Estimation of the Direction of Arrival of Sounding Reference Signal with Neural Networks

Akseli Koljonen

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 24.02.2020

Supervisor

Prof. Olav Tirkkonen

Advisor

M.Sc. Marko Hassinen

Aalto University
School of Electrical
Engineering

Aalto University
School of Electrical
Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Akseli Koljonen

Title Estimation of the Direction of Arrival of Sounding Reference Signal with Neural Networks

Degree programme Computer, Communication and Information Sciences

Major Communications Engineering **Code of major** ELEC3029

Supervisor Prof. Olav Tirkkonen

Advisor M.Sc. Marko Hassinen

Date 24.02.2020 **Number of pages** 71 **Language** English

Abstract

The fifth generation of cellular network technology, 5G, will be the next step in performance for wireless devices. The need for 5G technology is obvious since the ever-increasing demands for higher performance in terms of data rate and latency have brought pressure for the current technologies. Hence, 5G brings new technology solutions to the networks such as beamforming to increase the performance of the network. However, the new solutions might require more computational resources from the network, which need to be taken into account when designing the network.

This thesis examines if a neural network can be used in sounding reference signal (SRS) beamforming. SRS beamforming corresponds to the signal processing technique done in the physical layer in the base station of the network. SRS beamforming can be done with multiple different methods, where one of the methods requires a calculation of the direction of arrival (DOA) of a SRS from a user. The thesis aims to search for the optimal structure of a neural network for the estimation of the DOA in terms of accuracy and computational complexity. The results of the thesis indicate, if a neural network is a feasible solution for the DOA estimation in SRS beamforming and how well the network performs compared to the traditional DOA estimation method.

The implemented neural network in this thesis is compared to one of the traditional DOA estimation algorithm, ESPRIT. The comparison was done by using different channel properties to test the neural network in different scenarios. The observed results for the neural network were moderate compared to the ESPRIT. The neural network was able to estimate the DOA confidently to some point. The network was robust against changes in the channel but was not able to yield as accurate estimations as assumed. Also, the generalization of the network was not satisfying. However, the results indicated that a neural network can be a feasible solution for the DOA estimation problem, but it needs further research.

Keywords Neural network, machine learning, digital signal processing, mobile networks, channel estimation, DOA, sounding reference signal

Aalto University
School of Electrical Engineering

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Diplomityön tiivistelmä

Tekijä Akseli Koljonen

Työn nimi SRS-signaalin tulokulman estimointi neuroverkolla

Koulutusohjelma Computer, Communication and Information Sciences

Pääaine Tietoliikennetekniikka **Pääaineen koodi** ELEC3029

Työn valvoja Prof. Olav Tirkkonen

Työn ohjaaja FM Marko Hassinen

Päivämäärä 24.02.2020 **Sivumäärä** 71 **Kieli** Englanti

Tiivistelmä

Viidennen sukupolven matkapuhelinverkko 5G tulee olemaan seuraava askel tehokkuudessa langattomille laitteille. Tarve 5G-teknologialle on selvä, sillä jatkuvasti kasvava tarve nopeammalle siirtokapasiteetille ja lyhyemmälle viiveille on tuonut paineita nykyisille matkapuhelinverkoille. Tämän takia 5G-tekniikka pyrkii tuomaan uusia tekniikoita, kuten keilanmuodostusksen nostamaan verkon tehokkuutta. Uudet tekniikat voivat kuitenkin olla laskennallisesti kalliita, mikä tulee huomioida verkon suunnittelussa.

Tämä diplomityö tutkii mahdollisuutta käyttää neuroverkkoa SRS (Sounding reference signal) keilanmuodostusksessa. SRS keilanmuodostus tarkoittaa signaalinkäsittelytekniikkaa, joka tehdään verkon fyysisessä kerroksessa tukiasemassa. SRS keilanmuodostus voidaan tehdä monella eri tavalla. Yhdessä tavassa joudutaan laskemaan käyttäjältä saadun signaalin tulokulma (DOA). Työn tavoitteena oli löytää paras mahdollinen neuroverkko tulokulman estimoimiseen tarkkuuden ja laskennan raskauden suhteen. Tavoitteen kautta oli mahdollista tutkia, onko neuroverkon käyttö tulokulman estimoimisessa järkevää.

Neuroverkko, joka luotiin käytännön toteutuksessa, verrattiin yhteen perinteiseen algoritmiin (ESRPIT), jolla tulokulma voidaan myös estimoida. Vertailu tehtiin käyttämällä erilaisia kanavaparametrejä, jotta neuroverkkoa voitiin testata erilaisissa olosuhteissa. Saadut tulokset osoittivat, että neuroverkko oli kykenevä estimoimaan signaalin tulokulman tyydyttävästi verrattuna perinteiseen algoritmiin. Neuroverkko pystyi estimoimaan tulokulman luotettavasti tiettyyn pisteeseen asti, mutta ei pystynyt antamaan yhtä tarkkoja tuloksia, kun olisi haluttu. Lisäksi neuroverkon yleistyminen ei ollut tyydyttävällä tasolla. Tulokset kuitenkin antoivat ymmärtää, että neuroverkko voisi olla mahdollinen ratkaisu tulokulman estimoimisessa, mutta tämä tarvitsee enemmän tutkimusta.

Avainsanat Neuroverkko, koneoppiminen, digitaalinen signaalinkäsittely, langaton verkko, matkapuhelinverkko, kanavaestimointi, signaalin tulokulma, DOA, SRS

Preface

This Master's Thesis was carried out at Nokia Solutions and Networks in Espoo, Finland.

I am particularly grateful for the assistance given by my advisor Marko Hassinen. Many thanks also belong to my squad group leader Mikko Volanen who gave me this great opportunity to work at Nokia Networks.

I would also like to thank my supervisor, Prof. Olav Tirkkonen, for his guidance as well as feedback and advice during the final stages of writing the thesis.

Also, thanks to my parents Jaana and Jouni who gave me the inspiration for studying. Most importantly, I want to thank Iina for the loving support during the final years of my studies and for motivating me during difficult times with this thesis.

Espoo, 24.02.2020

Akseli Koljonen

Contents

Symbols and abbreviations

Symbols

$\mathbf{a}$	steering vector
$\mathbf{A}$	steering matrix
A	amplitude of a wave
$\mathbf{B}$	magnetic field
b	bias
c_0	speed of light in vacuum
$\mathbf{E}$	electric field
f	frequency
h	channel impulse response
$\hat{h}$	observed channel impulse response
$\mathbf{I}$	identity matrix
$\mathbf{J}$	selection matrix
k	wave-number
L	number of paths
M	number of antenna elements
N	number of PRBs
n	noise
P	signal power
$\mathbf{R}$	covariance matrix
$\hat{\mathbf{R}}$	sample covariance matrix
$\mathbf{U}$	unitary matrix
$\mathbf{V}_n$	noise eigenvector matrix
$\mathbf{w}$	weight vector
$\mathbf{x}$	input vector
y	label
$\hat{y}$	output of neural network
λ	wavelength
Δ	distance between antenna elements
θ	direction of arrival in degrees
φ	phase shift
ϕ	phase of a wave
τ	delay of a wave
ρ	charge density
μ_0	magnetic constant
ϵ_0	dielectric constant
ϵ	mean absolute error value
σ_s	singular values
σ_n	noise variance
σ	activation function
Φ	rotation operator
Ψ	subspace rotating operator
η	learning rate

