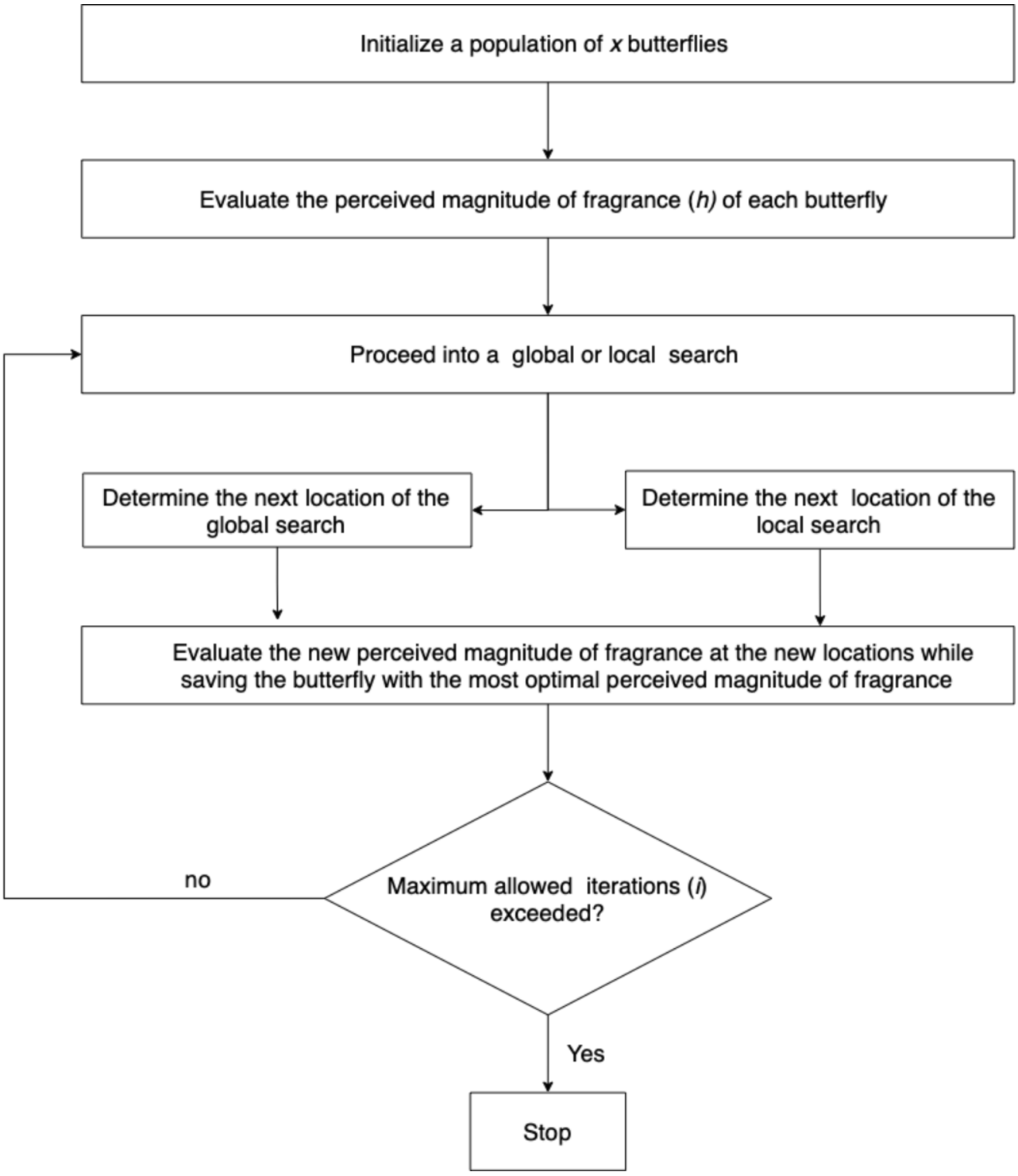


Figure 2: Flowchart butterfly optimization algorithm

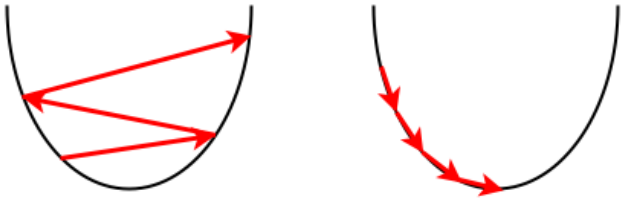


Figure 3: Difference in learning rate, large learning rate (left) and small learning rate (right)

The activation functions of the nodes define its output relative to the input. Each node in the same layer has the same activation function as these are selected per layer. The activation functions considered in this research are rectified linear units and a linear activation function. The rectified linear unit is a default choice in many feed forward neural networks today and is presented by figure 4.

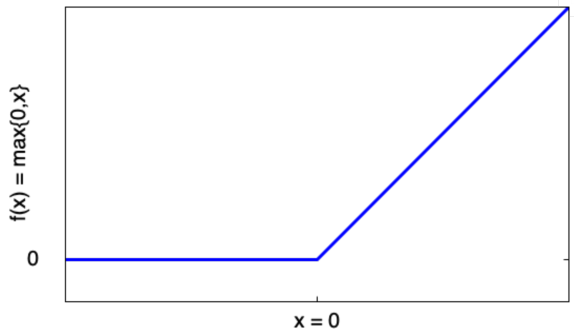


Figure 4: Rectified linear units (ReLU) activation function.

During the research a feed forward neural network was used. A feed forward network allows a signal only to move from input to output, these networks are mainly used for pattern recognition. The rectified linear unit reacts close to a linear function. The rectified linear unit activation function has a threshold value of 0 and is presented in formula 11.

$$f(x) = max(0, x) \tag{11}$$

The rectified linear unit implies the output is 0 if $x < 0$ and the output is a linear function $f(x)$ if $x > 0$. The value of x denotes the nodes input value. The rectified linear unit activation function can only provide outputs which are larger or equal than zero [14] [15].

During the research, the final layer of the neural network contains a linear activation function. A linear activation function is commonly used as the output layer for regression problems. The linear activation function enables the final output values to become positive and negative, which is desired considering the position of the microphone array relative the search area [8] [16].

The neural network created during this research is inspired by the neural network of Castellini et al. [8]. The cross spectral matrix could be considered an image which stores the phase relations of the acoustic scenario. Depending on the design choices, the cross spectral matrix is reshaped before being fed to the neural network. Castellinis approach on feeding data to the neural network is based on the principle that every cross spectral matrix is hermitian by nature. The cross spectral matrix contains phase differences between microphone signals. Therefore the diagonal elements should contain phase differences between an input signal of single microphones with the same input signal. Since the phase difference between an input signal and itself should be 0, the diagonal can be considered contaminated else wise. Therefore the main diagonal values are set equal to 0. Before feeding the cross spectral matrix to the neural network, the cross spectral matrix is split over its diagonal. The imaginary values are removed from the upper right diagonal and the real values are removed from the lower left diagonal. Both parts are joined again forming a reshaped cross spectral metric preserving all relevant information. This process is visualised in figure 5. Before feeding the reshaped cross spectral matrix to the neural network the matrix is normalised between values of 0 and 1. The normalised cross spectral matrix is split up into rows which are concatenated to form a single row containing a whole cross spectral matrix.

3 Case Studies

3.1 Simulation set-up

During the simulations all three methods were tested and trained to estimate acoustic source locations within the search area. The search area consists of a cube with sides of 40 [m] each. The microphone array was positioned in the middle of the bottom surface facing up towards the search area. The simulated microphone array consists of 64 microphones. The microphone configuration of the array used during the simulations is presented in figure 6.

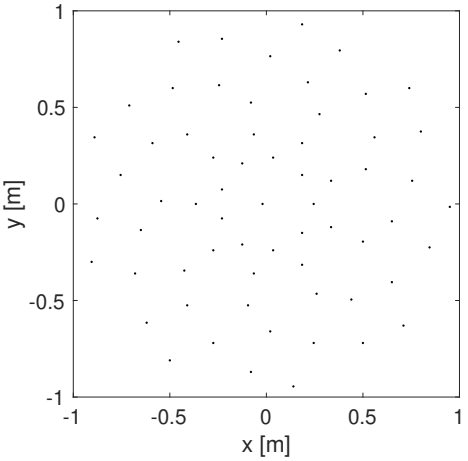


Figure 6: 64 Microphone positions of simulated phased microphone array

In the center of the microphone array, the microphones are located close together. Microphones located close together are preferable to record high frequencies due to small wavelength λ. At the outer parts of the microphone array the microphones are

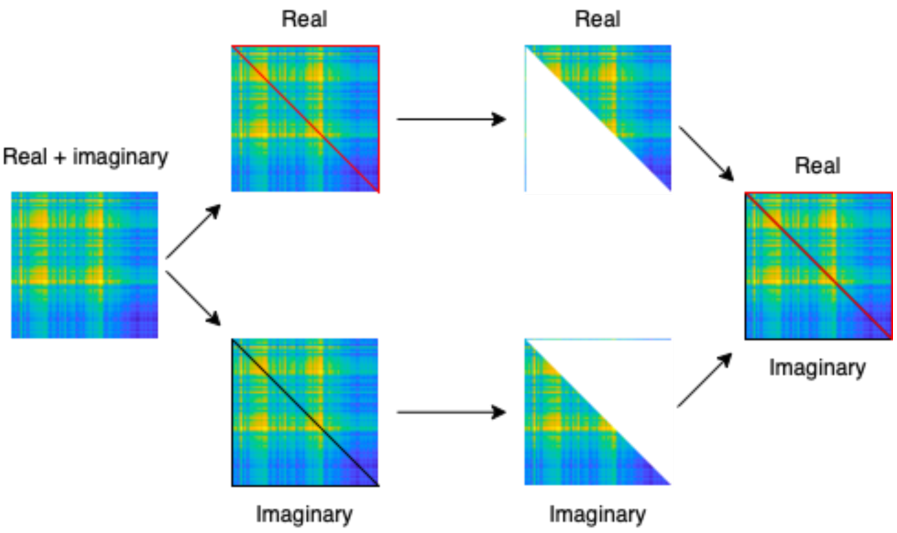


Figure 5: Cross spectral matrix division of Castellini

more spread, making the outside areas more suitable to capture large wavelengths. Formula 12 displays the relation between wavelength λ and frequency f. The constant c denotes the speed of sound. The processing of simulations and experiments is conducted using a 2017 Macbook Pro with 2.9 GHz Quad-Core Intel Core i7 processor and 16 GB 2133 MHz LPDDR3 ram storage.

$$\lambda = \frac{c}{f} \tag{12}$$

Simulated data was used to train and compare the methods. Each of the simulated cross spectral matrices accommodates a single omni-directional acoustic source at a uniformly random selected location within the search area. The sources emit a tone at $f = 3000$ [Hz] with a source strength of 10^{-8} $[Pa^2]$. Formula 13, 14 and 15 present the simulation of a single cross spectral matrix under ideal circumstances. Formula 13 presents the construction of predicted signals, phase variations over the array and the effect of geometrical spreading on the level. Parameter r_m denotes the distance between the acoustic source and each of the n microphones.

$$a_m = \frac{1}{r_m} e^{-2\pi j f(r_m)/c} \tag{13}$$

In formula 14, output vector $y(f)$ is obtained by multiplying the steering vector a_m with acoustic waveform $s(f)$.

$$y(f) = a_m(x_s)s(f) \tag{14}$$

In formula 15, vector output $\mathbf{y(f)}$ is multiplied with the transposed complex conjugate of $\mathbf{y(f)}$ and divided by 2 to obtain cross spectral matrix.

$$C = \frac{\mathbf{y}(f)\mathbf{y}(f)^*}{2} \tag{15}$$

This process is repeated to obtain multiple cross spectral matrices corresponding to different source positions. To compare the three methods a test data set was generated consisting of 1000 cross spectral matrices and corresponding source locations.

3.1.1 Differential evolution

The first method tested on the test data set was differential evolution. Differential evolution was executed while using the setting parameters displayed in table 1.

Variable	Value
Crossover probability p_c	0.74
Multiplication factor F	0.64
Population size q	64
Number of generations Ng	400

Table 1: Differential evolution variables

The setting parameters for differential evolution and the butterfly optimization algorithm were selected to grant each of the global optimization methods with an equal amount of inversions. An inversion denotes each time the energy function was calculated. The number of inversions for differential evolution is determined by multiplying the number of generations Ng with population size q.

3.1.2 Butterfly optimization algorithm

The second method tested is the butterfly optimization algorithm. The parameter settings used during simulation are presented in table 2.

Variable	Value
Switch probability p_s	0.7
Sensor modality b_{sm}	1.8
Absorption coefficient a	0.1
Number of iterations t	32
Number of butterflies i	800

Table 2: Differential evolution variables

The amount of inversions set for differential evolution was matched to create equal chances in locating the acoustic sources. The number of inversions made by the butterfly optimization algorithm can be determined by multiplying the number of iterations t with the number of butterflies i. The energy function used by the butterfly optimization algorithm is constructed with the same formulas 4 ,5 and 6 as for differential evolution. The algorithm is finished when the stopping criterion is met at $t = 32$ iterations.

3.1.3 Neural network

Before comparing the three methods, the neural network was trained and validated with training data containing 840000 cross spectral matrices. At each epoch the data set is shuffled to change the order at which the samples are presented to the network. By shuffling the data set at each epoch, the neural network becomes more suitable at finding a general pattern in the training data. The data set was split between 80% training data and 20% validation data. The first 80% of the samples is used to train the network and the remaining 20% is used to determine the validation loss. The validation loss can be used to quantify the progress of the training. The loss function used during this research is the mean absolute error. The mean absolute error is presented in formula 16. In which L denotes the number of samples and e_l the error between the estimated values and the actual values.

$$Mean\ absolute\ error = \frac{1}{L}\sum_{l=1}^{L}|e_l| \qquad (16)$$

At each epoch the loss function is used to determine the validation loss, this implies comparing the models estimates with the actual solutions. The mean absolute error is used to determine the difference between the estimated and the actual value. When the validation loss starts to increase, the network starts overfitting. Overfitting implies the network is becoming very capable of estimating outputs to corresponding inputs within the training data set but less suitable at estimating unseen data. Therefore overfitting is a undesired effect. During the training process, the validation loss was monitored at each epoch. Each time the validation loss decreased, a copy of the weights is saved to preserve the best model parameters. The batch size of the neural network was set at 32 samples. The batch size defines the number of training samples passing by before updating the models weighted links. The cross spectral matrices fed to the neural network were formulated according to Castellinis method described in section 2.3. After training, the neural network was tested using the same test data set used to test differential evolution and the butterfly optimization algorithm. The layers and nodes used in the network are presented in table 3. The network consists of 6 layers including the input and output layer. The first 4 active layers have a rectified linear unit activation function and the final layer has a linear activation function.

Layer	Neurons	Activation function	Type
1	4096	-	Input
2	400	ReLU	Fully connected
3	200	ReLU	Fully connected
4	50	ReLU	Fully connected
5	20	ReLU	Fully connected
6	3	Linear	Fully connected

Table 3: Neural network architecture simulation

The optimizer used by the neural network during the research was stochastic gradient descent. Stochastic gradient descent is an optimizer which can be applied to change the weights of the links. By changing the weights of the links, the optimizer aims to reduce the error values of the loss function. By training the neural network with random generated samples, the method aims to create a network which can solve general problems. The neural network used during simulations is presented in figure 7. The figure shows the first step in which the cross spectral matrix is reshaped to vector format before feeding the cross spectral matrix to the neural network. The first layer consists of the input layer which has the same amount of nodes as there are elements in the cross spectral matrix. The 4096 input nodes result from multiplying the 64 input signals with its complex conjugates as described in formula 15.

3.2 Experimental set-up

In addition to comparing the methods based on synthetic data, real measurements were considered. The microphone array used was a Bionic M-112 from CAE which contains 112 microphones. Differential evolution and the butterfly optimization algorithm were slightly adapted to fit the new microphone configuration. The neural network required new training due to the increased amount of inputs. The positions of the microphones are presented in figure 8. The recordings were made outside, causing exposure to environmental noise. Figure 9 presents the setup for the recordings above grass. The first two recordings were captured above a grass surface. The third recording was captured above a stone surface. The size of the search area used during the simulations was kept constant during the experiments.

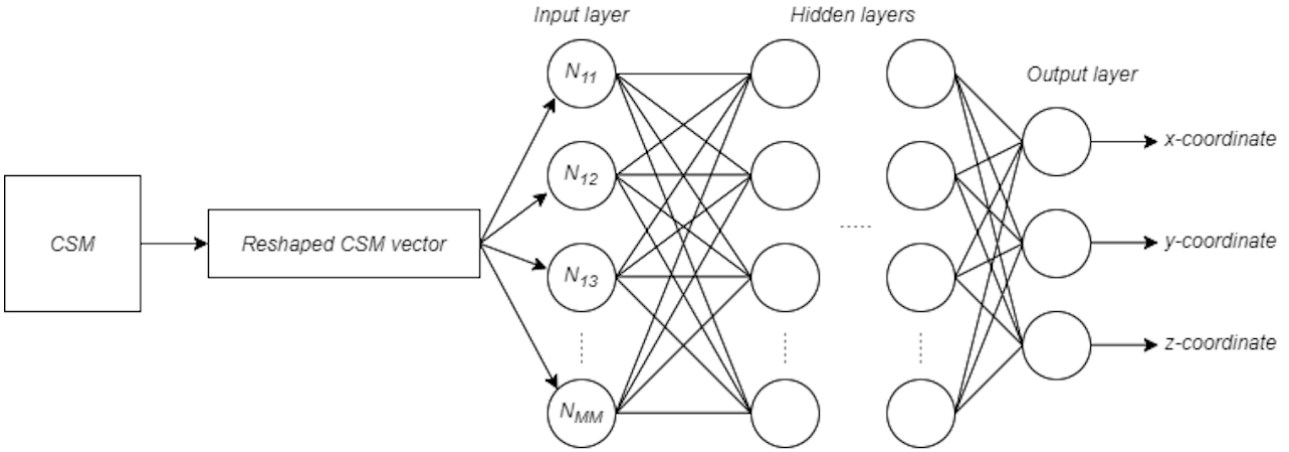


Figure 7: Multi-layer perceptron network

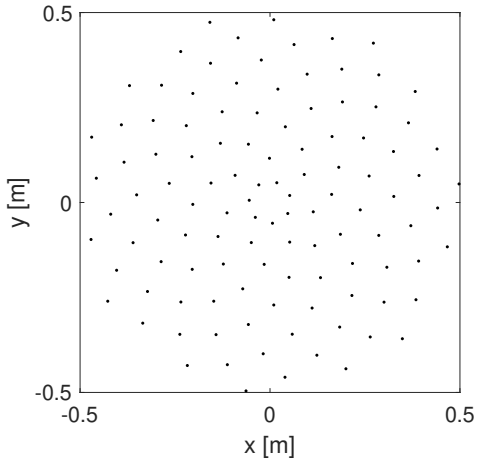


Figure 8: 112 Microphone positions of experimental phased microphone array

Figure 9: Experimental set-up

The acoustic source used during the recordings was an omni-directional source which emitted a multi-tone signal consisting out of 5 different frequencies. The multi-tone emitted frequencies at 200, 1000, 2000, 3500 and 5000 [Hz]. The acoustic source was placed at different locations to investigate the effect of surrounding influences. Table 4 presents the coordinates and the surface material at which the acoustic source an microphone array are located. Since differential evolution was proven to be successful it was used to confirm and improve on the measured source locations. The algorithm was set with an excessive amount of inversions by increasing the population size and the number of generations significantly. This process was repeated 100 times, all resulting to the coordinates presented in table 4. Because differential evolution was assumed to improve on the measurement accuracy, the decision was made to continue with the locations defined by differential evolution.

Recording	x[m]	y[m]	z[m]	Bottom
1	-0.4910	0.0972	2.9423	Grass
2	-0.6109	-0.6468	6.4229	Grass
3	-1.0117	0.2855	4.2959	Stone

Table 4: Recorded data locations

The acoustic input received by the phased microphone array consists of pressure differences recorded over time. The acoustic source was placed on a tripod to create a stationary source position. The investigated time signal was Hanning weighted, zero-padded and Fourier transformed to the frequency domain. The Hanning weighting was included to reduce the impact of sidelobes in the frequency domain. Zero-padding was applied to create a more clear representation of the frequency spectrum. The cross spectral matrices were extracted from the recorded signal at a frequency of 3500 [Hz] [17] [18].

The Neural network is trained again due to the increase in microphones of the second array. The amount of microphones in the array determines the size of the cross spectral matrix. The second microphone array consisted of 112 microphones, resulting in a cross spectral matrix containing 12544 elements. A new data set was created according to the configuration of the array. Each cross spectral matrix used was created at a frequency of 3500 [Hz]. During training the data set was split in 80% training data and 20%

validation data. 245000 samples were used to train the network. The decrease in sample size is a result of computational limitations. Besides the input layer, the network architecture was kept consistent. The amount of nodes in the input layer increased from 4096 nodes to 12544 nodes.

4 Results

4.1 Simulation results

The performance of the acoustic localization methods is evaluated based on accuracy and required processing time. The test data set containing 1000 samples was used to quantify the results. Table 5 presents the mean absolute deviation between the simulated acoustic source positions and the estimates by the localization methods. Besides the mean absolute deviation (MAD) the table presents the mean processing time (MPT) of a single sample.

	DE	BOA	NN
MAD $[m]$	0.0983	0.2417	0.2883
MPT $[s]$	4.567	1.234	1.634e-4*

Table 5: Mean absolute deviation (MAD) and mean processing time (MPT). * Denotes the time to estimate coordinates after training

The box-plots in figure 10 presents the absolute deviation between estimates and true source positions for each method. The absolute deviation of each sample is determined by calculating the difference between the estimated coordinates and the actual source coordinates. Global optimization methods have the ability of finding local optima instead of the global optimum. This could be a characteristic of the neural for network as well. Some outliers are presented in the boxplots of figure 10. Besides the values presented in the boxplots, the global optimization methods had a few outliers which got stuck in a local optimum outside the presented window of figure 10. Because the x, y and z values are correlated, a source location is likely to either locate a source or to miss out on all 3 coordinates. The neural network appears to have found a pattern in the cross spectral matrix to estimate acoustic source locations, without getting stuck in a local optimum. However the neural network has a wider range in deviation than the global optimization methods. The second performance metric, processing time varies from fractions to seconds. The Neural network requires a significant time to train before being applicable. Once the training process is completed the neural network requires little processing time compared to the global optimization methods. The histograms in figure 11 present a deviation comparison of the simulated test data set. The narrow peaks on the top row shows the high accuracy of differential evolution. The bottom row presents the results of the neural network. Especially in radial direction z the accuracy appears to decrease.

4.2 Experimental results

The recordings described in section 3.2 were tested with each of the localization methods. Figure 12 presents the power spectral density of recording 1. The power spectral density visualizes peaks at each of the frequencies emitted by the omni-directional source. The strongest peak can be found at 3500 [Hz]. Besides the emitted frequencies, a wide variety of peaks at other frequencies can be found in the power spectral density.

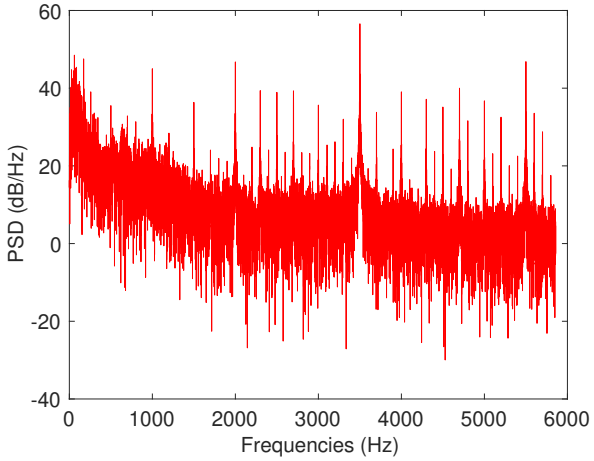


Figure 12: Power spectral density (PSD) of recording 1

Differential evolution and the butterfly optimization algorithm use equations 4, 5 and 6 to determine the energy values at the locations of interest. Each inversion is executed at a single frequency. Before feeding the cross spectral matrix to the global optimization algorithm, samples at a frequency 3500 [Hz] are selected. To visualize the acoustic environment a beamform plot is presented in figure 13. The plot consists of an 8 [m] by 8 [m] zoomed image at $z = 2.8702$ [m] (radial source distance). The beamform plot presents a mirrored source below the actual source. This is presumably a reflection from the omni-directional source on the grass surface. Figure 14 presents a beamform plot of recording 3, recording 3 was captured above a stone surface. The selected plane had equal dimensions as figure 13 at a radial distance of 4.2841 [m] from the microphone array.

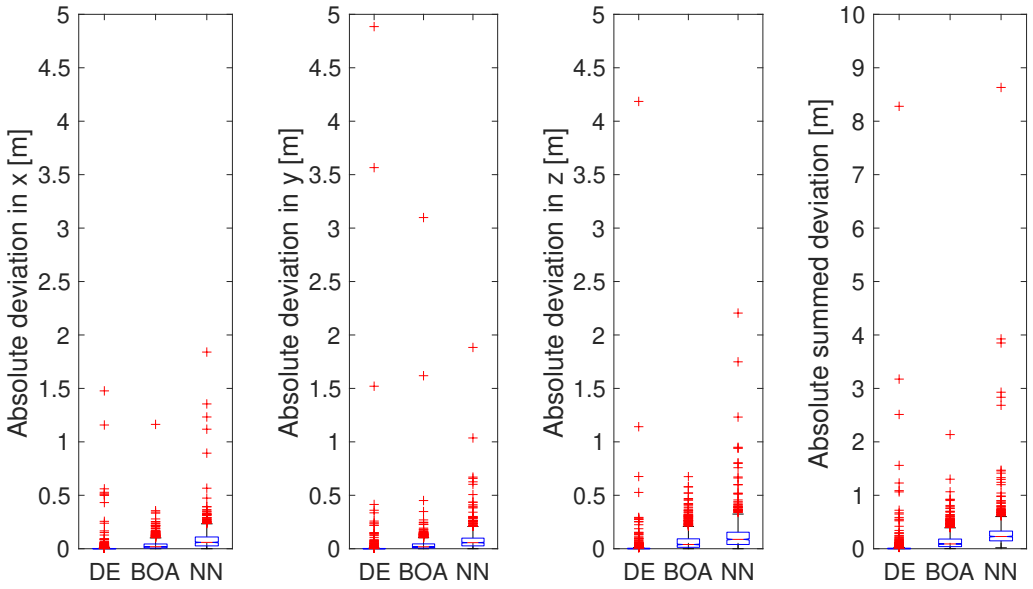


Figure 10: Absolute deviation between estimated and true source positions. Differential evolution is denoted by DE, the butterfly optimization algorithm by BOA and the neural network by NN

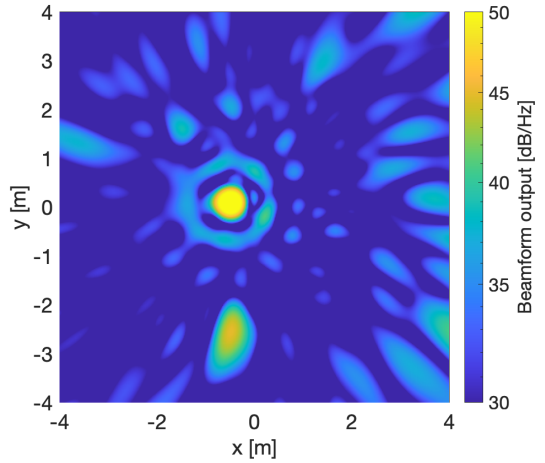


Figure 13: Beamform plot of recording 1 at 3500 [Hz]

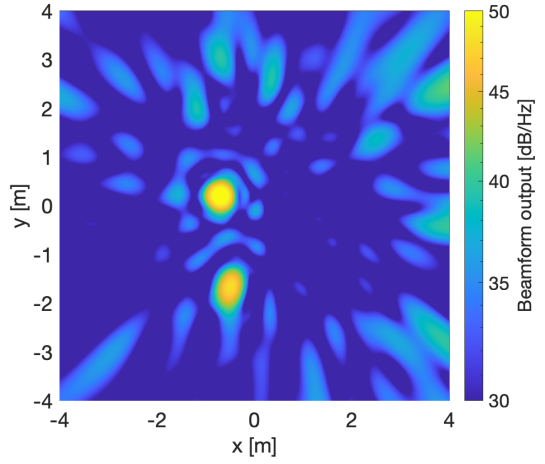


Figure 14: Beamform plot of recording 3 at 3500 [Hz]

Table 6, 7 and 8 present localization estimates of the recordings. The differential evolution algorithm requires the most time to process input signals. However the method obtains the most accurate results. The neural network produces estimates at a fraction of a second with the least accuracy. The butterfly optimization algorithm is a little less accurate than differential evolution at a quarter of the processing time.

	$x[m]$	$y[m]$	$z[m]$	Time $[s]$
Location	-0.4910	0.0972	2.9423	-
DE	-0.4910	0.0972	2.9423	6.654
BOA	-0.4792	0.0848	2.9018	1.325
NN	-0.33	0.313	27.528	0.08

Table 6: Locations and estimates of recording 1

	$x[m]$	$y[m]$	$z[m]$	Time $[s]$
Location	-0.6109	-0.6468	6.4229	-
DE	-0.6109	-0.6468	6.4229	6.196
BOA	-0.6095	-0.6455	6.4085	1.326
NN	-0.732	-0.494	9.632	0.0814

Table 7: Locations and estimates of recording 2

	$x[m]$	$y[m]$	$z[m]$	Time $[s]$
Location	-1.0117	0.2855	4.2959	-
DE	-1.0117	0.2855	4.2959	6.384
BOA	-1.0118	0.2849	4.2932	1.347
NN	-0.723	-0.536	8.386	0.0709

Table 8: Locations and estimates of recording 3

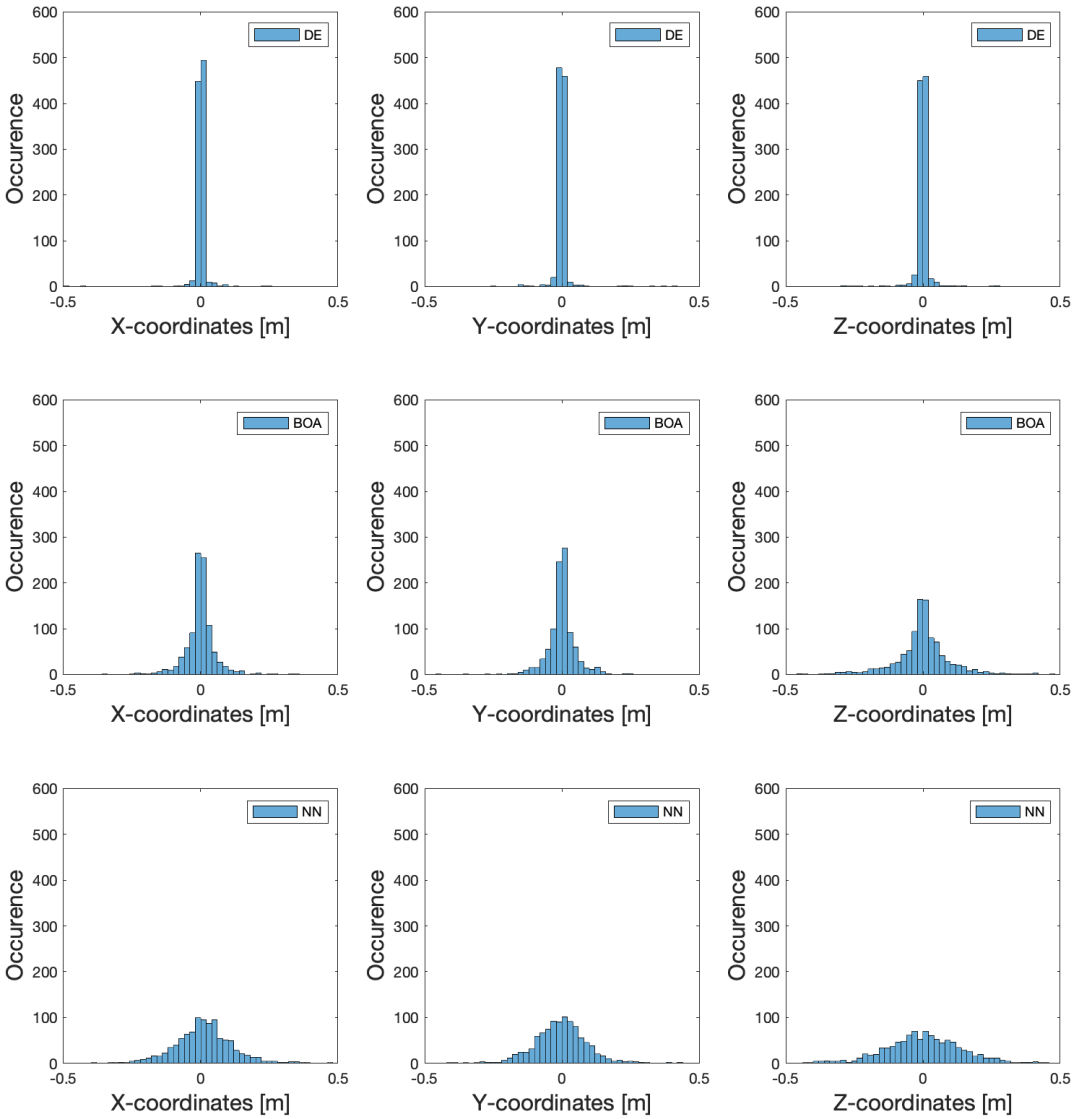


Figure 11: Histograms containing the simulation results on test data set. Differential evolution is denoted by (DE), the butterfly optimization algorithm by (BOA) and the neural network by (NN)

5 Discussion

During the research the search area is kept at constant format, a 3-dimensional cube with sides of 40 [m]. During the simulation stage the full extent of the search area was available to allocate acoustic sources. During the experiments a significant part of the search area was positioned under ground due to the microphone array located on a tripod facing sideways. Considering the application there is no reason to expect acoustics from underground, which only leaves room for error due to reflected signals. Still it is of interest to see if these reflections are found by the inversion. In future applications the array could be placed facing up to increase the effective search area.