Operators

$\nabla \times \mathbf{A}$	curl of vector $\mathbf{A}$
$\nabla \cdot \mathbf{A}$	divergence of $\mathbf{A}$
$\nabla \mathbf{A}$	gradient of matrix $\mathbf{A}$
$\mathbf{A}^H$	Hermitian of matrix $\mathbf{A}$
$\mathbf{A}^T$	Transpose of matrix $\mathbf{A}$
$\hat{\mathbf{A}}$	Estimate of matrix $\mathbf{A}$
$\frac{\mathrm{d}}{\mathrm{d}t}$	derivative with respect to variable t
$\frac{\partial}{\partial t}$	partial derivative with respect to variable t
$\sum_i$	sum over index i
$\mathrm{diag}\{\mathbf{s}\}$	diagonal matrix of vector $\mathbf{s}$
$\max\{\mathbf{s}\}$	maximum value of vector $\mathbf{s}$

Abbreviations

3GPP	3rd Generation Partnership Project
BS	base station
CNN	convolutional neural network
DOA	direction of arrival
DL	downlink
ESPRIT	Estimation of Signal Parameters Via Rotational Invariance Techniques
FDD	frequency-division duplex
FDX	full duplex system
FNN	feedforward Neural Network
HDX	half duplex system
MUSIC	Multiple Signal Classification
MLP	multilayer perceptron
MVDR	minimum variance distortionless-response
NN	neural network
LOS	line of sight
LS	least squares
LTE	Long Term Evolution
PARV	path array response vector
PRB	physical resource block
ReLU	Rectified linear unit
RNN	recurrent neural network
SNR	signal to noise ratio
SRS	sounding reference signal
SVD	singular value decomposition
TDD	time-division duplex
TLS	total least squares
UE	user equipment
UL	uplink
ULA	uniform linear array
ZC	Zadoff-Chu

1 Introduction

Wireless technology is a fundamental part of modern society. Modern technologies including smart mobile phones, laptops, and cars rely on connections that are usually established wirelessly. The huge increment of wireless devices and the demand for higher performance networks have brought challenges to existing wireless technologies.

Traditional wireless communication systems use multiple antennas to transmit and receive information. The use of multiple antennas allows the system to increase data rates. However, due to the superposition principle of radio waves, the use of multiple antennas may lead to destructive interference at the receiver. This means that the waves cancel each other in the receiver. The destructive interference decreases the performance of the network. This issue is solved by beamforming.

Beamforming is a precoding technique used in wireless communication systems to increase the overall performance of the system. Beamforming allows multiple simultaneous data transmission by directing energy in a particular direction. It controls the relative phases of the transmitted or received waves by weights. This means that beam patterns created by the antennas are steered towards the wanted direction by weighing the transmitted or received wave. This creates constructive interference in the desired direction while creating destructive interference in the undesired direction.

The beam patterns are formed by utilizing a smart antenna array. In the receiver, the antenna array is used to estimate the angle of the arriving wave. In the transmitter, however, the antenna array focuses the beam to the desired user by using the information from the estimation of the angle of the arriving wave.

The problem of estimating the angle of the arriving wave is referred to as DOA (direction of arrival) problem. Since of its widespread applications and difficulties of obtaining an accurate estimator, the problem is widely studied and there has been significant progress in the development of DOA estimation methods over the last three decades. In cellular communication systems, the DOA estimation can be done by utilizing channel estimation information. Channel estimation is used to analyze the spatial channel properties where it is possible to find relevant DOA information. One way of estimating the channel is to use a reference signal called Sounding Reference Signal (SRS). However, these so-called traditional DOA estimation methods contain limitations in terms of computational complexity or accuracy, which may decrease the performance of the system. At the same time, neural networks (NN) have gained popularity among researchers. Neural networks have provided feasible solutions in many signal processing problems since they can recognize underlying relationships in given data. Thus, they are universal approximators to different problems.

This thesis examines the suitability of a neural network in the DOA estimation in a cellular communication system. The examination is done by creating a neural network and analyzing its performance compared to a traditional method. The performance is compared in different real-world scenarios with different network configurations. The neural network is trained and tested with data sets, which are created by a simulation that mimics a cellular network. The thesis aims to examine if a neural network can increase the accuracy and decrease the computational complexity in the

DOA estimation compared to a traditional method.

The thesis is divided into three parts. The first part discusses the signal processing related to the DOA estimation problem. It introduces the system model in Chapter 2 and the channel estimation in Chapter 3. Next, the traditional DOA estimation methods are introduced in Chapter 4. The second part of the thesis discusses the neural networks. In Chapter 5, the basics of neural networks and the suitability to a cellular communication system are discussed. Chapter 6 proposes the implemented neural network for the DOA estimation problem. The last part of the thesis compares the proposed neural network to a traditional DOA estimation method. In Chapter 7, the results are presented and analyzed. Lastly, Chapter 8 gives the research summary and conclusions.

2 Array receiving system

In cellular communication systems, the information is transmitted by using radio waves. Radio waves are electromagnetic waves, which propagate through a wireless channel. These waves are transmitted or received by antennas. Antennas can be divided into two groups: transmitters and receivers. Transmit antennas take in electric signals out of electronic devices and transform them into corresponding radio signals. Receiver antennas do the inverse operations. They take radio signals as inputs and transforms them into corresponding electric signals, which are then fed into electronic devices. Antennas are used in different communication systems as they act as an interface between the channel and devices.

Cellular communication systems use multiple transmission and reception antennas with signal processing capabilities. This is called smart antenna technology. Smart antenna technology is used in beamforming to multiply the capacity of a radio link and to automatically optimize the radiation and reception patterns of the antennas in response to the corresponding channel [1] [2]. This is useful since it enhances the coverage by range extension and increases the capacity of the system. Also, smart antenna technology can be used to distinguish signals by spatially separating them, which allows different subscribers to share the same spectral resources [2]. Therefore, multiple users can operate in the same cell and on the same frequency-time slot provided by utilizing adaptive beamforming techniques of the smart antennas. This reduces congestion in cellular communication systems, where the number of mobile subscribers is increasing rapidly.

This chapter introduces the smart antenna technology in the reception side, which plays a key role in the estimation of the DOA of a signal. First, the signal model is derived by using Maxwell's equations. Next, a single antenna system is introduced and the parameters which affect the performance of an antenna is defined. After this, an antenna array is introduced and lastly, it is explained how an antenna system receives a signal.

2.1 Signal model

The signal of interest is a time-varying electromagnetic wave that is used in modern technologies. These waves are used in order to transmit information wirelessly. The electromagnetic waves equation in a vacuum and charge-free space can be derived from Maxwell's equations using modern methods. Maxwell's equations contain four different equations describing the electric and magnetic fields arising from the distribution of electric charges and currents. It also explains how these fields change

in time. These equations can be written as follows

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} = 0 \tag{1}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{2}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{3}$$

$$\nabla \times \mathbf{B} = \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}, \tag{4}$$

where $\mathbf{E}$ and $\mathbf{B}$ are electric and magnetic field respectively, μ_0 and ϵ_0 are the magnetic and dielectric constants. Further, $(\cdot)$ and $(\times)$ denotes the divergence and curl, t and ρ are time and charge density. The Equation (1) is the Gauss' law, which describes the electric field pattern due to electric charges. In charge- and current-free region, the charge density $\rho = 0$. Equation (2) is the same law but for magnetism. It expresses that there are no magnetic monopolies. Equation (3) is the Faraday's law which implies that electric fields are generated by time changing magnetic fields. Further, Equation (4) is the corrected version of Ampère's circuital law, which states that magnetic fields can be generated by changing electric fields. Since only electric field $\mathbf{E}$ is considered, the magnetic field $\mathbf{B}$ can be eliminated by $\mathbf{E}$ taking the curl of the (3) and then use (4) which gives

$$\nabla \times (\nabla \times \mathbf{E}) = -\frac{\partial}{\partial t} \nabla \times \mathbf{B} = -\mu_0 \epsilon_0 \frac{\partial^2 \mathbf{E}}{\partial t^2} \tag{5}$$

By using vector identity

$$\nabla \times (\nabla \times \mathbf{V}) = \nabla(\nabla \cdot \mathbf{V}) - \nabla^2 \mathbf{V} \tag{6}$$

where $\mathbf{V}$ is any vector function of space, the (5) can be represent as

$$\nabla(\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E} = -\mu_0 \epsilon_0 \frac{\partial^2 \mathbf{E}}{\partial t^2} \tag{7}$$

Using the information from (1), the first term on the left vanishes

$$-\nabla^2 \mathbf{E} = -\mu_0 \epsilon_0 \frac{\partial^2 \mathbf{E}}{\partial t^2} \tag{8}$$

The final wave equation can be expressed as

$$\frac{\partial^2 \mathbf{E}}{\partial t^2} = c_0^2 \nabla^2 \mathbf{E} \tag{9}$$

where $c_0 = \frac{1}{\sqrt{\mu_0 \epsilon_0}} = 2.99792458 \times 10^8$ m/s is the speed of light in a vacuum. The Equation (9) describes the propagation of electromagnetic waves in a vacuum. The equation is in a vector form which contains three separate equations for each components x, y and z. For x component the (9) can be formed as

$$\frac{\partial^2 E_x}{\partial t^2} = c_0^2 \left(\frac{\partial^2 E_x}{\partial x^2} + \frac{\partial^2 E_x}{\partial y^2} + \frac{\partial^2 E_x}{\partial z^2} \right), \tag{10}$$

and likewise for E_y and E_z. All of the components are in general functions of four coordinates: x, y,z and the time t [3].

One possible solution to Maxwell's equations is the sinusoidal plane-wave. A plane-wave traveling in the x-direction is of the form

$$E(x,t) = A(t)\cos\left(2\pi f t + \phi - kx\right) = A(t)\cos\left(\omega t + \phi - kx\right) \tag{11}$$

for A and ϕ which are the amplitude (the maximum value) and the phase of the wave, respectively. The $k = \frac{2\pi}{\lambda}$ is the wave-number, which is the spatial frequency of the wave. Further, λ is the wavelenght, f is the frequency indicating complete oscillations per unit of time and x is the position of the wave in the x-axis.

By assuming one emitting source and that each transmitted wave shares the same carrier frequency, (11) can be expressed in a narrowband signal model as

$$s(t) = E(x,t) \tag{12}$$

This signal acts as a base signal for this thesis. It is a narrowband signal when the amplitude $A(t)$ and phase $\phi(t)$ varies slowly with respect to τ which is the time delay between the wave front propagating to the different elements of the antenna array, which will be introduced later in this chapter. This can be expressed as

$$A(t-\tau) \approx A(t) \text{ and } \phi(t-\tau) = \phi(t) \tag{13}$$

Since different signals are sinusoids with the same carrier frequency but with different amplitudes and phases, it is convenient to represent the signal with a phasor. A phasor defines a wavefront in a complex notation as follows

$$s(t) = s^p = A(t)e^{j(\phi(t))} \tag{14}$$

By using the phasor, the base signal from (12) can be represented as

$$s^r = \mathrm{Re}\{s^p e^{j(\omega t + kx)}\} \tag{15}$$

This model of complex signal i.e. signal which consists of the real and imaginary signal is supported by most of the receivers which decompose the received signals into real and the imaginary part. The signal in (15) is the same as in (12) but it is in complex notation.

2.2 Antenna

An antenna is a device that acts as an interface between the channel and the electronic device. It is designed to transmit or receive electromagnetic waves in a certain frequency range according to its size and form [2]. Physically, an antenna is an arrangement of conductors and surrounding materials that will either generate or induce radiating waves. By changing the arrangement or positioning of the conductors and materials it is possible to create antennas with different behaviors and properties.

The performance of an antenna is affected by multiple parameters that are reciprocal between transmit and receive antennas [4]. Therefore, only the receiving

properties of an antenna are studied and analyzed in this section. When an antenna is used in transmission, its transmission properties are simply extracted from its receiving properties through the reciprocity. The main characteristics of a receiving antenna are

- **Directionality** - An antenna may be non-directional (isotropic) or directional. The non-directional antenna receives waves uniformly from all directions i.e. the receiving power is always the same regardless of the direction of arrival. A non-directional antenna, however, receives signals more efficiently from certain directions. One type of non-directional antenna is an omnidirectional antenna. The omnidirectional antenna radiates equally across the azimuth angle but varies in the elevation angle.

- **Directivity and Gain** - Directivity and gain are two key parameters in assessing the performance of an antenna. Directivity measures the ability of an antenna to receive radiation from a certain direction. The directivity of an isotropic antenna is 1 (0 dB) meaning that it can receive radiation from every direction at the same intensity. Gain, however, combines the directivity and electrical efficiency of an antenna. The gain describes how well an antenna converts the received wave into electrical power from a certain direction.

- **Radiation pattern** - Radiation pattern is a three-dimensional representation, which describes the directional dependencies of the strength of the received wave. For an isotropic antenna, the pattern would be a sphere, meaning that the antenna receiving efficiency is the same for every direction. Usually, the radiation pattern is determined in the far-field region and is represented as a function of the directional coordinates.

- **Polarization** - The polarization refers to the plane in which the electromagnetic wave vibrates. This is important when looking at antennas since they are sensitive to polarization. In general, antennas only receive signals with a particular polarization.

  It is important to use the correct polarization in the antennas. The maximum signal can be obtained, when the polarization of an antenna is matched to the incoming signal. Otherwise, there will be a corresponding decrease in the level of the signal.

The antennas in this thesis are assumed to be omnidirectional with two different slant polarization. Slant polarization means that the polarization is at -45 and +45 degrees rather than horizontal and vertical.

2.3 Antenna array

To establish efficient beamforming by estimating the DOA, the receiving system needs to use multiple antenna elements. Multiple antenna elements are called an antenna array where antenna elements are aligned in a certain pattern.

The elements can be aligned in different patterns depending on the receiving system. One pattern is called the Uniform Linear Array (ULA). The ULA can estimate the azimuth angle of the impinging signal.

2.3.1 Uniform linear array

The ULA is a geometrical pattern for an array of antenna elements where the elements are equally spaced along a straight line. The ULA consist of M elements and the distance between the elements is Δ.

Suppose that a single source emits a narrowband base signal $s^r(t)$ defined in (11). Assuming far-field conditions, where the distance d between source and receiver is large, the emitted signal can be seen as a planar in the receiver. This means that the propagating field of the signal arrived at the array of sensors is considered parallel to each other. The plane wave is received by the array of dual-polarized elements at an angle θ. Dual polarized means that the elements can respond to both slant polarized waves simultaneously.