During training and testing the neural network was slightly adjusted between the simulations and the experimental process. The 4096 inputs used during simulation were increased to 12544 inputs during the experimental phase. By investigating the impact of the input layer on the networks performance, the neural network could be optimized further. Besides optimizing the network, the network could be enhanced by increasing the size of the data set used during training. The neural network used during simulation was trained using a data set consisting of 840000 samples. After training, the network was tested with 1000 samples which were constructed according to the same process as the training data. During the experimental phase, the network was trained using 245000 samples, due to the increase in microphones and computational limitations. The network was trained with simulated data after which it was tested with recorded data. The difference of training and testing with data obtained from different methods could impact the results. However to obtain training samples working with synthetic data is a requirement. From the results is can been seen that this neural network is not yet capable of locating acoustic sources. Especially the deviation in radial direction z contributes to reduced accuracy.

The multi-tone signal emitted by the omnidirectional source created a signal at 200, 1000, 2000, 3500 and 5000 [Hz]. The 200 [Hz] signal has a large wavelength λ due to its low frequency. Large wavelengths are more challenging to capture with a microphone array in which the microphones are located close together. Therefore the higher frequencies are more accessible to use for localization. The global optimization methods used during this research base their search on a single chosen frequency. However by investigating combinations of multiple frequencies, more source characteristics might be retrieved. Besides, the global optimization methods outperform the neural network in accuracy under the conditions of this research, there is always a possible uncertainty. The chance for a global optimization method to end up in a local optimum can not be excluded.

6 Conclusion

In this paper three methods were presented to localize single acoustic sources in a 3-dimensional search area. The butterfly optimization algorithm and the multi-perceptron neural network are not yet used to localize 3-dimensional sources based on a microphone array. The methods were presented and tested in a simulation and experimental phase. After testing, the methods were evaluated based on accuracy and speed. Under the stated conditions, all three methods are capable of localizing single acoustic sources in the simulation process. The accuracy varies in which differential evolution appeared to be the most accurate in locating an acoustic source. Although the neural network showed to be capable of locating an acoustic source with less accuracy, it was able to do so within a fraction of a second. The capability of estimating acoustic source locations within a small time window could be a prerequisite considering extending the method to real time tracking. However sufficient accuracy remains of greater importance. During the experimental phase the global optimization methods proved to be able of locating acoustic sources. The neural network appears to find some patterns in the cross spectral matrix with far less accuracy. All three methods were capable of working with simulated and experimental microphone array input data. The butterfly optimization algorithm performs faster than differential evolution at the cost of obtaining slightly less accurate results. The neural network is able to estimate locations significantly faster than the global optimization methods, however with less accurate results. Depending on the application, one of the methods could be selected based on required accuracy or processing time. Still there is much more potential in these methods to investigate on what contributions they could bring to increase surveillance in sensitive areas.

References

[1] F. G. Martin Blass, *A real-time system for joint acoustic detection and localization.* Paris, France: in proceedings of the Quiet Drones symposium, e-symposium ed., 19-20 October 2020.

[2] A. Malgoezar, M. Snellen, P. Sijtsma, and D. Simons, "Improving beamforming by optimization of acoustic array microphone positions," in *Proceedings of the 6th Berlin Beamforming Conference*, p. 5, 2016.

[3] A. Malgoezar, M. Snellen, D. Simons, and P. Sijtsma, "Using global optimization methods for acoustic source localization," in *Proceedings of the 23rd International Congress on Sound and Vibration*, 2016.

[4] A. M. Malgoezar, M. Snellen, R. Merino-Martinez, D. G. Simons, and P. Sijtsma, "On the use of global optimization methods for acoustic source mapping," *The Journal of the Acoustical Society of America*, vol. 141, no. 1, pp. 453–465, 2017.

[5] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, pp. 715–734, 2019.

[6] A. Kujawski, G. Herold, and E. Sarradj, "A deep learning method for grid-free localization and quantification of sound sources," *The Journal of the Acoustical Society of America*, vol. 146, no. 3, pp. EL225–EL231, 2019.

[7] W. Ma and X. Liu, "Phased microphone array for sound source localization with deep learning," *Aerospace Systems*, vol. 2, no. 2, pp. 71–81, 2019.

[8] P. Castellini, N. Giulietti, N. Falcionelli, A. F. Dragoni, and P. Chiariotti, "A neural network based microphone array approach to grid-less noise source localization," *Applied Acoustics*, vol. 177, p. 107947, 2021.

[9] D. Simons, *Aircraft Noise and Emissions reader.* Delft University of Technology, 2019.

[10] M. Snellen and D. G. Simons, "An assessment of the performance of global optimization methods for geo-acoustic inversion," *Journal of Computational Acoustics*, vol. 16, no. 02, pp. 199–223, 2008.

[11] S. Arora and S. Singh, "An improved butterfly optimization algorithm with chaos," *Journal of Intelligent & Fuzzy Systems*, vol. 32, no. 1, pp. 1079–1088, 2017.

[12] S.-C. Wang, *Interdisciplinary computing in Java programming*, vol. 743. Springer Science & Business Media, 2003.

[13] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[17] J. Luo, Z. Xie, and M. Xie, "Interpolated dft algorithms with zero padding for classic windows," *Mechanical Systems and Signal Processing*, vol. 70, pp. 1011–1025, 2016.

[18] P. Sijtsma, *Phased array beamforming applied to wind tunnel and fly-over tests.* National Aerospace Laboratory NLR, 2010.

II

Literature study previously graded under AE4020

Acoustic localization based on optimization techniques and neural networks

R.D. Schoorl

Literature Survey

TU Delft Delft University of Technology

Aircraft Noise and Climate Effects

Acoustic localization based on optimization techniques and neural networks

Literature Survey

R.D. Schoorl

November 30, 2021

Faculty of Aerospace Engineering (AE) · Delft University of Technology

Abstract

The past years a significant increase in commercial availability and popularity of unmanned drones has taken place. This development can result in privacy violations, security concerns and noise emissions since no expert knowledge is required to control these devices. Conventional surveillance methods are not completely suited to detect and protect against drones [1]. The aim of this literature survey is to investigate current and potential methods in the field of acoustic localization. Current localization methods that have proven to be capable of finding acoustic sources are explained, while new methods are introduced. The most familiar method used in the field of acoustic imaging is Conventional Beamforming (CB). This robust method is explained and the areas were the methods falls short are noted. Global Optimization (GO) methods have proven to make up for some of the limitations of CB although not all shortages are resolved. Therefore this paper elaborates on GO methods by proposing the Butterfly Optimization Algorithm (BOA) to improve current applications. Besides CB and GO methods, new extensions of localization methods are discussed. The implementation of neural networks on CB maps are presented, and the trade-offs between resolution and accuracy are reviewed. The report finishes by summarizing acoustic localization methods and the benefits they can bring.

Table of Contents

List of Figures

"The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking."

— *Albert Einstein*

Chapter 1

Introduction

1-1 Social aspect

In the past decade, the development of drones has increased at a rapid pace. Drones are becoming more autonomous, and the applications are widely diverse. Conjointly to these advancements have been the increase in commercial availability and popularity. So far, there is no expert knowledge required to control small drones. Without mandatory expert knowledge and with the increase of drones, violations are waiting to happen. The absence of regulations could be expressed in privacy violations, security violations or public annoyance due to the noise produced. Current surveillance systems are not capable of resolving this issue yet. Therefore developing new methods could contribute to strengthen current surveillance methods. Popular drones in the smaller segments are often equipped with 4 or 6 engines. When drones are flying they produce noise. The noise characteristics depend on weight, size, rotors and flight manoeuvres. Microphones can record the produced noise. By recording drone noise and investigating this field, it could become a reality that drones will be tracked by the noise they produce [1].

1-2 Acoustic imaging

In the field of acoustic localization, multiple techniques are capable of locating acoustic sources. One of these methods is Conventional Beamforming (CB); this method can be characterized as simple and robust [4]. Like many other techniques, CB uses a microphone array as the input for localizing acoustic sources. However, being robust, CB does come with shortcomings. The number of grid points considered is limited due to the finite aperture of the array. In addition when using a pre-defined grid the number of points sampled is limited. The existence of sidelobes contributes to the possibility of finding sources that do not exist. The method is computational intensive and therefore unsuitable for real-time tracking. Some methods improve on some of these shortcomings. Global Optimization (GO) methods have proven to be successful in finding acoustic source locations [4]. GO can reduce the required amount of calculations due to smart searches within the solution space. However, they have not been able to provide sufficient speed for real-time tracking yet.

1-3 Research structure

In chapter 2 the working principles and limitations of a microphone are explained. Next in chapter 3 current methods for acoustic localization are explained. The discussed methods vary from CB to GO methods and the application of neural networks. Finally, chapter 4 summarizes the findings and proposes possible research objectives.

Chapter 2

Microphone array

A microphone array is based on the principle of multiple microphones working simultaneously. By bundling multiple microphones onto a two-dimensional plane, details on sound sources can be extracted. By spatially spreading the microphones over a two-dimensional plane, the distances between acoustic sources and microphones differ. The difference in distance results in arrival time variations; this characteristic can be exploited to obtain information on the acoustic source. A microphone array has the ability to measure multiple acoustic sources simultaneously.

2-1 Signal processing

Once the microphone array receives acoustic signals from a sound source, the microphones translate the pressure differences into voltage changes. For the second step, the system can be evaluated in the time-domain or in the frequency domain. The frequency domain is faster, however it is not suitable to track moving sources. The time-domain method is able to amplify preferred signals while mitigating unwanted signals. The signal's output can be Fourier transformed towards the spectral domain to present the spatial distribution and intensity of acoustic sources. The frequency-domain option is to transform the pressure signal obtained from the microphones to the spectral domain directly. A Cross Spectral Matrix (CSM) can be constructed for the frequencies of interest by summing the specified spectral levels together. A time delay in the time domain results in a phase delay in the frequency domain. Therefore the time is delayed in the time domain, and the phase is delayed in the frequency domain. Beneficial of working in the frequency domain is the opportunities that arise from working with the CSMs and the insights it can deliver in the acoustic sampled field. By applying beamforming on the received signal, a map can be created to present spatially acoustic intensities [5].

2-2 Limitations of microphone arrays

One of the advantages and disadvantages of microphones are the ability to pick up all sound instead of only recording the sound of interest. The Signal-to-Noise Ratio (SNR) can be used to quantify the relative importance these two contributions. The SNR displays the power of the signal of interest divided by the power of noise [6]. SNR increases intrinsically when using multiple microphones simultaneously [7]. Another limitation, the Rayleigh limit will be explained in subsection 2-2-1.

Figure 2-1 presents a microphone array. In this situation the microphone is situated facing an omni-directional sound source. The black foam dots protect the microphones as well as prevent the wind from interfering with the received signals.

Figure 2-1: Microphone array facing an omni-directional sound source

2-2-1 Rayleigh limit

When more than one acoustic source is situated close together, the sources could be recognized as a single source. Formula 2-1 represents the Rayleigh limit and describes the criteria for discriminating single acoustic sources [2]. The output of formula 2-1 will present the minimum required distance between acoustic sources to be able of discriminating them from each other.

$$R = \theta_B z_s = 1.22 \frac{c z_s}{f D} \tag{2-1}$$

R in formula 2-1 describes the Rayleigh limit in [m], θ_B is the beam width in $[rad]$, z_s [m] is the distance between the microphone array and the acoustic source and L is the microphone array aperture in [m]. Because the speed of sound c and the array aperture D are fixed in most cases, the frequency and source distance play a key role in discriminating acoustic sources.

Figure 2-2 presents two acoustic sources emitting sound at 2000 [Hz]. The acoustic sources on the images are simulated with varying distances in between. By reducing the separation between the acoustic sources, it becomes hard to impossible to discriminate the sources from each other.

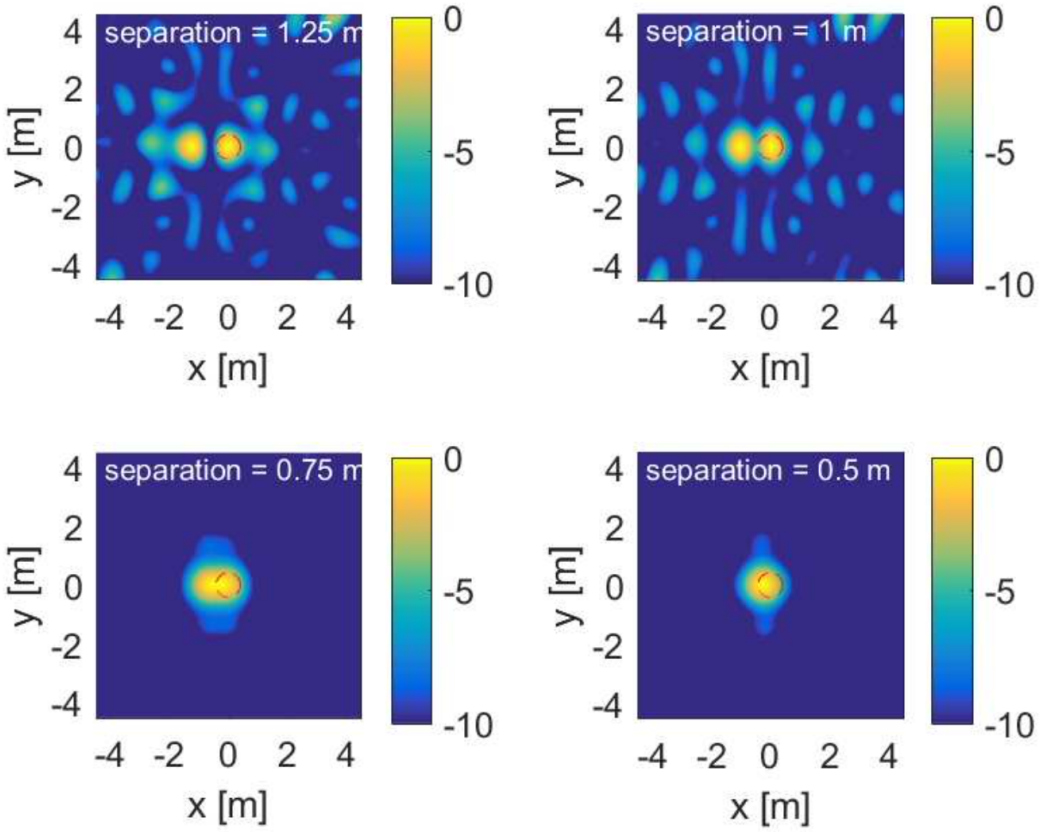


Figure 2-2: Two sources emitting sound at 2000[Hz], output is presented in [dB]. Image obtained from [2]

Chapter 3

Localization methods

Chapter 3 describes current localization methods and additional methods which could improve on localization performance. The methods are described based on the formulas and experiments found in literature. First Conventional Beamforming (CB) will be explained based on the underlying [8] [2]. Followed by global optimization methods in section 3-2 and the extension of neural networks in 3-3.