The signal is received first by the right-most element, which acts as a reference element located in the origin ($x = 0$). The received signal $s_1(t)$ is thus a delayed version of the base signal with a delay of $\tau = \frac{d}{c}$. Hence, the received signal is

$$s_1(t) = s^r(t - \tau) = A(t - \tau)\cos\left[\omega(t - \tau) + \phi(t - \tau)\right] \tag{16}$$

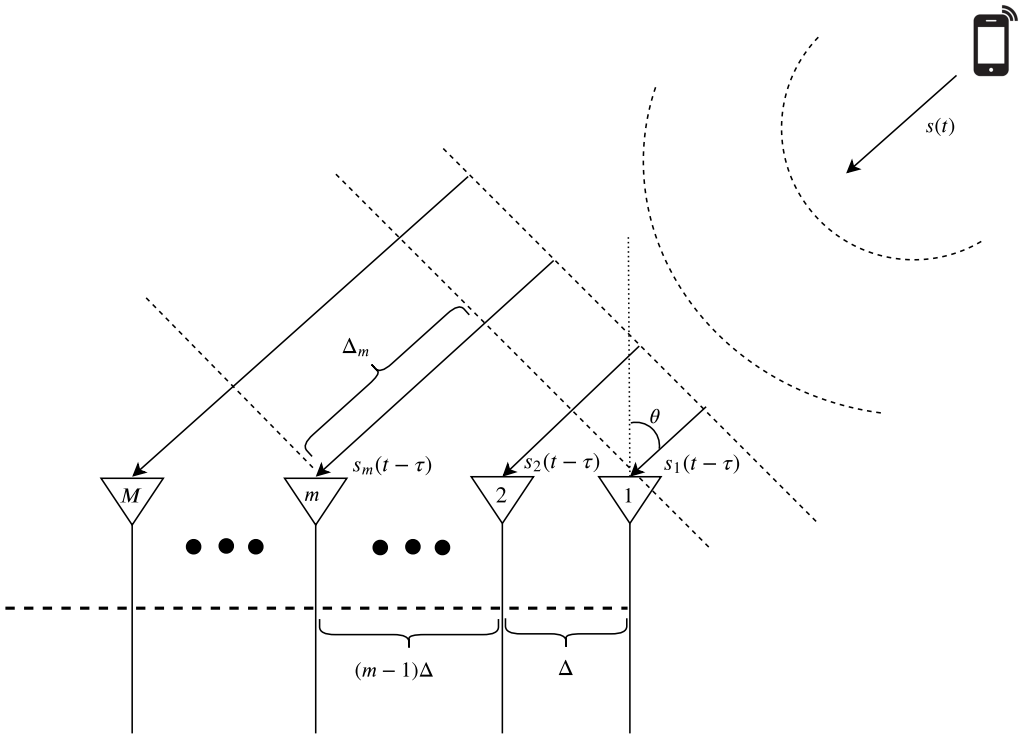

Figure 1: an Uniform Linear Array with M elements receiving signal

The complex representation of this signal is

$$s_1(t) = \mathrm{Re}\{A(t-\tau)e^{j[\omega(t-\tau)+\phi(t-\tau)]}\} = \mathrm{Re}\{s(t)\} \tag{17}$$

where $s(t)$ is the corresponding phasor.

Since in ULA the elements are equally spaced, the signal travels an extra distance to the mth element compared to the reference element as seen in Figure 1. The extra distance causes a time delay between the received signals. The time delay of a signal arriving at the mth element is

$$\tau_m = \frac{\Delta_m}{c}, \quad m = 1,\, 2,\, 3\, \ldots\, M \tag{18}$$

where $\Delta_m = (m-1)\Delta \sin\theta$ is the extra distance that the signal travels. Therefore, the real signal received by mth element is a delayed version of the signal s_1 received by the reference element

$$\begin{aligned} s_m(t) &= s_1(t-\tau_m) = s^r(t)(t-\tau-\tau_m) && (19)\\ &= A(t-\tau-\tau_m)\cos\left[\omega(t-\tau-\tau_m)+\phi_l(t-\tau-\tau_m)\right] && (20)\\ &= A(t-\tau)\cos\left[\omega(t-\tau)+\phi_l(t-\tau)+k(m-1)\Delta\sin\theta\right] && (21)\\ &= \mathrm{Re}\{s(t)e^{j[k(m-1)\Delta\sin\theta]}\} = \mathrm{Re}\{s(t)e^{j(m-1)\varphi}\}, && (22) \end{aligned}$$

where $\varphi = -k\Delta\sin\theta$ denotes the phase shift associated to signal at angle θ. The phase shift expresses the extra distance that the signal travels to the mth element in radian. In other words, the time delay between the received signals is related to the phase of the signal in frequency, meaning that the phase is different at the elements. The phase shift simply denotes the phase different in each element. In the complex phasor form the signal can be expressed as

$$s_m(t) = s(t)e^{j(m-1)\varphi}, \tag{23}$$

where it can be seen that the received signal by mth element is same as the signal received by the reference element but with an additional exponential factor $e^{j(m-1)\varphi}$. This factor is only dependent on the phase shift φ and the distance between the mth element and the reference element. The phase shifts at each element can be expressed with a steering vector

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 & e^{j\varphi} & e^{j2\varphi} & \cdots & e^{j(m-1)\varphi} \end{bmatrix} \tag{24}$$

The steering vector shows that for each incident θ that determines a signal, there is a corresponding phase shift φ. This fact can be used in the estimation of the DOA θ by extracting the phase shift from the signal received by the array.

To distinguish signals coming from distinct angles θ from φ, the phase shifts φ are limited to $\pm\pi$ and the range of possible DOAs is restricted to the interval of $\pm 90°$. These restrictions requires that the spacing between the elements satisfies $\Delta \leq \lambda/2$ [2]. Otherwise, there is an ambiguity in the DOA estimation. The ambiguity causes that there will be two solutions for the angles from a specific value of φ. This means

that grating lobes (side lobes with the same amplitude as the main lobe) permit signals from undesired directions.

In addition to the phase shifts, the signal experiences different phenomena when propagating from the transmitter to the receiver. These phenomena are discussed in the next chapter.

3 Wireless channel

In wireless communication systems, signals are transmitted in a wireless channel. In Chapter 2, the channel was assumed to be a vacuum and charge-free space. However, in reality, the channel has properties that affect the propagating signal. These properties add distortion and noise to the signal which causes errors in the receiving side. The errors occur since the channel corrupts the signal, thus the receiver is not able to extract the transmitted information from the received signal. To remove the errors caused by the channel, the channel needs to be estimated. By estimating the channel, it is possible to mitigate the negative effects. Additionally, the channel estimates contain essential directional information of the signal, which can be used in the DOA estimation problem.

The chapter discusses first the wireless channel in which the signal propagates. Channel properties need to be addressed to design a reliable and efficient network. Next, the chapter introduces methods to estimate the channel to find the channel properties and mitigate the negative effects on the receiver. Lastly, the chapter derives the channel estimation matrix and the spatial covariance matrix, which will be used in the DOA estimation.

3.1 Communication channels

As explained earlier, the thesis considers cellular communication networks that use base stations to provide communication with mobile subscribers over a large geographic area. In cellular networks, the signal can be either transmitted by the mobile subscriber i.e. user and received by the base station or vice versa. The former direction is called the uplink and the latter is the downlink. For a cellular system, it is important that data can be sent in both uplink and downlink simultaneously. This kind of system is called a duplex communication system. There are two basic schemes of duplexing used in cellular systems: [5]

- **Time Division Duplexing** (TDD) where downlink and uplink operate on the same frequency band. The system assigns alternative time slots for transmitting and receiving operations. This allows an asymmetric flow for uplink and downlink data transmission.