3-1 Conventional beamforming

The objective of CB is to localize and calculate the pressure levels of acoustic sources at specified grid-points. A microphone array is used to record sound waves in the time domain. The Fourier transformed pressure amplitudes are captured by vector $\mathbf{p_l}$ in 3-1, which represent the pressures measured by a set of microphones.

$$\mathbf{p_l} = \begin{bmatrix} p_1(f) \\ p_2(f) \\ \vdots \\ p_N(f) \end{bmatrix} \tag{3-1}$$

The information used for localization as introduced below are the pressure differences over the array. The phase of the source signal can therefore be unknown without hampering the localization. The next step is to obtain the Cross Spectral Matrix (CSM). This matrix is determined by multiplying $\mathbf{p_l}$ from 3-1 with the transpose of its complex conjugate. The asterisk sign in formula 3-2 and 3-3 denotes the complex conjugate transposed. The CSM can differ depending on the goal of the research and available data. Formula 3-2 or formula 3-3 can be used to determine the CSM. Depending on the application, one of the formula's can be chosen. Formula 3-2 can be used in case the recorded sound has a small time window. Formula 3-3 can be used if the recorded time window is larger than a snapshot. Beneficial of formula 3-3 is the ability to reduce the impact of background noise by combining average values of the recorded signal. This formula cuts the the measured signal into M samples per time block. In formula 3-3 the computed blocks have an overlap of 50%.

$$\mathbf{C} = \frac{1}{2}\mathbf{p_l}\mathbf{p_l}* \tag{3-2}$$

$$\mathbf{C} = \frac{1}{2(2L-1)} \sum_{l=1}^{2L-1} \mathbf{p_l}\mathbf{p_l}* \tag{3-3}$$

In formula 3-4 the distances between each microphone and grid points are determined. The n in formula 3-4 denotes the microphones number and j denotes the the grid points of interest.

$$r_{n,j} = \sqrt{(x_n - x_j)^2 + (y_n - y_j)^2 + (z_n - z_j)^2} \tag{3-4}$$

By inserting frequency f, speed of sound c and $r_{n,j}$ in formula 3-5 the steering vector $g_{n,j}$ can be determined. The steering vector corresponds to the expected phases over the array for each grid point j.

$$\mathbf{g_{n,j}} = \frac{e^{-2\pi i f(\frac{r_{n,j}}{c})}}{r_{n,j}} \tag{3-5}$$

$$B(x, y, z, f) = \frac{\mathbf{g}^* \mathbf{C} \mathbf{g}}{\|\mathbf{g}\|^{\mathbf{2}}} \tag{3-6}$$

The output $B(x, y, f)$ of formula 3-6 is called the beamform output. Presenting B for all grid points provides a source map. It quantifies how much variation of the CSM matches with the steering vector g. When measurements and steering vector $\mathbf{g}$ have a high level of resemblance, this indicates a high probability of having an acoustic source nearby. The output obtained from formula 3-6 is expressed in $[Pa^2]$ at a specific frequency f. The beamform output B can be converted to Sound Pressure Level (SPL) in Decibels (dBs) by filling in formula 3-7.

$$SPL = 10 \log(\frac{p_e^2}{p_{e0}^2}) \tag{3-7}$$

In which p_e^2 represents the outputs of $B_{x,y,f}$ and p_{e0}^2 the refrerence pressure. By calculating the SPL at each grid point and thereby knowing the pressures at each grid point, the source intensity can be expressed in dBs. The strongest source is the grid point which contains the highest value in dBs. Multiple sources that are simultaneously present can be found with CB. This characteristic specifies CB as a simple and robust method. By determining the SPL at each grid point, the method presents an exhaustive search and requires significant calculations [4]. A second drawback is a trade-off between grid size. A grid consisting of more grid points provide more insight. However, this does come at the cost of additional computation power. CB does not concern multiple potential acoustic sources when analyzing each grid point. Despite not accounting for multiple sources, the results of finding them are quite accurate. The method is based on the principle that every grid point has the ability to contain an acoustic source. Other methods like Global Optimization (GO) elaborated on in section 3-2 can improve by pinpointing the exact amount of sources prior to calculations. The formulas to calculate steering vector 3-5 and beamform output 3-6 shown above can vary. The combination of these factors can change between having accurate localization or to determine accurate source strength, This phenomenon will be elaborated on in subsection 3-3-1.

3-1-1 Conventional beamform plots

The microphones on the array capture pressure differences over time. By transforming the signal to the frequency domain the phase differences of the incoming signals can be compared

[9]. One of the first steps of the CB method is to define the area in which the acoustic source is expected. CB is very suitable to create 2-dimensional representations of acoustic sources. The figures 3-1 and 3-2 present two examples of CB plots. The simulated acoustic source in these plots is located at $x = 2[m]$, $y = 3[m]$ and $z = 5[m]$, noted by the red star in the figures. Figure 3-1 shows the beamform output in $[Pa^2]$ at height (z value) of $4[m]$ and figure 3-2 shows the beamform output in $[Pa^2]$ at a height of $5[m]$. By comparing both images, it can be seen that the red star is accurately found in figure 3-2. Besides the colored side axis of both figures present the beamform output. The beamform output of the captured source in figure 3-2 at a height of $5[m]$ is higher and therefore will be closer to the actual source.

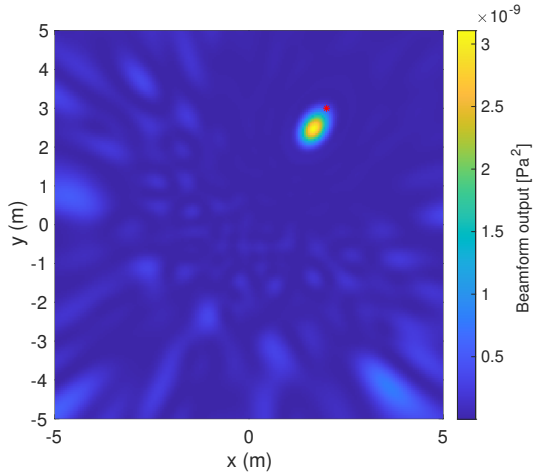


Figure 3-1: CB plot at 4 meter distance from simulated source at 3000hz

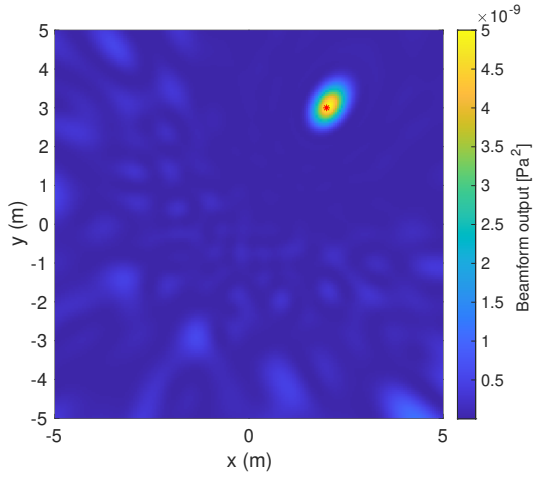


Figure 3-2: CB plot at 5 meter distance from simulated source at 3000hz

By creating multiple layers of 2-dimensional planes and placing them on top of each other a three dimensional image can be formed. Figure 3-3 shows a CB plot of a simulated acoustic source at $x = 2[m]$, $y = 3[m]$ and $z = 5[m]$ with a frequency of $2000[Hz]$. In figure 3-3 all beamform output values below the value of $5e - 11[Pa^2]$ are considered to be non existing, this is done to create a clear visualization of the more present areas of the beamform output.

Figure 3-3 shows the impact of side lobes and local optima besides the global optimum which represents the acoustic source. The formulas used to calculate the data in the image are focused on finding the correct source strength rather than have the highest localization accuracy. In figure 3-3 it can be seen that the combination of CB and a microphone array is more accurate at localizing an acoustic source in the x and y plane than in the z plane. This is a result of only having microphones in the x and y plane and not in the z plane.

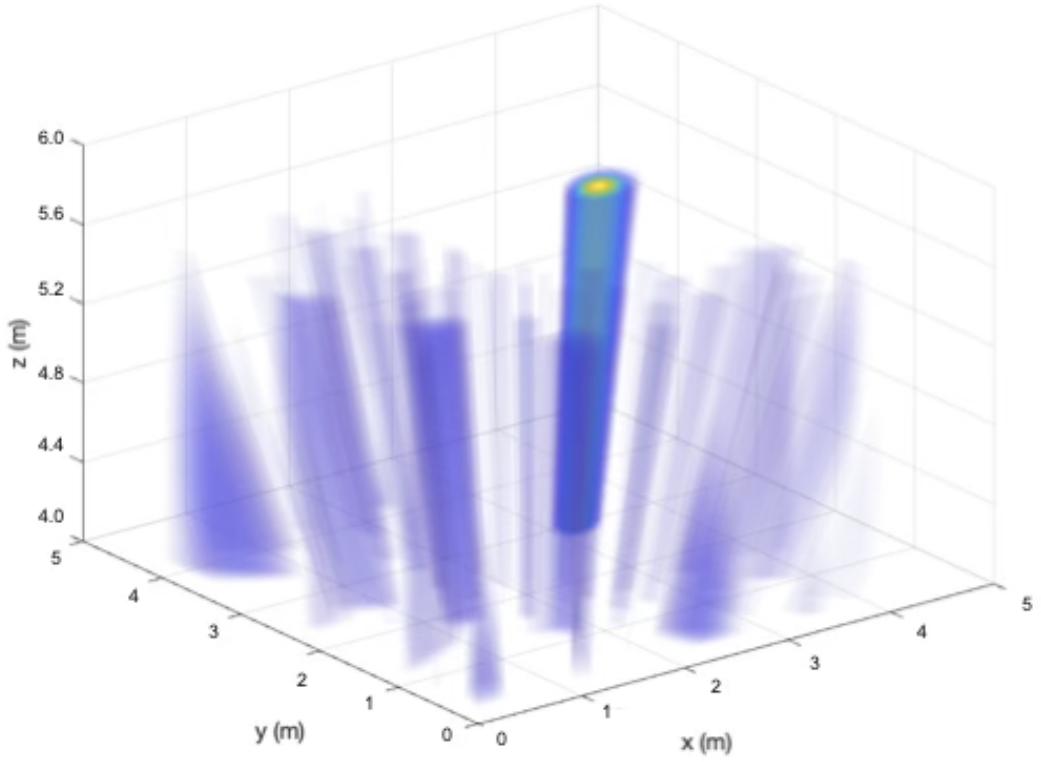


Figure 3-3: Three dimensional CB plot at $2000[Hz]$

3-2 Global optimization algorithms

As stated in section 3-1, CB is a simple and robust method to locate acoustic sources. A downside of CB is the required exhaustive search and being limited to searches in planes instead of grid-free searches. One of the challenges in finding the correct location of an acoustic source is caused by the presence of side lobes. Accessory to creating the main signal, side lobes are created. Side lobes exist at a lower intensity at different angle than the main acoustic signal. Side lobes could be considered local optima and the main signal could be considered the global optimum. By having global and local optima alongside each other, the possibility arises to find a local optimum when searching for the global optimum. Figure 3-4 present a two-dimensional intersection of a main lobe and side lobes.

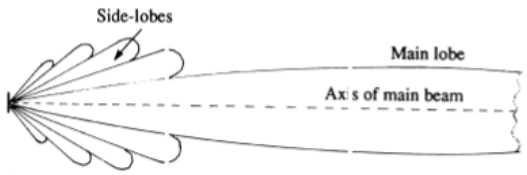


Figure 3-4: 2D visualization of a main lobe and side lobes. Image obtained from Arpa manual [3]

Nature is still a great source of inspiration in the process of finding and developing suitable solutions to real-world problems. A lot of GO methods are based on situations encountered in nature and it's ways to solve challenges. One of these challenges could be the search through three dimensional acoustics. At higher frequencies side lobes have a higher intensity, making the acoustic landscape more difficult for GO algorithms to find the global optimum. The paper of Malgoezar et al. [4] shows that it is possible to locate an acoustic source with help of a GO method. Beneficial of GO is the ability of locating the actual source location with fewer calculation than CB. Fewer calculations are required because the method does not have to calculate pressure intensities at every grid point. The use of GO methods does come with the risk of getting stuck in one of the local optima and thereby finding a faulty source location.

3-2-1 Energy functions

GO methods work by solving an objective function. During the solving process the algorithm strives to minimize or maximize this function depending on the application. This objective function is often referred to as the energy function [4]. The energy function can differ depending on the goal of the algorithm. Different goals could be focusing on finding accurate locations, accurate source intensity or searching for multiple acoustic sources. The energy function has several input parameters which are adapted during the iterations of the search. By simulating and experimenting, the quality of the energy function can be evaluated. Two examples of energy functions are the Bartlett energy function in formula 3-8 and the CSM energy function in formula 3-9 [4]. The summation in formula 3-9 is done over all $M * M$ elements of the matrices containing the differences between C_{meas} and $C_{model,g}$ at a specified frequency.

$$E_{Bartlett}(g) = \frac{y(g,\omega)^H C_{meas}(\omega) y(g,\omega)}{\| y(g,\omega) \|^2 \, tr(C_{meas}(\omega))} \tag{3-8}$$

$$E_{CSM}(g) = \sum([Re(C_{meas}) - Re(C_{model,g})]^2 + [Im(C_{meas}) - Im(C_{model,g})]^2) \tag{3-9}$$

As presented in the formulas, both energy functions have different objectives. The Bartlett energy function only considers phase differences in pressure obtained from the microphone array. The energy value of the Bartlett energy function is most optimal when the phase differences in pressure are fully in phase with the steering vector. The Bartlett energy function only searches for the location of acoustic sources. Whereas the CSM energy function considers location and source amplitude [10].

3-2-2 Simulated Annealing

Simulated Annealing (SA) is a method based on the cooling process of metal. The goal of the algorithm is to optimize a given objective function. During this process, the temperature changes from high to low. SA starts a high temperature, in this stage there is a significant chance of accepting solutions which are worse than the previous. When the temperature decreases the chance of accepting a solution which is worse than the previous solution decreases. All solutions are compared, and the beneficial steps are incorporated towards the final objective function [11]. SA uses randomly chosen initial values and finds the most suitable solutions based on trial and error. To obtain the best outcome, the process is executed multiple times while saving the best results. When using slight variations in temperature, the SA method has a certainty of finding the global optimum. The downside of working with slight temperature variations is the required increase in calculations. By increasing the amount of calculations, the required computations increase and thereby extending the computational time. Therefore SA becomes less suitable for fast searches [12].

3-2-3 Differential Evolution

The second optimization method discussed is Differential Evolution (DE). The method is based on the genetic algorithm and imitates the natural evolution of species. According to the method, promising solutions are considered superior and are more likely to reproduce than less promising solutions. Compared to the exhaustive search performed in CB, the DE algorithm creates the possibility of calculating more unknown parameters. In the research done by Malgoezar et al [4], the amount of sources are considered to be known beforehand, and no predefined grid is used, making this method grid-free. Beneficial of the GO algorithms is their ability to search large spaces of candidate solutions while having to make barely or no assumptions on the problem.

The DE algorithm begins with a random set of starting points. After the initial points are chosen, the algorithm makes new generations to improve the quality of the candidate solutions. In the process of developing a population, the DE algorithm crosses current generations with candidate solutions, in which the most suitable candidates or generations are preserved. This process is repeated to close in on the global optimum as accurate as possible [13][14]. When

searching for the global optimum, a possible danger is to get caught in a local optimum. To prevent the algorithm from getting stuck in local optima, some of the less suitable solutions have a probability of evolving as well. By doing so, the ability is created to escape local optima. The chances given to less suitable solutions to evolve decline as the generations progress [15]. The goal is to optimize the energy function and thereby finding the coordinates converging to the correct x,y and z parameters. The success of the DE algorithm depend on the setting parameters. To make sure the algorithm achieves its goal, some parameters have to be chosen beforehand. The size of the population, the multiplication factor, the crossover probability and the total number of generations play a crucial role in finding the global optimum.

3-2-4 Butterfly Optimization Algorithm

This subsection is based on the paper of Arora, and Singh [16]. This paper explains the concept of the Butterfly Optimization Algorithm (BOA) and compares the algorithm with different global optimization methods.

The BOA is based on characteristics of butterflies and mimics the behaviour of butterflies searching for food and mating partners. In both of these searches, scent plays a key role. In many applications, optimization algorithms are constructed with complex constraints and dispose only over a limited time to find the global optimum. Modern optimization methods like Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO) and many more have proven to increase performance with non-linearity and multi-modality [16] [17].

In the BOA each butterfly produces its own and unique scent. The butterflies scent is used to distinguish butterflies from each other. The idea of sensing is explained by Arora and Singh [16] based on sensor modality (c), stimulus intensity (I) and power exponent (a). The sensor modality captures the energy input obtained by the sensors. Modalities can be described as light, smell, sound or other stimuli. Formula 3-10 presents a simplification of butterflies ability to smell, in which h is the perceived magnitude of fragrance.

$$h = cI^a \tag{3-10}$$

In formula 3-10 h could be seen as relative because the ability to get smelled depends on the emitting and the receiving butterfly. When the value of a is chosen to be equal to 1, no absorption of the scent occurs, and each butterfly smells the particular butterfly with the same intensity. This situation could only be present in an ideal environment. In case $a = 0$, no scent is emitted at all, and none of the other butterflies will be able to pick up the smell. Next, the value of c is essential, as this significantly impacts the algorithm's convergence speed. In this concept, butterflies with less intensity are attracted towards butterflies that emit a scent with a higher intensity. In the BOA algorithm, there are three rules that clarify butterflies behaviour. The first rule is that every butterfly emits a scent, giving other butterflies the ability to become attracted. The second rule implies that every butterfly will move either at random or towards other butterflies based on the received scent. Third, each butterfly's intensity of scent is affected by the obtained values from the solution space.