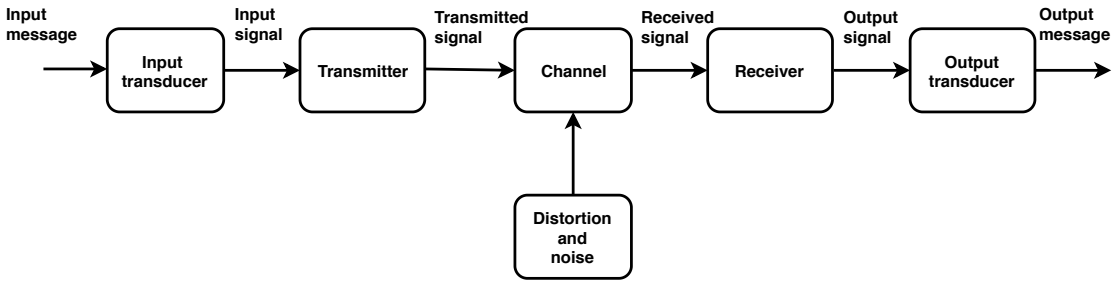

Figure 2: Wireless communication system

- **Frequency Division Duplexing** (FDD) where downlink and uplink operate on different frequency bands. Therefore, FDD requires two different channels for communication. Besides, if the channels are adjacent, there must be a guard band between the channels to mitigate the interference between the channels. Hence, FDD uses at least twice the spectrum needed by TDD.

The TDD duplex scheme brings several advantages and flexibilities important to cellular communication systems. One advantage is the channel reciprocity. Channel reciprocity means that the channel properties, which will be explained next, are the same for uplink and downlink [6]. By estimating the channel in the uplink direction, downlink direction is also estimated assuming that the channel does not change in the estimation interval. As a result, the reciprocity leads to better transmit parameter optimization for resource allocation [6]. This channel reciprocity is assumed throughout the thesis.

3.1.1 Spatial channel properties

In communication systems with low antenna heights, there are often multiple indirect paths between the transmitter and the receiver. The paths are due to reflections, scatterers, and diffractions by obstacles such as buildings, terrain and other objects. In addition to these indirect paths, there can be a direct path called LOS (Line of Sight). A LOS path exists only if there is a visual line between transmitter and receiver. The multipath propagation is particularly significant in urban environments, where multiple buildings and surfaces cause strong reflections to the signal. Due to the multipath propagation, the received signal is a summation of several components i.e. signal copies with different amplitudes, delays, phases, and DOAs.

Figure 4 illustrates the multipath propagation channel in uplink direction. There are five different paths that the signal travels, where the orange is the LOS component.

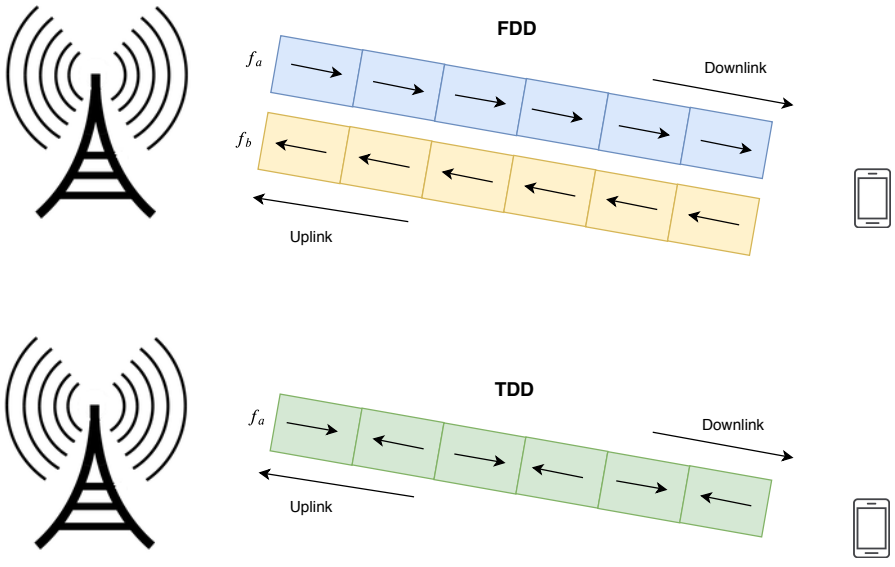

Figure 3: FDD vs. TDD

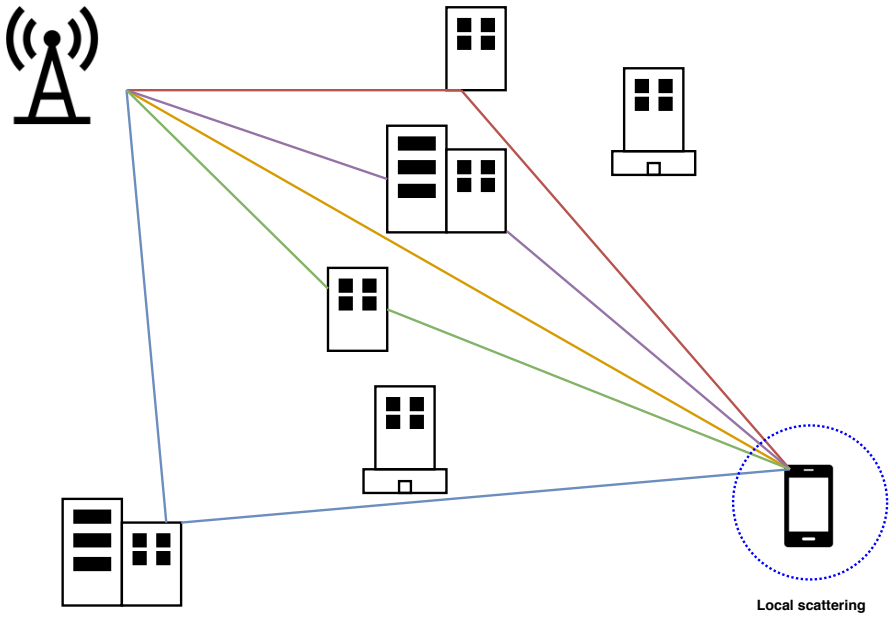

Figure 4: The wireless propagation environment

As seen, each signal component is received from a different angle and each component has individual power and delay. These differences between the components are due to the different attenuation and lengths of the paths. By using these differences, each path can be characterized individually. The characterization can be done with a channel impulse response. A channel impulse response characterizes the multipath propagation channel and describes how the channel affects the signal.

Suppose that a transmitted unit impulse input signal $\delta(t)$ impinges a receiving system with M elements in the uplink direction. The channel impulse response for path l can be written as

$$h_l(t) = \sum_{m=1}^{M} A_l e^{-j\phi_l} e^{j(m-1)\varphi_l} \delta(t-\tau_l) = \sum_{m=1}^{M} h_l^{CH} e^{j(m-1)\varphi_l} \delta(t-\tau_l) \tag{25}$$

where A_l, τ_l and ϕ_l are the attenuation, delay and phase, respectively. h_l^{CH} expresses the complex gain i.e. complex response of path l and $e^{j(m-1)\varphi_l}$ is the corresponding phase shift. The channel impulse response for the whole multi-antenna channel can be written as

$$\mathbf{h}(t) = \sum_{l=1}^{L} h_l^{CH} \mathbf{a}(\theta_l) \delta(t-\tau_l) \tag{26}$$

where $\mathbf{a}(\theta_l)$ is the steering vector defined in (24). It should be assumed, that each path is isotropic and linear. This means that a path has the same physical properties in every direction.

The receiver must be able to cope with the dispersion of the signal arising from the echoes in the channel as well as the quick changes in the dispersion. The expected degree of dispersion is determined through the measurement of the power-delay profile of the channel. The power-delay profile provides an indication of the dispersion of

transmitted power over different paths in the channel. The profile can be analyzed via wideband channel estimation measurements.

3.2 Channel estimation

As explained in the previous chapter, the deployment of multiple antennas at the transmit and receive side increases the capacity of the wireless communication systems. However, this increase is based on the assumption that the channel between transmitter and receiver is known. In practice, the channel needs to be estimated via channel estimation methods. These methods will be the focus of this section.

Channel estimation is a vital method in wireless communication systems. Channel properties affect the transmitted signal and cause a loss in the information due to the changes in the received signal compared to the transmitted signal. To make efficient use of the massive number of antennas, each base station needs to estimate the channel impulse response for each user. One way of estimating the channel is by sounding. Sounding is done with known reference signals, which are used to measure the channel. In this thesis, the sounding is done by using Sounding Reference Signal (SRS).

3.2.1 Sounding reference signal

A Sounding Reference Signal is transmitted by the user and received by the base station, thus the channel is estimated in the uplink direction. To estimate the channel of a particular user, the base station needs to know which SRS this user has transmitted. The SRSs are deterministic sequences and the signal assignment is typically made when the user connects to the base station [7].

Sounding reference signals are cyclically shifted complex-value Zadoff-Chu (ZC) sequences. The sequences are generated from cyclic shifts of a root ZC sequence. The ZC sequences have multiple attractive properties, which are the reasons they are used in channel estimation.

First, the amplitude of ZC sequences is constant, which ensures efficient power amplifier utilization and maintains low Peak to Average Power Ratio (PAPR) properties of the uplink [8]. This is important since the efficiency of the power amplifier is critical due to the limited battery power in a mobile terminal of a user. Second, ZC sequences have an ideal cyclic auto-correlation, which is necessary for obtaining an accurate timing estimation at the base station [9]. The ideal cyclic autocorrelation means that the correlation between a sequence and its circularly shifted version is a delta function as

$$r_{s,s}(\tau) = \int s(t)s^*(t-\tau)dt = \delta(t), \tag{27}$$

where $r_{s,s}$, $s(t)$ and $s(t-\tau)$ are continuous autocorrelation functions, ZC sequence and the shifted version of this sequence, respectively. Further, $(\cdot)^*$ is the complex conjugate. Finally, the cross-correlation between different cyclically shifted sequences of the same root ZC sequence is zero at the receiver [7]. This means that the base station can separate the users and remove interference by addressing sequences from the same root ZC sequence to each user in a cell.

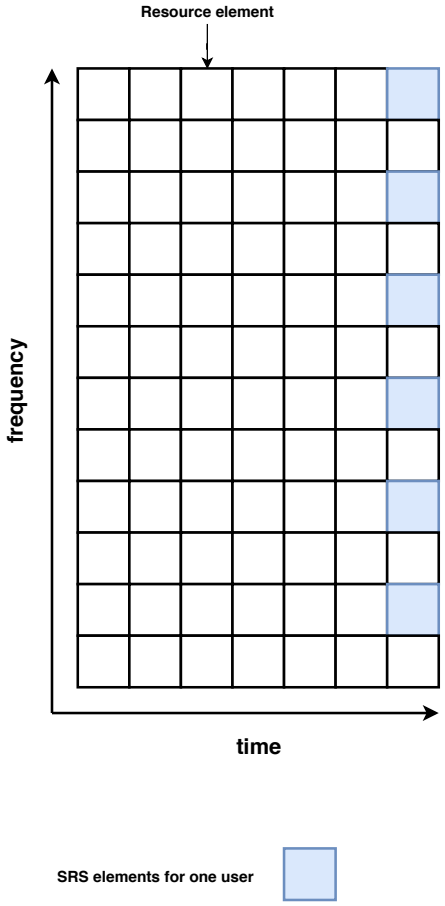

Figure 5: Physical resource block

The length of the SRS sequence depends on the number of time-frequency physical resource blocks (PRB) used in the transmission. A PRB, seen in Figure 5, is the smallest unit of resources that can be allocated to a user. In each PRB, there are 6 SRS resource elements (RE), where the RE corresponds to one time-frequency instant. Hence, the received signal in (28) contains REs of each used PRBs. The number of PRBs depends on the used configurations. The overall structure of the antenna element inputs can be seen from Figure 5.

Suppose that a user transmits a SRS denoted as $s \in \mathbb{C}$. It is assumed to have unit-magnitude elements to obtain a constant power level. The received reference signal for element m can be written as

$$y_m(t) = \sum_{l=1}^{L} h_{l,m}(t) + n_m(t) = \sum_{l=1}^{L} h_l^{CH} e^{j(m-1)\varphi_l} s(t-\tau_l) + n_m(t), \tag{28}$$

where $s(t-\tau_l)$ is the delayed version of $s(t)$ by the delay of τ_l. Further, $n_m(t)$ is the Gaussian noise process. The noise, which corrupts the signals, is assumed to be of a complex white Gaussian process. The noise has common variance σ^2 at all array elements and is uncorrelated among these elements. The noise is also additive which is taken from a zero mean random process which is uncorrelated among the

signal components. As shown in (28), the received signal on the mth element is a superposition of the waves from different paths and the noise.

Now suppose that the base station is scheduled to estimate the channel for a user in a certain frequency range. The user transmits a SRS $s(t)$ over one PRB, where the exact structure of the SRS is known at the base station. Since the cross-correlation between different cyclically shifted sequences from same root sequence is zero, the base station can estimate the channel by correlating $y(t)$ with $s(t)$. This channel estimation for the user and for mth element can be written as

$$\begin{aligned}
\hat{h}_m(t) &= \langle y_m(t), s(t) \rangle && (29)\\
&= \left[\sum_{l=1}^{L} h_l^{CH} e^{j(m-1)\varphi_l} s(t-\tau_l) + n(t) \right] * s^*(t) && (30)\\
&= \sum_{l=1}^{L} h_l^{CH} e^{j(m-1)\varphi_l} s(t-\tau_l) * s^*(t) + n(t) * s^*(t) && (31)\\
&= \sum_{l=1}^{L} h_l^{CH} e^{j(m-1)\varphi_l} \delta(t-\tau_l) + n(t) * s^*(t) && (32)\\
&= \sum_{l=1}^{L} h_l^{CH} e^{j(m-1)\varphi_l} \delta(t-\tau_l) + n'(t), && (33)
\end{aligned}$$

where $n'(t)$ is the correlation result of $n(t)$ and $s(t)$. Based on the autocorrelation property of ZC sequences, the correlation results in a series of peaks containing the delay and complex gain of each path. The paths can be identified by finding the peaks in the function. This means that, even though the signal components through different paths are correlated, they are separated and their incident angles are estimated individually. Thus, the correlation between the incident signals is removed.

The peaks in the observed channel estimations can be extracted and organized in an $M \times L$ matrix, as follows

$$\mathbf{H} = \begin{bmatrix} h_1^{CH} a_1(\theta_1) & h_2^{CH} a_1(\theta_2) & \dots & h_L^{CH} a_1(\theta_L) \\ h_1^{CH} a_2(\theta_1) & h_2^{CH} a_2(\theta_2) & \dots & h_L^{CH} a_2(\theta_L) \\ \vdots & \vdots & \ddots & \vdots \\ h_1^{CH} a_M(\theta_1) & h_2^{CH} a_M(\theta_2) & \dots & h_L^{CH} a_M(\theta_L), \end{bmatrix} + \mathbf{n}' \qquad (34)$$

The mth row of $\mathbf{H}$ corresponds to the output of mth element and the lth column contains the impulse response of the lth path on the array. Hence, one column can be called path array response vector (PARV) which is associated to an individual path.