The BOA is constructed to work in 3 phases. The phases are initialization phase, the iteration phase and the final phase. During the initialization phase the objective of the algorithm is

noted, the setting parameters are defined and the limitations of the search area are set. Next the initial butterfly population is created at random chosen locations in the search area. The amount of butterflies does not vary during the iterations. In the second phase the butterflies will move through the search area at each iteration. Each iteration causes the butterflies to move towards new locations and adapt their fitness values according to formula 3-10. The search strategy is divided into two categories, the global search and the local search. Formula 3-11 represents steps of butterflies who move towards the global optimum. Parameter k^* indicates the butterfly which found the most suitable objective at that specific iteration. In Formula 3-11, the x_i^t represents a solution vector for the ith butterfly in iteration t. The values for r is chosen at random between $[0, 1]$.

$$\mathbf{x_i^{t+1}} = \mathbf{x_i^t} + (r^2 \times \mathbf{k^*} - \mathbf{x_i^t}) \times h_i \tag{3-11}$$

$$\mathbf{x_i^{t+1}} = \mathbf{x_i^t} + (r^2 \times \mathbf{x_j^t} - \mathbf{x_k^t}) \times h_i \tag{3-12}$$

In formula 3-12, $\mathbf{x_j^t}$ and $\mathbf{x_k^t}$ present the solution vectors of butterflies j and k at iteration t. If the values of $\mathbf{x_j^t}$ and $\mathbf{x_k^t}$ are equal or very similar, the butterfly will converge towards a local optimum. Because butterflies are subjected to environmental vectors like wind, rain or predators, they can be split into fractions searching for global and local optima. A probability of $\mathbf{p}$ is used for butterflies to switch between a common global search or an extensive local search. The algorithm is finished after one of the stopping criteria is met. Possible stopping criteria could be: maximum computer power used, the preset limit of iterations is reached, the iterations are not improving fast enough, or a specified error rate was found. Once a chosen criterion is met and the algorithm has been executed, the fittest solution can be presented.

3-2-5 Bidirectional optimization

In the article of Ahandani et al. [18] variations are made based on the DE algorithm. One of the modifications is to let the algorithm work in multiple directions, the Bidirectional Differential Evolution (BDE) method. If moving forwards in the solution space does not result in better objectives, a high probability holds that backward movement will. The modified DE algorithm appeared to have a higher success rate than the original DE algorithm; however, the modified algorithm does consume more time than the original algorithm. In the research of Sharma et al. [19], the idea is discussed and applied to change search directions if an objective function is becoming less attractive. The modification was applied on the BOA as well and thereby calling it the Bidirectional Butterfly Optimization Algorithm (BBOA). The BBOA is characterized as more greedy than the BOA. According to [19] the BBOA algorithm helps to escape local optima and accelerates the convergence rate. Whether the application is suitable of tracking sound sources and benefits, the algorithm has yet to be proven.

3-2-6 Particle Swarm Optimization

In 1995 Kennedy and Eberhart proposed the optimization technique called PSO [20]. The algorithm was based on the social behaviour of bird flocking, fish schooling and animals which operate in a swarm. Swarm animals move in a cooperative manner; they learn from each other and share their own experiences with others. By operating in this manner, the PSO shares characteristics with evolutionary algorithms. This characteristic is expressed by using a large swarm that simultaneously investigates the solution space in search of the global optimum. In the research of trying to mimic animal social behaviour with computers, five basic principles were set by Mark Millonas [21]. These principles could be defined as swarm intelligence.

- Proximity, The group should be able to make basic time and space calculations to respond directly to environmental impulses.
- Quality, the swarm should be able to estimate the quality of food or whether a location is safe.
- Diverse response, the swarm should have multiple ways of communicating with each other as this is a beneficial property in a changing environment.
- Stability, the swarm should not change its behaviour mode based on each fluctuation as this costs energy and might not be a valuable investment.
- Adaptability, when it appears the change of investing in different behaviour, the group should have the ability to switch.

The stability and the adaptivity principle are somewhat conflicting; however, they do not rule each other out. The five principles describe the main behaviour properties of an artificial life system. During its iterations, the swarm is continuously searching the solution space for optimal solutions. At each moment in the search for the optimal values, each member of the swarm can memorize the optimal locations of itself and is aware of the other swarm members optimal locations. After each iteration is executed, all information is combined to determine the next point of interest. All swarm members are constantly changing their states until the global optimum is found [22]. The initial PSO algorithm appeared not to be very efficient. The initial algorithm could not find the optimum if it was not on its preset path. This situation is unlikely to appear, and therefore the PSO algorithm was adapted. In 1998 Shi and Eberhart [23] updated the algorithm without adding much complexity. In this update, the velocity vectors were adapted to contain weights. The update proved to have a better performance. This performance was quantified by comparing time spent finding the optimum, number of iterations and chance of finding the optimum.

The most significant differences between PSO algorithm and BOA is their origin. The PSO algorithm is based on coordinated group behaviour of birds flocking, fish schooling or animals that operate in a swarm. The BOA is based on social individuals looking for food or other butterflies. Another difference between the algorithms is the availability of information. In the PSO algorithm, all swarm members are assumed to know all other members' information. In the BOA algorithm, not all information is known to all butterflies and therefore, some information could be lost. So far, no research has been conducted to determine the impact of this loss [16].

3-3 Combining neural networks with conventional beamforming

The following method combines neural networks with conventional beamforming. According to CB, a pre-defined grid is created in which pressure or phase differences are presented. Low frequencies result in relatively large wavelengths [λ] as follows from formula 3-13 [2]. The spatial resolution for low frequencies is restricted. This is a result of required dimensions of the microphone array combined with large wavelengths.

$$\lambda = \frac{c}{f} \tag{3-13}$$

The application of using deep learning on image recognition has made rapid development in the past years. By allowing an algorithm to learn, it obtains possibilities to circumvent mathematical calculations. Reiter and Bell [24] published a paper in which they show the possibility of estimating acoustic sources on photo-acoustic images. This achievement was accomplished with the help of neural networks. The goal of a neural network was to determine the location of the acoustic source.

In the method described by Adam Kujawski et al. [25], CB images are combined with neural networks to locate the sound source on CB images. The location coordinates in the two-dimensional grid planes are related. Convolutional Neural Networks (CNN) are capable of working with spatially correlated and multidimensional data. These properties make CNN suitable to image processing tasks. Therefore CNN architecture appears to be a suitable candidate to locate acoustic sources on CB images. CNNs are build up with a feed-forward architecture consisting of multiple layers, often referred to as kernels. Those properties create the ability to generalize given inputs and identify objects efficiently. The optimization of the model applies supervised learning. In the training stage of the algorithm the model is given a set of test data. The test data exists of input data and known output data. By feeding the input data and expecting the output data, the parameters of the algorithm can be tuned in order to improve the algorithm. One of the strengths of CNN is the implementation of weight sharing into the design; by doing so, the number of parameters that require training is substantially reduced which leads to improved generalization. Additionally, the classification stage has a build-in extraction stage, which both require a learning process. In the extraction stage significant features from the raw data are automatically found. Third, the implementation of CNNs is less complicated compared to artificial neural networks [26].

In the article of Kujawski [25] use was made of the residual network (ResNet), a derivation of CNNs. The Residual Network (ResNet) was chosen due to its improvements in classification and regression tasks and its ability to work with small-sized images. CB images in the work of Kujawski have a low resolution of $51*51$ [25]). CNN is often used for images consisting of more pixels and therefore not every architecture is suitable to work with images of a lower resolution. The difference between the ResNet and CNNs is the presence of added shortcuts. The shortcut sends the input data parallel through the filters alongside the standard processing path called identity mapping. Both images are combined at the end of the filters, thereby learning the ResNet only to map residual data between the input and output. The ResNet thereby solved the problem of increasing training errors and improving accuracy [27]. The work of Kujawski [25] shows it is possible to identify acoustic sources with sub-grid precision. At large Helmholtz numbers, the error in distance increases slightly, and the best localization

performance is obtained at the lower range Helmholtz numbers. This phenomenon applies for every localization method

3-3-1 Selecting the correct steering vector

As stated earlier, CB does not account for multiple sources during its iterations. However, the CSM is impacted by all acoustic sources around. When multiple sources emit sound, the sound waves can interfere and create a different pressure field then a single source would create individually. As a result of multiple acoustic sources emitting simultaneously, the CSM is influenced as well. The sound waves created by each monopole can be considered separate until the waves meet and the collide together. If the receiver of the sound is sufficiently far away, the sound can be interpreted as originating from a single source. The knowledge of knowing the amount of sources can be helpful choosing a suitable energy function.

In the research of Sarradj [28] none of the investigated steering vectors are suitable to provide both accurate localization and correct source strength. The steering vectors used are described as trade-offs by either choosing a steering vector that is superior at locating sound sources or choosing a steering vector that is superior at finding the source strength. The figures 3-5 and 3-6 present plots of a simulated acoustic sources. In these plots the received intensity is plotted against an increasing value of X. The simulated source in figure 3-5 is located at point [0, 0, 0] and the source in figure 3-6 is located at [2, 3, 5] both in meters. In both plots the Y and Z axis are fixed. These images present the impact of different steering vectors in combination with beamform output B. In the top half of the images the intensity is shown in $[Pa^2]$ and the bottom half of the images this value is normalized, to emphasis relative behaviour.

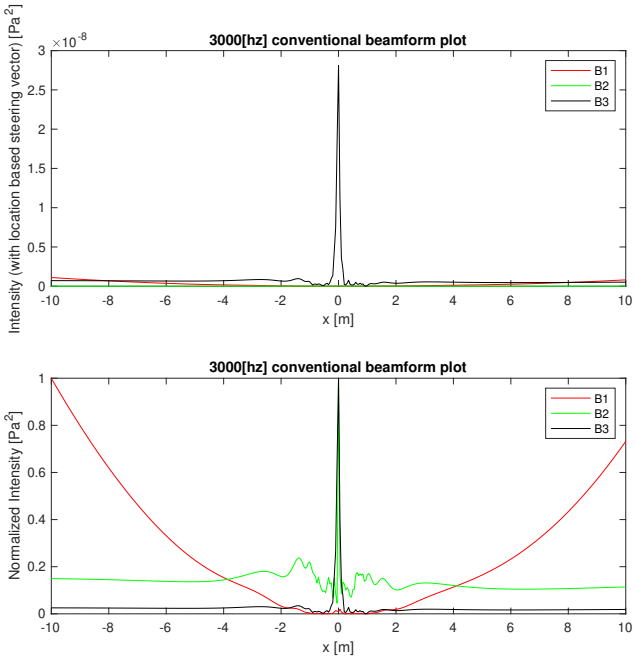


Figure 3-5: CB plot of intensity vs X coordinate, with fixed Y and Z coordinates at location X=0 [m], Y=0 [m] and Z=0 [m]

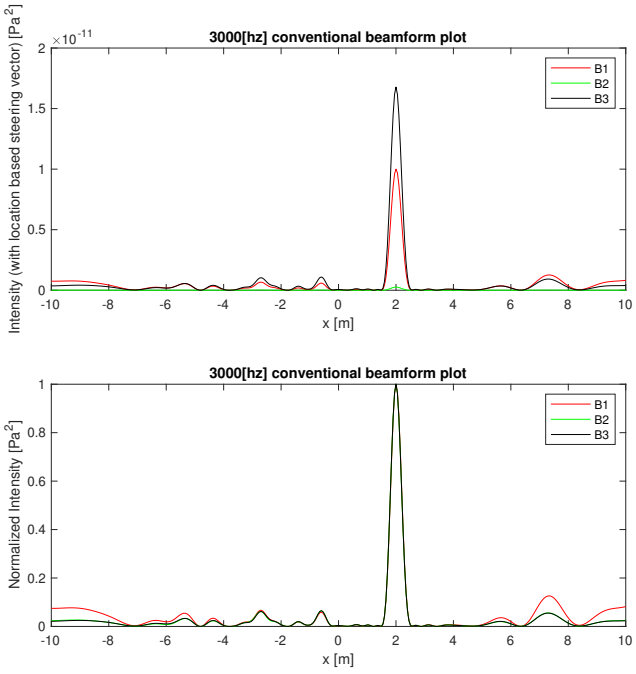


Figure 3-6: CB plot of intensity vs X coordinate, with fixed Y and Z coordinates at location X=2 [m], Y=3 [m] and Z=5 [m]

In the figures 3-5 and 3-6 different steering factors and beamform outputs are used to emphasize their impact. The formulas 3-14 until 3-19 are used to create images 3-5 and 3-6. Image 3-5 shows the impact of combining the value r in the denominator of the steering vector g_1 and having the steering vector to the power 4 in the denominator of B_1. When distances become further from the source, B_1 increases significantly for values of r which are further away from the actual source. This causes incorrect results and could impact GO methods by creating global optima which do not exist. The combination of steering vector g_2 with output B_2 has the ability of locating the acoustic source most accurate however this combination is less suitable in finding the correct source intensity. The third combination of steering vector g_3 with output B_3 is most suitable in finding the correct intensity of the acoustic source however this method is less accurate in finding the correct location. This leads to a trade-off between finding the correct source intensity or accurate localization, as discussed by Sarradj in [28]. This problem could be solved by first localizing the acoustic source and once the location is found to perform an additional step to obtain the correct source intensity with steering vector 3-18 and output formula 3-19. The definitions of parameters and formulas are explained in section 3-1.

$$\mathbf{g_1} = \frac{e^{\frac{-2\pi i f r}{c}}}{r} \tag{3-14}$$

$$\mathbf{B_1} = \frac{\mathbf{g'Cg}}{||\mathbf{g}||^4} \tag{3-15}$$

$$\mathbf{g_2} = e^{\frac{-2\pi i f r}{c}} \tag{3-16}$$

$$\mathbf{B_2} = \frac{\mathbf{g'Cg}}{||\mathbf{g}||^4} \tag{3-17}$$

$$\mathbf{g_3} = \frac{e^{\frac{-2\pi i f r}{c}}}{r} \tag{3-18}$$

$$\mathbf{B_3} = \frac{\mathbf{g'Cg}}{||\mathbf{g}||^2} \tag{3-19}$$

Chapter 4

Summary

This chapter will summarize the findings from the literature survey. The synthesis between topics are tightened, and possible contributions are stated. In the second chapter the working principle and limitations of a microphone array are reviewed. The third chapter explains techniques which can be used to localize acoustic sources. The discussed techniques are Conventional Beamforming (CB), Global Optimization (GO) and a trade-off is made in the neural network's method of Sarradj. The third chapter finishes by elaborating on variation of steering vectors and their impact.

CB described in chapter 3 has the capability of locating acoustic sources in a three dimensional grid. This localization process is based on the principle that each location can be characterized by having an unique phase gradient. The output matrix of CB presents scale of how much the phase gradient of the Cross Spectral Matrix (CSM) matches with the steering vector g. When the beamform output $B(x, y, f)$ and steering vector g have a high level of resemblance, this indicates to a high probability of correctly locating an acoustic source. CB does not account for multiple acoustic sources when analyzing each grid point. The method is based on the principle of every grid point having the ability of housing an acoustic source. Other methods like GO profit from pinpointing the exact amount of sources prior to calculation, this grants the ability to account for changes in the CSM. This can be achieved by adapting the energy function. Limitations of CB are the mandatory use of a predefined grid and therefore having to make the redundant calculations. By determining the pressure differences in each grid point, the required computational capacity increases, therefore this method is labeled as an exhaustive search and is considered a time consuming approach. GO methods like Differential Evolution (DE) and Simulated Annealing (SA) approach the optimization problem without a predefined grid. These methods offer a lot of potential and are proven to be capable of finding a global optimum. The BOA found in section 3-2-4 is compared to many other global optimization techniques and appears to have the promising results. Therefore it would be interesting to apply the BOA to acoustic source localization to see if the search performance can be improved.

Another interesting technique is the extension of neural networks to CB. In section 3-3 is explained how neural network can be combined with CB. The method applies neural network imaging on CB maps in order to locate the acoustic source. This process is executed after CB has taken place and a source map has been generated. Beneficial of this method is the ability to locate acoustic sources with sub-grid accuracy. The limitation of using Convolutional Neural Networks (CNN) in combination with CB, is the required use of CB maps which are considered computational exhaustive. Interesting would be to investigate if reducing the grid resolution on the CB images could still result in accurate results obtained from the neural network. This could lower the total computation time if less redundant calculations are requisite.

Last the microphone array is discussed in chapter 2. The chapter discusses the spatial constraints due to the Rayleigh limit. Combined with the CB method of section 3-1 the working principle of a microphone array is explained. Interesting would be to find out what the correlations could be between the raw data obtained from the microphone array and known source locations. If so, what parameters would be interesting to investigate. This approach might result in possibilities of bypassing searching algorithms by recognition of patterns in the obtained raw data. Key role will be to have sufficient data to verified potential finds.

Bibliography

[1] F. G. Martin Blass, *A real-time system for joint acoustic detection and localization*. Paris, France: in proceedings of the Quiet Drones symposium, e-symposium ed., 19-20 October 2020.

[2] D. Simons, *Aircraft Noise and Emissions reader*. Delft University of Technology, 2019.

[3] B. D. Alan Bole and A. Wall, *Radar and Arpa Manual, Radar and target tracking for professional mariners, yachtsmen and users of marine radar*. Amsterdam: Elsevier Butterworth-Heinemann, second ed., 2005.

[4] A. M. Malgoezar, M. Snellen, R. Merino-Martinez, D. G. Simons, and P. Sijtsma, "On the use of global optimization methods for acoustic source mapping," *The Journal of the Acoustical Society of America*, vol. 141, no. 1, pp. 453–465, 2017.

[5] L. Brusniak, J. Underbrink, and R. Stoker, "Acoustic imaging of aircraft noise sources using large aperture phased arrays," in *12th AIAA/CEAS Aeroacoustics Conference (27th AIAA Aeroacoustics Conference)*, p. 2715, 2006.