Now, select p PARVs ($1 \leq p \leq M$) randomly from $\mathbf{H}$, thus it can be written as

$$\hat{\mathbf{H}} = \begin{bmatrix} h_1^{CH} a_1(\theta_1) & \dots & h_p^{CH} a_1(\theta_p) \\ h_1^{CH} a_2(\theta_1) & \dots & h_p^{CH} a_2(\theta_p) \\ \vdots & \ddots & \vdots \\ h_1^{CH} a_M(\theta_1) & \dots & h_p^{CH} a_M(\theta_p) \end{bmatrix} + \mathbf{n}' = \mathbf{A}\mathbf{X} + \mathbf{n}' \qquad (35)$$

where $\mathbf{A} = \begin{bmatrix}\mathbf{a}(\theta_1) & \mathbf{a}(\theta_2) & \cdots & \mathbf{a}(\theta_p)\end{bmatrix}$ is the $M \times p$ steering matrix and $\mathbf{X} = diag\begin{bmatrix}h_1^{CH} & h_2^{CH} & \cdots & h_p^{CH}\end{bmatrix}$ is the complex channel response matrix.

The $(M \times p)$ matrix $\hat{\mathbf{H}}$ contains important information from the channel and thus DOA information. This information, such as average complex gain in each path and the correlation between the transmitting antenna and receiving antenna element gains can be extracted from a spatial channel covariance matrix.

3.2.2 Covariance matrix

The next chapter will discuss the estimation of the DOA where the estimation is based on spatial covariance matrices. Spatial covariance matrix $\mathbf{R}$ is the second-order statistic to the random process measured at the antenna elements. Elements of the matrix denote the correlation between the transmitter and receiver elements.

Spatial covariance matrices are used to extract the information related to the DOA estimation from the channel. This information is based on the property that the received signals are corrupted by the noise, which is uncorrelated in the array of elements. Since the signal components are originated from the same source, they are correlated with each other. The DOA information contains the signal and noise subspaces which will be also covered in the next chapter.

The spatial covariance matrix contains the entire directional informational of the channel and it can be written as

$$\mathbf{R} = \mathbb{E}\left[\hat{\mathbf{H}}\hat{\mathbf{H}}^H\right] = \mathbf{A}\mathbf{R}_{xx}\mathbf{A}^H + \sigma^2\mathbf{I}_M \tag{36}$$

where $[\cdot]^H$ and $\mathbb{E}[\cdot]$ denotes the Hermitian operation and expected value, respectively. Further, the $p \times p$ matrix $\mathbf{R}_{xx} = \mathbf{X}\mathbf{X}^H$ represents the path gain covariance matrix, σ^2 is the common variance of the noise and $\mathbf{I}$ is the identity matrix.

Equation (36) assumes the existence of exact quantities i.e. infinite observation time. Since this is impossible to achieve in practice, a sample covariance matrix is defined as

$$\mathbf{R} \approx \hat{\mathbf{R}} = \frac{1}{K}\hat{\mathbf{H}}\hat{\mathbf{H}}^H, \tag{37}$$

where K is the number of samples.

4 Estimation of direction of arrival

The Direction of Arrival (DOA) refers to the direction from which a propagation wave arrives. Since the multipath channel contains multiple paths, thus multiple DOAs, the estimation in this thesis is based on the strongest signal. This means that the system estimates the DOA of the strongest signal component. DOA estimation is used in cellular communication systems to increase the downlink transmission efficiency [10]. Detecting and estimating the DOA in uplink can be used to design a beamformer at downlink since the geometry of the propagation channel remains the same in terms of DOAs, paths and average power [11]. A downlink beamformer means that the base station steers the transmitted signal towards the user as seen from Figure 6. The transmission is steered by changing the weights i.e. steering vector of the transmit antenna elements by utilizing the information from the DOA estimation. This means that the successful design of a smart antenna array depends highly on the performance of the DOA estimation algorithm. The processing flow of the complete system can be seen in Figure 7

In this thesis, the DOA is estimated by utilizing information from the channel estimation. This enables to identify propagation paths and separate them. Also, since the channel estimation is done via SRSs, it is possible to distinguish users.

In general, the DOA estimation techniques can be classified into spectral based and parametric methods. Spectral-based methods create a spectrum-like function of the DOA, and the peaks in the spectrum correspond to the DOA estimation. These methods are computationally attractive, but they do not always yield sufficient accuracy [2]. On the other hand, parametric methods estimate the DOA along

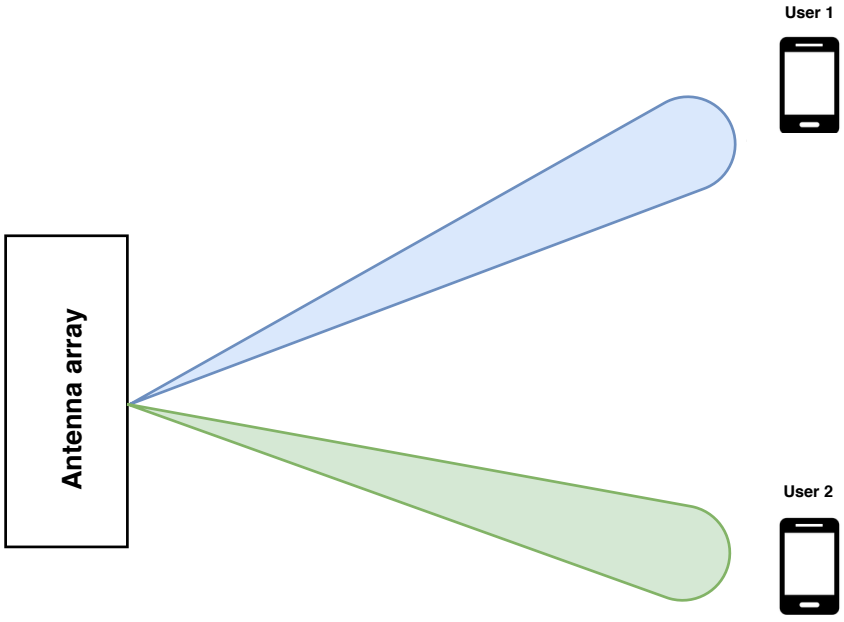

Figure 6: Two beams steered towards users in downlink.

with other parameters by optimizing a complicated nonlinear function over a high-dimensional problem space [12]. It means that they require a simultaneous search for all parameters of interest. Parametric methods are superior to spectral-based methods in terms of accuracy but they are computationally intensive [2]. Since the thesis considers a cellular communication system, parametric methods are not introduced due to computational complexity.

The chapter first introduces uplink beamforming techniques that represent the traditional spectral-based DOA estimation techniques. Following this, the chapter introduces subspace methods, which exploit the eigenspace of the spatial channel covariance matrix. These methods represent modern spectral-based DOA estimation techniques where the most relevant method for the thesis, ESPRIT, is introduced lastly.