[6] D. H. Johnson, "Signal-to-noise ratio," *Scholarpedia*, vol. 1, no. 12, p. 2088, 2006.

[7] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing*, vol. 1. Springer Science & Business Media, 2008.

[8] P. Sijtsma, *Phased array beamforming applied to wind tunnel and fly-over tests*. National Aerospace Laboratory NLR, 2010.

[9] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE assp magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[10] D. F. Gingras and P. Gerstoft, "Inversion for geometric and geoacoustic parameters in shallow water: Experimental results," *The Journal of the Acoustical Society of America*, vol. 97, no. 6, pp. 3589–3598, 1995.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[12] H.-L. Shieh, C.-C. Kuo, and C.-M. Chiang, "Modified particle swarm optimization algorithm with simulated annealing behavior and its numerical verification," *Applied Mathematics and Computation*, vol. 218, no. 8, pp. 4365–4383, 2011.

[13] M. Snellen and D. G. Simons, "An assessment of the performance of global optimization methods for geo-acoustic inversion," *Journal of Computational Acoustics*, vol. 16, no. 02, pp. 199–223, 2008.

[14] R. Storn, "On the usage of differential evolution for function optimization," in *Proceedings of North American Fuzzy Information Processing*, pp. 519–523, IEEE, 1996.

[15] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[16] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, pp. 715–734, 2019.

[17] G. C. Onwubolu and B. Babu, *New optimization techniques in engineering*, vol. 141. Springer, 2013.

[18] M. A. Ahandani, N. P. Shirjoposh, and R. Banimahd, "Three modified versions of differential evolution algorithm for continuous optimization," *Soft Computing*, vol. 15, no. 4, pp. 803–830, 2010.

[19] T. K. Sharma, A. K. Sahoo, and P. Goyal, "Bidirectional butterfly optimization algorithm and engineering applications," *Materials Today: Proceedings*, vol. 34, pp. 736–741, 2021.

[20] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

[21] M. M. Millonas, "Swarms, phase transitions, and collective intelligence," *arXiv preprint adap-org/9306002*, 1993.

[22] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, 2018.

[23] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*, pp. 69–73, IEEE, 1998.

[24] A. Reiter and M. A. L. Bell, "A machine learning approach to identifying point source locations in photoacoustic data," in *Photons Plus Ultrasound: Imaging and Sensing 2017*, vol. 10064, p. 100643J, International Society for Optics and Photonics, 2017.

[25] A. Kujawski, G. Herold, and E. Sarradj, "A deep learning method for grid-free localization and quantification of sound sources," *The Journal of the Acoustical Society of America*, vol. 146, no. 3, pp. EL225–EL231, 2019.

[26] S. Indolia, A. K. Goswami, S. Mishra, and P. Asopa, "Conceptual understanding of convolutional neural network-a deep learning approach," *Procedia computer science*, vol. 132, pp. 679–688, 2018.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[28] E. Sarradj, "Three-dimensional acoustic source mapping with different beamforming steering vector formulations," *Advances in Acoustics and Vibration*, vol. 2012, 2012.

Glossary

List of Acronyms

ABC	Artificial Bee Colony
BBOA	Bidirectional Butterfly Optimization Algorithm
BDE	Bidirectional Differential Evolution
BOA	Butterfly Optimization Algorithm
CB	Conventional Beamforming
CNN	Convolutional Neural Networks
CSM	Cross Spectral Matrix
dB	Decibel
DE	Differential Evolution
GO	Global Optimization
PSO	Particle Swarm Optimization
ResNet	Residual Network
SA	Simulated Annealing
SNR	Signal-to-Noise Ratio
SPL	Sound Pressure Level

III

Supporting work

1

Introduction

The following chapters will provide background information on the theories used in the scientific paper of part I. Section 2.1 will explain the principles of beamforming, followed by signal processing, which explains design choices made during the research. The process of emitting an acoustic tone to localizing an acoustic source is covered. Background knowledge and parameter choices of the global optimization methods are explained in section 3. Section 4 elaborates on the fundamentals of neural networks and substantiated design choices. The final chapters will contain the appendices with additional beamform plots and results.

2

Microphone array

2.1. Beamforming

Beamforming is based on the comparison between input signals of multiple microphones at slightly different positions. By combining all input signals, information on source location and strength can be recovered. Figure 2.1 presents the microphone array used during the experiments.

Figure 2.1: Microphone array with omni-directional source setup

During the recordings each microphone records sound pressures over a specified time interval K. The microphones convert pressure differences to a digital input signal. The input signals in the time domain are noted by x_k with discrete time steps t_k, $k = 0, ..., K-1$. To analyze the input signal in the frequency domain, signal x_k is Fourier transformed to the frequency domain by formula 2.1.

$$X_m = \Delta t \sum_{k=0}^{K-1} x_k e^{-2\pi i t_k f_m} \tag{2.1}$$

The parameter Δt used in formula 2.1 presents the sample distance in time. The sample distance is determined by dividing 1 with sampling frequency f_s and is presented by formula 2.2.

$$\Delta t = \frac{1}{f_s} \tag{2.2}$$

The signal X_m presents the Fourier transformed signal x_k at discrete frequencies m, $m = 1, ..., M$. By Fourier transforming the input signals of each microphone, a matrix can be determined. The matrix will exist of N rows and M columns, in which N denotes the number of microphones and M the number of discrete frequencies. Processing of input signals is explained in more detail in section 2.5.
Based on the frequency of interest, column b can be selected from matrix X_m. The column presents an input signal at a single discrete frequency from each of the microphones. In formula 2.3, the cross spectral matrix C is created. The cross spectral matrix can be determined by multiplying single column b of matrix X_m with its transposed complex conjugate.

$$C = X_{m,b} X^*_{m,b} \tag{2.3}$$

In conventional beamforming a scan plane is determined at a fixed distance z from the microphone array. The plane can be divided into grid points as presented in figure 2.2.

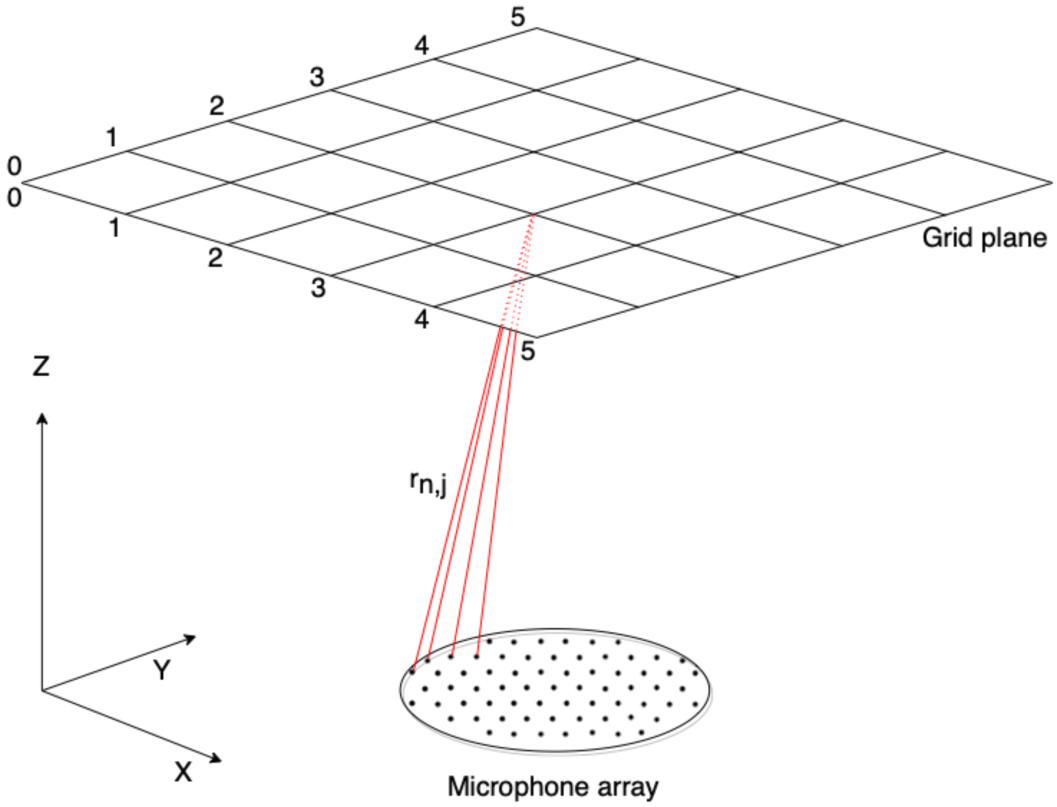


Figure 2.2: Conventional beamforming, consisting of a microphones (black dots) and grid plane

The locations of the microphones are key knowledge in the beamforming process. Formula 2.4 can be used to determine the distance between the grid points and the microphones. The red lines in figure 2.2 visualize a few of the distance values of $r_{n,j}$

$$r_{n,j} = \sqrt{(x_n - x_j)^2 + (y_n - y_j)^2 + (z_n - z_j)^2} \tag{2.4}$$

The values of $r_{n,j}$ can be implemented in the so called steering vector described by formula 2.5 in which the individual vector elements $g_n(\zeta_j, f_m)$ are determined.

$$g_n(\zeta_j, f_m) = \frac{e^{-2\pi i f_m \frac{r_{n,j}}{c}}}{r_{n,j}} \tag{2.5}$$

The steering vector contains phase information of the different microphone inputs at specified frequency f_m. Beamform output $B(\zeta_j, f_m)$ in $[Pa^2]$ can be determined according to formula 2.6, by combining all elements in steering vector $\mathbf{g_n}$ with cross spectral matrix C.

$$B(\zeta_j, f_m) = \frac{\mathbf{g_n}^* \mathbf{C} \mathbf{g_n}}{||\mathbf{g_n}||^2} \tag{2.6}$$

By determining the acoustic source strength at each of the grid points, a map can be created presenting the acoustic landscape at a fixed plane above the microphone array. By determining the beamform outputs at all grid points, the grid point with the highest beamform output is likely to accommodate an acoustic source. By increasing the number of grid points in the scan plane, the beamform image becomes more accurate [5][6]. Figure 2.3 presents a beamform plot at a radial distance of 5 [m] at frequency 3000 [Hz]. The beamform plot demonstrates a single acoustic source located at $x = 2$ [m] and $y = 3$ [m].

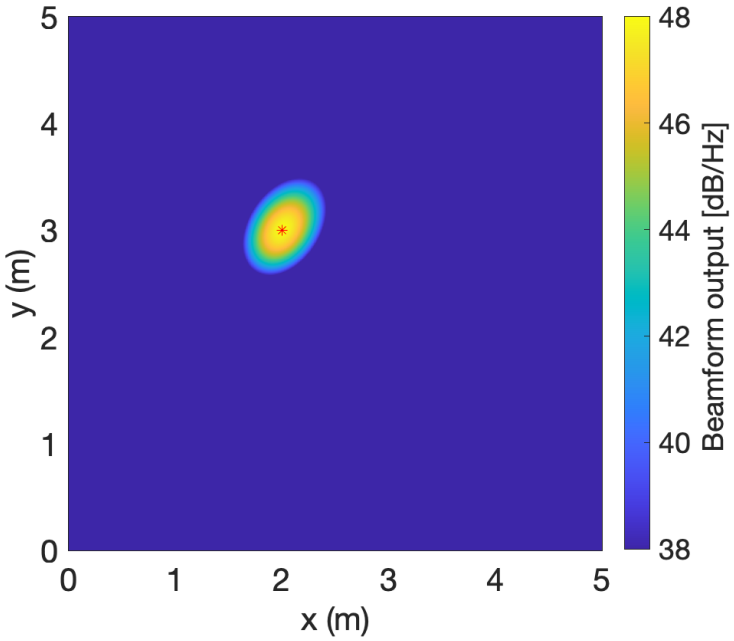


Figure 2.3: Beamform plot at 3000 [Hz], the red cross denotes the actual source position

2.2. Side lobes

When using a microphone array to record acoustic sources, additional non-existing sources can be found. The main lobe of the signal corresponds to the highest energy level. However other locations can present energy levels without accommodating an acoustic source, these energy levels are called side lobes. The presence of side lobes is undesirable as they form local optima at locations which do not accommodate actual sources. A solution to reduce the impact of side lobes is to increase the density of microphones on the microphone array. However the positions of microphones are based on the application of the array. The distance between the microphones relate to the frequencies of interest. High frequencies require a high density of microphones and low frequencies allow more spacing between microphones but require a large aperture. Another method to reduce the impact of side lobes is windowing. By applying a window function, for example Hanning or Hamming to the signal, the influence of side lobes can be reduced. However when applying Hanning or Hamming to the input signals, the outer microphones are significantly surpressed. In the post-processing stage, the method of incoherent averaging can also be applied to reduce the impact of side lobes. Formula 2.7 presents incoherent averaging.

$$B_{incoh}(\zeta_j) = \frac{1}{M_f} \sum_{m=1}^{M_f} B(\zeta_j, f_m) \tag{2.7}$$

First multiple beamform plots are required at multiple frequencies. Next, the beamform outputs are summed and divided by the total number of frequencies M_f. The side lobes located at different positions are frequency dependent and will be reduced while the strong main lobe can be preserved.

2.3. Steering vector

In section 2.1 the principles of beamforming are explained. Based on the principle of beamforming, the energy function can be implemented. The energy function applies beamforming only at specific locations without being subjected to a beamform grid. The global optimization methods investigated in this research make use of the energy function to determine the energy value at their current positions within the search area. The combination of the steering vector and beamform output formulas determine the energy function. Multiple energy functions are available and each has different advantages and disadvantages compared to each other. In the research of Malgoezar et al.[4], 2 different energy functions are compared. The Bartlett and the cross spectral matrix energy function of which the cross spectral matrix energy function includes source strength estimation. The Bartlett energy function is presented by 2.8

$$E_{Bartlett} = \frac{\mathbf{g}(f)^T C_{meas}(f)\mathbf{g}(f)}{||\mathbf{g}(f)||^2 tr(C_{meas}(f))} \tag{2.8}$$

The vector $\mathbf{g}(f)$ denotes the steering vector at frequency f. The matrix $C_{meas}(f)$ denotes the measured cross spectral matrix. The trace function in the denominator specifies the values located on the diagonal of the cross spectral matrix. The diagonal of the cross spectral matrix defines the phase differences between single microphone inputs, which should be non-exiting. Therefore the diagonal is assumed to be contaminated and the diagonal values are set to zero. The cross spectral matrix energy function is described by 2.9.

$$E_{CSM} = \sum_{q=1}^{N}\sum_{r=1}^{N}\left\{[Re(C_{meas\ q,r}) - Re(C_{model\ q,r})]^2 + [Im(C_{meas\ q,r}) - Im(C_{model\ q,r})]^2\right\} \tag{2.9}$$

The energy function used in formula 2.9 compares the values of the cross spectral matrix with the values of a modeled cross spectral matrix and squares them. The summation is taken over all the elements present in the cross spectral matrices.

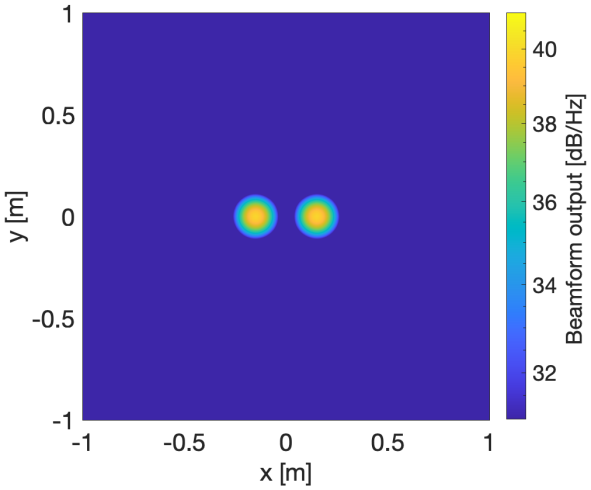
During the research of Malgoezar et al. both energy functions were tested on multiple scenarios. The functions were tested on simulated and experimental data. The simulation contained searches for single or multiple mono-pole sources emitting at a single frequency. The experiments were conducted in an an-echoic room with speakers serving as the acoustic source. Both of the energy functions were combined with differential evolution to locate the acoustic source within the search area. The Bartlett energy function has the detriment of not determining the source strength. However the localization performance was better compared to the cross spectral matrix energy function. Therefore this research proceeds using the Bartlett energy function [4].

2.4. Rayleight limit

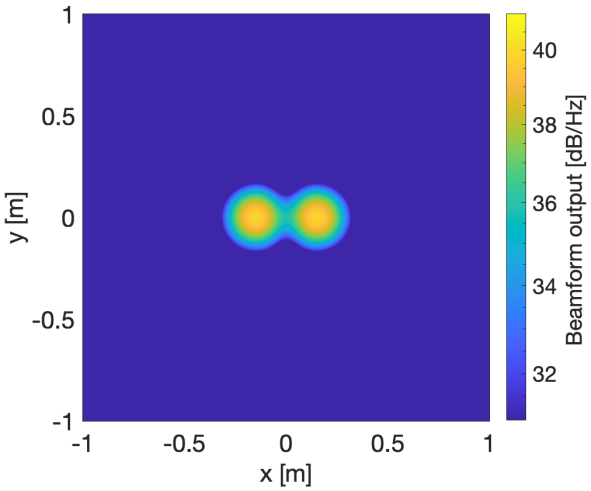
When two acoustic sources are located close together, spatial resolution can determine whether its possible to distinguish the acoustic sources from each other. If the main lobe of one source overlaps the main lobe of another source, the sources can be seen as a single source. Spatial resolution also called the Rayleigh criterion, defines the limit at which acoustic sources can be seen as separate sources. Formula 2.10 defines the spatial resolution in a scan plane.

$$\theta_B z_s = 1.22\frac{cz_s}{fL} \tag{2.10}$$

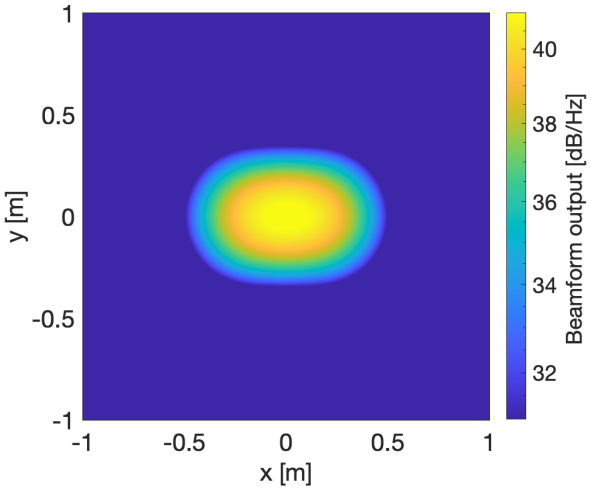
The angular resolution can be defined by θ_B, parameter z_s denotes the distance between the microphone array and the scan plane, constant c the speed of sound, L the array aperture and f the investigated frequency. The spatial resolution can also be noted as the Rayleigh distance according to $R = \theta_B z_x$. The criterion holds under the assumption that the provided sources are located close to the origin of (0,0). Figure 2.4 presents an example of two simulated acoustic sources at multiple frequencies. The acoustic sources are located at $(-0.15, 0, 2)$ and $(0.15, 0, 2)$ $[m]$ with a source strength of $7 * 10^{-5}$ $[Pa^2]$. The frequencies presented in the beamform plots emit tones at 3000, 2000 and 1000 $[Hz]$, while the other values remain constant. By decreasing the frequency and thereby increasing wavelength λ, the sources are overlapping. Overlapping of acoustic sources is undesired as the sources can not be distinguished anymore [4] [6] [8].



(a) Two sources at 3000Hz



(b) Two sources at 2000Hz



(c) Two sources at 1000Hz

Figure 2.4: Two acoustic sources at multiple frequencies

2.5. Signal processing

During the research, the global optimization methods and neural networks were trained and tested on simulated and experimental data. The recordings present an acoustic source emitting a single or multi-tone signal. The signals are recorded or simulated in the time domain, in which they are captured by microphones. More specifically, during the research 2 microphone arrays are used consisting of 64 and 112 microphones. A microphone works according to the principle of measuring pressure over time and converting these differences to voltage changes. Once the experimental or simulated recordings were obtained, the signals were converted from the time domain to the frequency domain. By investigating signals in the frequency domain, the ability arises to search at specific frequencies within the investigated signals. The investigated frequencies can characterise specific sources. The fast Fourier transform was used to convert the signals from the time domain to the frequency domain. The fast Fourier transform is a computational faster form of the discrete Fourier transform and often used to process signals. The Fourier transform can be determined using formula 2.1 and at discrete time samples presented in formula 2.2. Formula 2.11 presents the frequency resolution, in which N denotes the amount of specified samples.

$$\Delta f = \frac{f_s}{N} \tag{2.11}$$

The frequency resolution captures the step size between investigated frequencies and denotes the smallest frequency changes which can be detected. When evaluating a signal, the frequency resolution should be sufficiently small to capture enough details on the investigated signal.
However there are many additions which can be combined with the Fourier transform to obtain more desirable results. A signal can be zero padded. Zero padding implies adding zeros to the signal before feeding the signal to the Fourier transform. By adding more zeros to the signal the frequency steps decrease, which results in better visualisation when presenting the signal without altering signal properties. Another reason to apply zero padding is speed. The fast Fourier transform can be computed faster when the amount of samples is a factor of 2, therefore an investigated signal can be supplemented with zeros to obtain a power of 2 samples. A third method which can be applied to process signals combined with the Fourier transform is windowing. By multiplying the signal with a windowing function, signal characteristics can be highlighted. Windowing can minimize error sources which are present in the signal, for instance the impact of side lobes can be reduced. During the research a Hanning window and zero-padding were applied before converting the signal to the frequency domain [6].

3

Global optimization methods

The first 2 methods investigated were global optimization methods. The first method considered was differential evolution and the second method the butterfly optimization algorithm. Both meta-heuristic algorithms aim to find the global optimum within the search area. While doing so the methods make use of the energy function to quantify the fitness of the potential source locations. In section 2.3 is determined to use the Bartlett energy function, due to its capabilities of accurate source localization. However global optimization methods have the ability of easily altering energy functions according to the application. The energy function could be changed to focus on finding accurate source strengths or localizing multiple sources.

3.1. Algorithm tuning

During the simulation and experimental phase both algorithms were set to have an equal amount of forward calculations. A forward calculation can be noted by each time the energy value is obtained from the energy function. Both of the global optimization methods posses setting parameters. These parameters are set by increasing the amount of simulations and monitoring the performance of the algorithms. First a single parameter is optimized, by changing only one setting parameter while the other setting parameters remain constant. After the most suitable setting parameter is found, this parameters is temporary fixed. This process is repeated for each of the setting parameters in search for the optimal setting parameters. During the investigation on simulated and experimental data the search area remained constant. The search area consisted of a 3-dimensional squared cube with sides of 40 [m] each. The search area is presented by figure 3.1

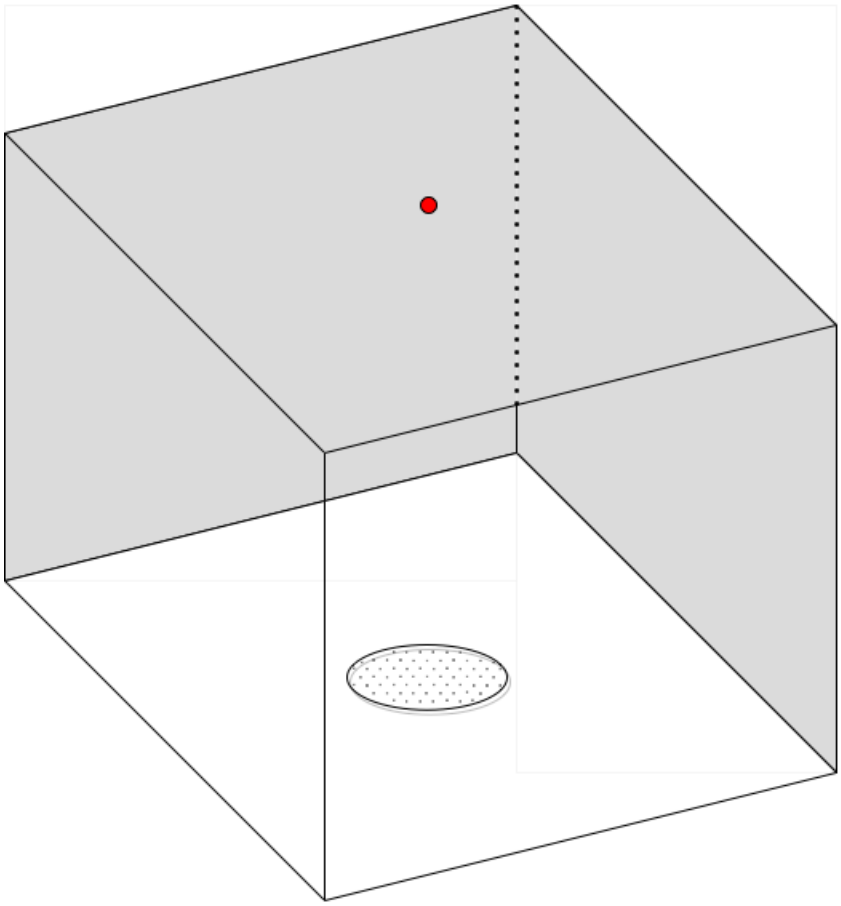


Figure 3.1: Simulated and experimental search area, the red dot denotes an acoustic source

4

Neural networks

The technology of deep learning dates back until the 1940s. The method appears to be new due to its variety in names and unpopularity throughout the years. The name changes relate to the different researchers implementing their ideas and perspectives. The method of deep learning is inspired by the brain however not all researches acknowledge this resemblance. The applications could better be interpreted as modelling specific biological functions. Current applications of deep learning appear to be more focused on creating general frameworks than the relation to its biological inspiration. Some researchers do not express their concern on the connection with neural science and obtain their inspiration from mathematical fundamentals instead.
In a feed-forward neural network the inputs flows through the network without connecting the output back to the network. When a feedback connection is added to a feed-forward network, the network becomes a recurrent neural network. During the research, use was made of a feed-forward multi-layer perceptron neural network. Beneficial of this type of network is its ability to estimate output values. Estimating output values is a requirement considering the localization of acoustic sources.

4.1. Layers & activation function

A neural network consists of multiple layers. The first layer is denoted as the input layer, followed by a single or multiple hidden layers and the final layer is the output layer. In literature there are multiple interpretations on whether to count the input layer, as no mathematical changes occur within the input layer. Each layer contains nodes, the number of nodes in the input layer is determined by the number of input elements. The number of nodes in the output layer is defined by the desired output. The number of input nodes used during this research was bounded by the number of elements in the cross spectral matrix. The cross spectral matrix contains information on phase relations between microphone inputs, on which location estimates will be based on. The output was specified by x, y and z coordinate estimates of potential acoustic sources. The amount of layers in a neural network can be described by the depth and the amount of nodes in a layer can be denoted by the width. By increasing the number of nodes and layers, the neural network is able to resolve more complex problems. However by increasing the number of nodes and layers, the network requires additional training due to the increase in trainable parameters. During the research use is made of fully connected networks. A fully connected network implies, that the nodes in each layer are connected to all of the nodes in its neighbouring layers. Figure 4.1 presents an example of the fully connected network used during the research.
Each layer in the neural network contains an activation function. The activation function defines the output of a node considering its inputs. Most activation functions contain a simple mathematical function which decides whether to suppress node inputs or feed the inputs through. The activation function enables a neural network to solve non-linear problems. Without non-linear activation functions, the neural network would become a linear regression problem solver. During this research two types of layers are considered, the rectified linear unit and a linear layer. The rectified linear unit is an example of a non-linear activation function which is often used as a default choice in hidden layers. Equation 4.1 and figure 4.2 present the rectified linear unit. The value of x denotes the input value of the node.

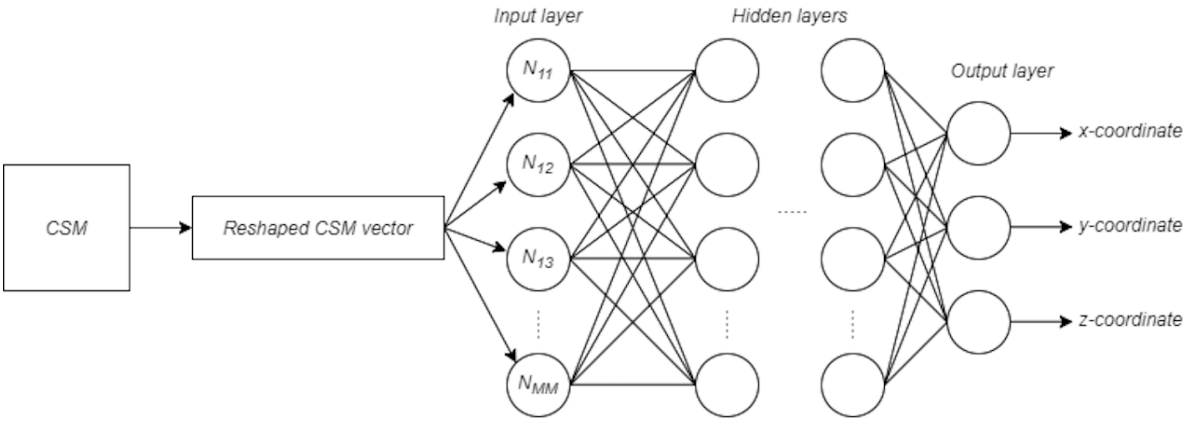


Figure 4.1: Multi-layer perceptron neural network

$$f(x) = max(0, x) \tag{4.1}$$

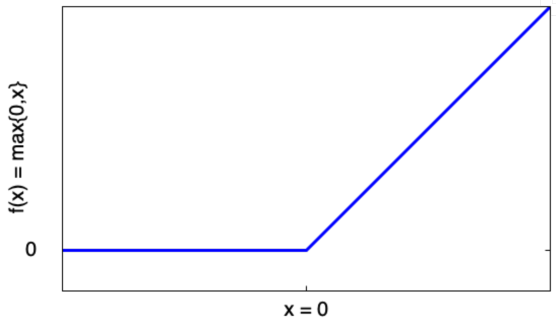


Figure 4.2: Rectified linear units (ReLU) activation function

It can be seen, that the rectified linear unit activation function disregards input values below 0 and only accepts values above 0. When an input value is equal or lower than 0, the nodes output is 0. Besides the rectified linear unit there is a wide variety of activation functions which can be assigned to hidden layers. The second activation function considered is a linear function. Choosing a linear activation function as the output layer is a common design choice in a multi-layer perceptron network. The linear activation function creates the ability for output values to obtain positive and negative values [3] [2]. The linear activation function is presented by formula 4.2 and figure 4.3.

$$f(x) = x \tag{4.2}$$

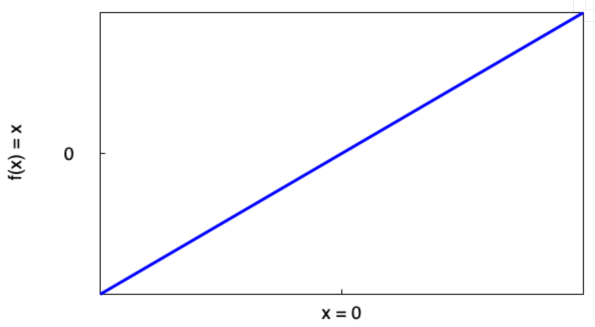


Figure 4.3: Linear activation function

4.2. Optimizers

Another important design parameter is the optimizer. Optimizers are algorithms with the capability of changing weights and in some cases the learning rate as well. The goal of the optimizer is to reduce the values of the loss function as much as possible. The loss function denotes the difference between estimates and actual outputs. Explanation on the loss function is elaborated on in 4.3.
There is a wide variety of optimizers, each comes with its own advantages and disadvantages. During the research two optimizers were tested, the Adam optimizer and the stochastic gradient descent. Adam is short for adaptive moment estimation. Currently the Adam optimizer is a popular choice due to its good and fast results. However a lot of researcher still apply the stochastic gradient descent by stating, that the stochastic gradient descent finds more optimal solutions at longer training time. The stochastic gradient descent originates from the well-known gradient descent method. During this research both of these optimizers were tested. The stochastic gradient descent appeared to perform better during the simulation process. Therefore the decision was made to proceed this research with the stochastic gradient descent optimizer [3] [7].

4.3. Training

One of the most characteristic trades of the neural network is their capability of finding patterns in data based on learning. By training a neural network, relations can be found between input and output data. Once the connection between input and output is found, the method has the ability of rapidly estimating new output values. Usually the data sets used in the training process of neural networks are split up in a training set, a validation set and a test set. The training set, the largest set is used to train the network. Next the validation set is used to quantify the training progress. At each cycle the entire training and validation data set is fed to the neural network, this is noted as an epoch. The training is finished when the preset amount of epochs is achieved. Between each of the epochs the order of the samples within the data set is shuffled. A sample is denoted by an input with corresponding output. The samples are shuffled to train the neural network in a general manner and prevent the network of getting accustomed to the order of the samples. Once the training is completed, the unseen test data set is fed to the network to check the final performance of the network. This learning process is considered supervised learning, as the network is trained with matched input and output data [2].
During training the samples of the data set are fed to the neural network. Before feeding the data to the network, the data set is split into multiple batches. A batch can be denoted as a part of the complete data set. At the end of a single batch moving through the network, the coordinate estimates are compared with the actual source coordinates and the link weights are updated. When the number of samples in the data set is not an exact multiple of the batch size, the last batch is filled with the residual samples.
The learning rate specifies the step size at which an optimizer searches towards the global optimum. Some optimizers have the ability to change the learning rate during the training process. Figure 4.4 presents a 2-dimensional comparison between different learning rates.

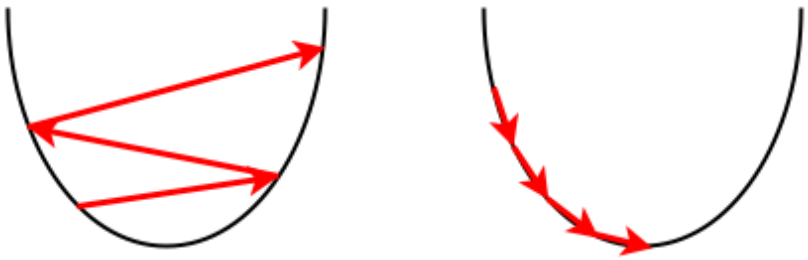


Figure 4.4: Difference in learning rate, large learning rate (left) and small learning rate (right)

By choosing a large learning rate the neural network is able to learn fast. However when choosing a large learning rate the possibility exists of stepping over the global optimum. Choosing a learning rate to small guarantees at finding an optimum, however this does not necessarily has to be the global optimum. Therefore the size of the learning rate can have a significant influence on the performance of the algorithm.
The loss functions can be used to quantify the difference between the model estimates and the actual values. Based on the output of the loss functions, the optimizer can adapt the link weights within the network. During the research two loss functions are tested. The considered loss functions are the mean absolute error

and the mean squared error. The mean absolute error is presented by formula 4.3 and the mean squared error is presented by 4.4.

$$Mean\ absolute\ error = \frac{1}{L}\sum_{l=1}^{L}|e_l| \tag{4.3}$$

$$Mean\ squared\ error = \frac{1}{L}\sum_{l=1}^{L}(e_l)^2 \tag{4.4}$$

The value e_l presents the difference between the actual and estimated output values [3]. The training was based on a systematic approach, in which model parameters were adapted based on results. The model architecture used by Castellini et al. in [1] was used for inspiration of the initial approach. From there a variation of design parameters were changed and compared. Table 4.1 presents the nodes, layers and activation functions used during simulation.

Layer	Neurons	Activation function	Type
1	4096	-	Input
2	400	ReLU	Fully connected
3	200	ReLU	Fully connected
4	50	ReLU	Fully connected
5	20	ReLU	Fully connected
6	3	Linear	Fully connected

Table 4.1: Neural network architecture used during simulation

The choice of model parameters are presented in table 4.2. These design parameters appeared to be most successful during tuning of the network.

Design parameter	Chosen parameter
Optimizer	Stochastic gradient descent
Loss function	Mean absolute error
Batch size	32
Learning rate	0.01

Table 4.2: Selected design parameters

Bibliography

[1] Paolo Castellini, Nicola Giulietti, Nicola Falcionelli, Aldo Franco Dragoni, and Paolo Chiariotti. A neural network based microphone array approach to grid-less noise source localization. *Applied Acoustics*, 177: 107947, 2021.

[2] Ivan Nunes Da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, and Silas Franco dos Reis Alves. Artificial neural network architectures and training processes. In *Artificial neural networks*, pages 21–28. Springer, 2017.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[4] Anwar MN Malgoezar, Mirjam Snellen, Roberto Merino-Martinez, Dick G Simons, and Pieter Sijtsma. On the use of global optimization methods for acoustic source mapping. *The Journal of the Acoustical Society of America*, 141(1):453–465, 2017.

[5] Pieter Sijtsma. *Phased array beamforming applied to wind tunnel and fly-over tests*. National Aerospace Laboratory NLR, 2010.

[6] Dick Simons. *Aircraft Noise and Emissions reader*. Delft University of Technology, 2019.

[7] S Vani and TV Madhusudhana Rao. An experimental approach towards the performance assessment of various optimizers on convolutional neural network. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 331–336. IEEE, 2019.

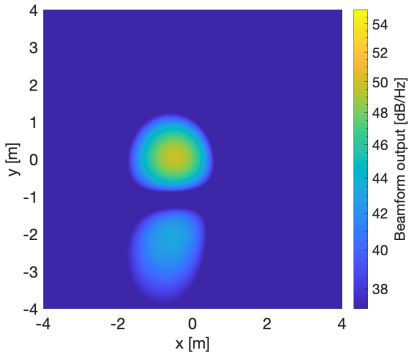
[8] B. von de Hoff. Assessment of the use of global optimisation techniques for aircraft noise source identification. *TU Delft repository*, 2020.

[9] Peter Wellig, Peter Speirs, Christof Schuepbach, Roland Oechslin, Matthias Renker, Urs Boeniger, and Hans Pratisto. Radar systems and challenges for c-uav. In *2018 19th International Radar Symposium (IRS)*, pages 1–8. IEEE, 2018.

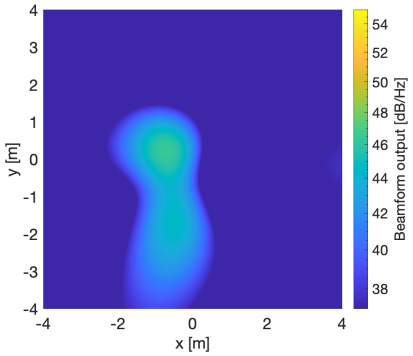
5

Appendix 1

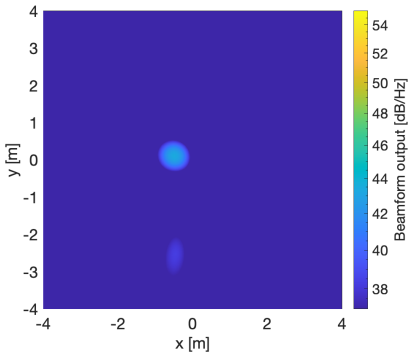
Figure 5.1 and figure 5.2 present beamform plots at multiple frequencies. The beamform plots are presented at frequencies 1000, 2000, 3500 and 5000 [Hz]. All images present the same acoustic source at different locations above grass and stone surfaces. Figure 5.1 presents the recording at $t = 4$ seconds. Figure 5.2 presents the recording at $t = 7$ seconds. Although the beamform plots are very similar, minor changes can be seen by investigating the side lobes. By investigating multiple selections of time within single recordings, potential unwanted noise inputs can be highlighted.



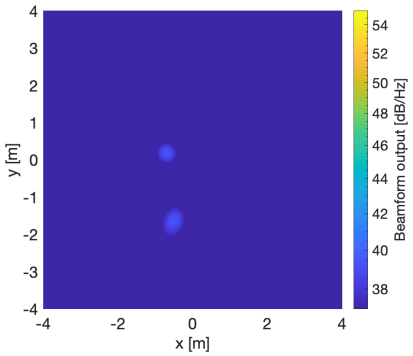
(a) Recording 1 above a grass surface at 1000 (Hz)



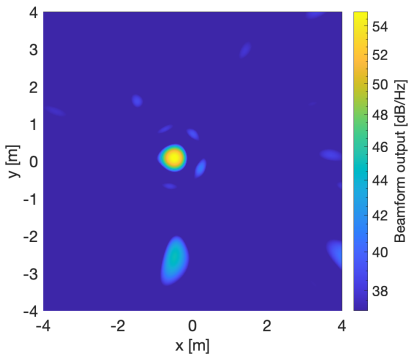
(b) Recording 3 above a stone surface at 1000 (Hz)



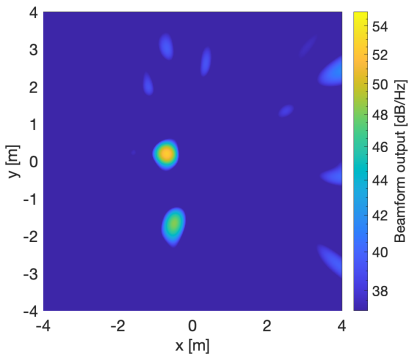
(c) Recording 1 above a grass surface at 2000 (Hz)



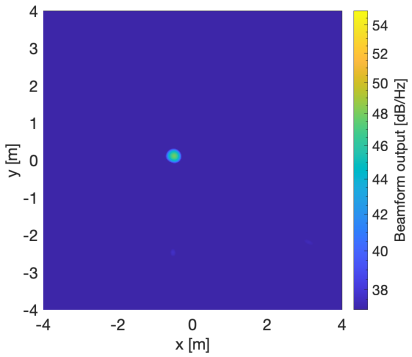
(d) Recording 3 above a stone surface at 2000 (Hz)



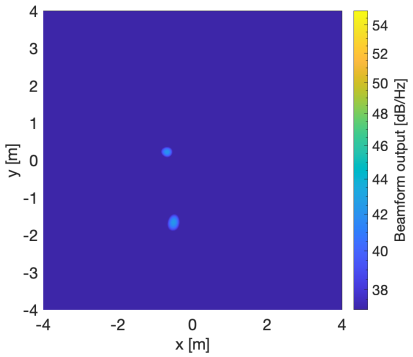
(e) Recording 1 above a grass surface at 3500 (Hz)



(f) Recording 3 above a stone surface at 3500 (Hz)

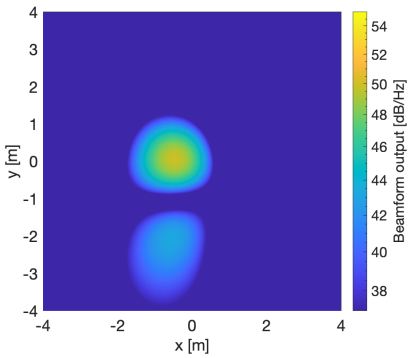


(g) Recording 1 above a grass surface at 5000 (Hz)

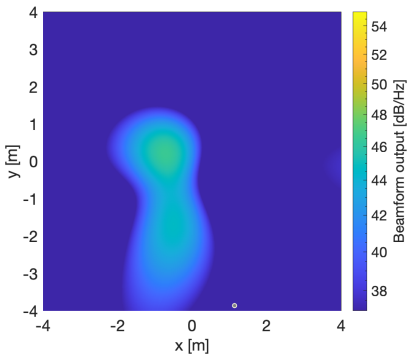


(h) Recording 3 above a stone surface at 5000 (Hz)

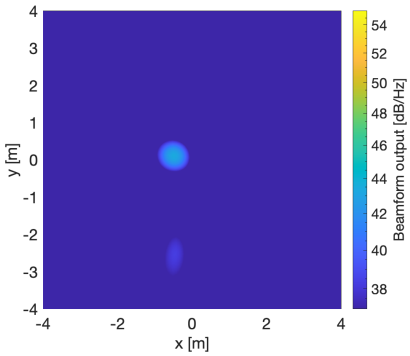
Figure 5.1: Beamform plots above different surfaces at multiple frequencies second 4 of the recording



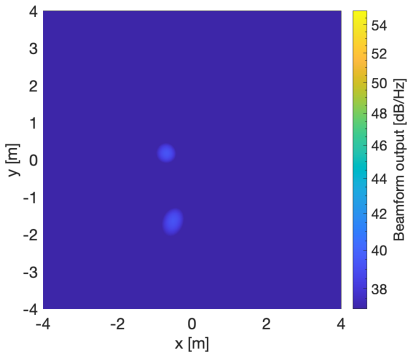
(a) Recording 1 above a grass surface at 1000 (Hz)



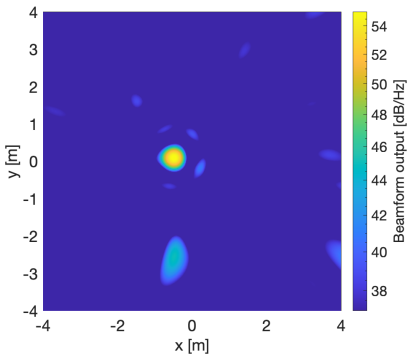
(b) Recording 3 above a stone surface at 1000 (Hz)



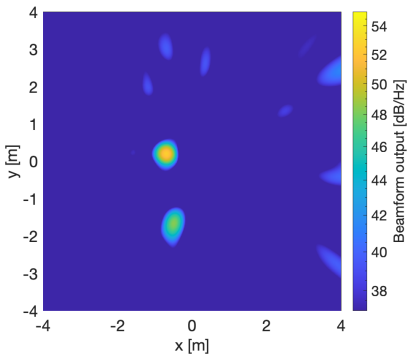
(c) Recording 1 above a grass surface at 2000 (Hz)



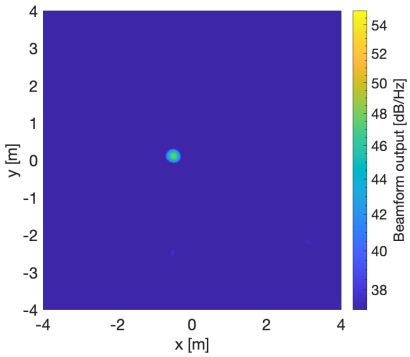
(d) Recording 3 above a stone surface at 2000 (Hz)



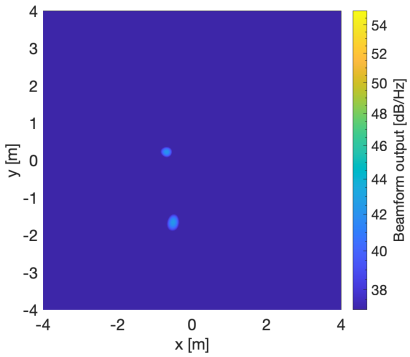
(e) Recording 1 above a grass surface at 3500 (Hz)



(f) Recording 3 above a stone surface at 3500 (Hz)



(g) Recording 1 above a grass surface at 5000 (Hz)



(h) Recording 3 above a stone surface at 5000 (Hz)

Figure 5.2: Beamform plots above different surfaces at multiple frequencies second 7 of the recording

6

Appendix 2

Figure 6.1 presents a scatter plot containing results of the simulation. The rows are sorted by method, the columns are sorted by coordinates. The x, y and z estimates are plotted against the actual source coordinates. It can be seen, that the neural network has the least straight lines compared to the global optimization methods. This is a result of obtaining the least accurate estimates. Often when one of the estimated coordinates of the global optimization methods is not correct, the other two coordinates are neither correct. This is a result of the global optimization method getting stuck in a local optimum. A neural network could possibly be sensitive to correlated misses as well.

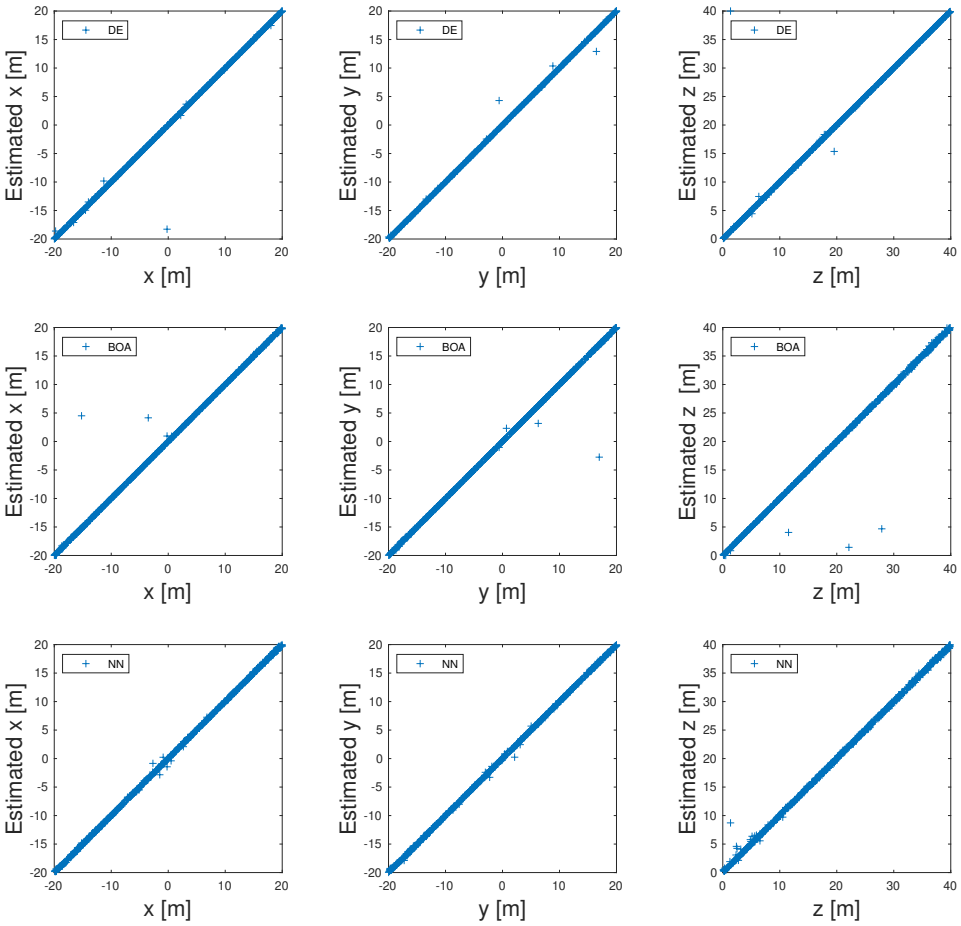


Figure 6.1: Actual source locations presented against simulated estimates