4.1 Beamforming techniques

An uplink receiver beamformer combines the outputs of the elements in the antenna array into a single output signal. Because the received signals by the array elements at angle θ are out of phase, the combination of the signals would not give enhanced signal compared with the source. By time-shifting the signals so that they are in phase with each other, the signals are added constructively [2]. However, since the DOA is unknown, the most common strategy is to use a predefined set of presumed angles on an angular region and measure the output power of each presumed angle. The presumed angle θ_B which yields the largest output power is the estimated DOA [13]. This is called an array steering where the output power of one direction is measured at a time t.

The idea behind uplink beamforming is to rotate the array in one direction at one time and measure the output power level. The rotating is done by weighing each array response and then combining them linearly [13]. The final output for one sample is formed by

$$y(t) = \mathbf{w}^H \hat{\mathbf{H}} \tag{38}$$

where $\mathbf{w}$ is a weight vector. For K samples the output power is measured by

$$P(\mathbf{w}) = \frac{1}{K} \sum_{k=1}^{N} |y(t_k)|^2 = \mathbf{w}^H \hat{\mathbf{R}} \mathbf{w} \tag{39}$$

where $\hat{\mathbf{R}}$ is the sample covariance matrix. Since the channel is assumed to be same in the uplink and downlink, the uplink beamforming weights can be used steer the

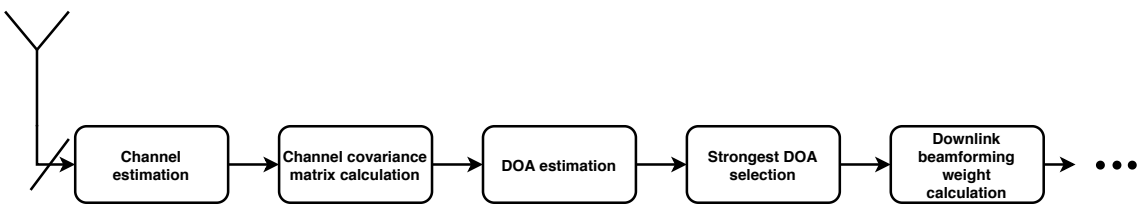

Figure 7: Basic processing flow of DOA based downlink beamforming

signal in the downlink transmitter beamforming, towards the wanted user.

4.1.1 Conventional beamforming

Conventional beamforming also known as Bartlett beamforming maximizes the power of the output of an input signal coming from an angle θ [13]. The weight vector $\mathbf{w}$ is equal to the steering vector $\mathbf{a}(\theta_B)$ where the presumed angle θ_B is scanned over the angular region. This steering vector of an ULA isotropic elements is defined similarly as in (24), but with presumed scanning angle θ_B:

$$\mathbf{a}_{ULA}(\theta_B) = \begin{bmatrix} 1 & e^{j\mu} & \cdots & e^{j(M-1)\mu} \end{bmatrix}^T \tag{40}$$

where $\mu = -k\Delta \sin\theta_B$. For each presumed angle, the output power is measured using (39). Therefore, when the presumed angle is the same as the real angle of the signal, $P(\mathbf{w})$ will have a peak in the spectrum. For practical computations, the weight vector is normalized as

$$\mathbf{w} = \mathbf{w}_{BAR} = \frac{\mathbf{a}(\theta)}{\sqrt{\mathbf{a}^H(\theta)\mathbf{a}(\theta)}} \tag{41}$$

This weight vector can be seen as a spatial filter that is matched to the impinging spatial angles of the input signal to produce a peak. At the same time, the beamformer is attenuating the output power for signals coming from other directions. The weighting equalizes the delays due to various elements and maximally combines their respective contributions. By inserting the normalized weight vector to (39) the output power as a function of the DOA is obtained as

$$P(\theta) = P_{BAR}(\theta) = \frac{\mathbf{a}^H(\theta)\hat{\mathbf{R}}\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\mathbf{a}(\theta)} \tag{42}$$

Hence, the conventional beamformer assumes that pointing to the strongest beam in a particular direction yields a peak of power arriving from that direction. However, if there is more than one input signal, the signals over a wide angular region contribute to the output of the array. This means that the local maximum of average output power can be shifted away from the true DOA of a weak signal by strong interference signals. Also, the resolution limitation of the conventional beamformer is fairly poor meaning that conventional beamformer is not capable of distinguishing closely spaced signal sources. In other words, the conventional beamformer works well, if there is only one incoming signal from the channel. If there is more than one signal present, the array output power contains signals coming from an unwanted direction. This is because a conventional beamformer uses all the degrees of freedom available to the array to form a beam in the desired direction.

4.1.2 Capon's beamforming

Capon beamforming technique [14] was proposed to improve the conventional beamforming in order to resolve power of two sources spaced closely. The problem was

posed for a particular direction θ as

$$\min P(\mathbf{w}) \quad \text{subject to} \quad \mathbf{w}^H \mathbf{a}(\theta) = 1, \tag{43}$$

where $P(\mathbf{w})$ is defined in (39). The Equation (43) minimizes the noise and the average power of signals coming from other directions than θ while maintaining the power level of the signal of interest constant. The resulting weight vector is formed by

$$\mathbf{w} = \mathbf{w}_{CAP} = \frac{\hat{\mathbf{R}}^{-1}\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\hat{\mathbf{R}}^{-1}\mathbf{a}(\theta)} \tag{44}$$

By inserting (44) into (39), the following spatial power spectrum is obtained

$$P(\theta) = P_{CAP}(\theta) = \frac{1}{\mathbf{a}^H(\theta)\hat{\mathbf{R}}^{-1}\mathbf{a}(\theta)} \tag{45}$$

Unlike the conventional beamformer, the Capon's beamformer uses degrees of freedom to form a beam in the desired direction. At the same time, the beamformer forms nulls to other directions [2]. This means that for a particular direction, Capon's method uses all but one degree of freedom to minimize the array output power. The remaining degree of freedom is used to constrain a gain in the desired direction to be unity [2]. This is why Capon's beamformer is usually called the minimum variance distortionless-response (MVDR) beamformer since it minimizes the variance i.e. average power of the array output signal while passing the signal of interest from the look direction without distortion.

Although the Capon's method reduces the spectral leakage from closely spaced sources, the resolution capability is still low if correlated signals are present [15]. This is because the correlation matrix $\hat{\mathbf{R}}$ becomes singular for the correlated signals. This means that the correlated components will be combined destructively when the output power is minimized. Also, Capon's method requires more computation power which can be expensive for large arrays.

4.2 Subspace-based techniques

Subspace-based techniques for DOA estimation are based on the spectral decomposition of the covariance matrix to examine the space of the covariance matrix. Subspace-based techniques are based on the properties that the space of the covariance matrix can be divided into two subspaces. These subspaces are the signal and noise subspaces [2]. The signal subspace is spanned by the eigenvectors associated with the larger eigenvalues of the covariance matrix. On the other hand, the noise subspace is spanned by the eigenvectors with the smaller eigenvalues.

4.2.1 Concept of subspaces

The concept of subspaces is derived by following the steps in [2]. Assume a matrix $\mathbf{B}$ of size $M \times N$. The matrix has p dimensional range (column space) if there are p ($p \leq M, p \leq N$) linearly independent columns. The p dimensional range is also the

subspace of the M-dimensional Euclidean space $\mathbf{C}^M$. The rank of $\mathbf{B}$ equals to the number of independent columns i.e. the dimension of the subspace. If $p = M$, the matrix $\mathbf{B}$ is of full rank. However, if $p < M$, the matrix is said to be rank deficient. The Euclidean space $\mathbf{C}^M$ can be spanned by the columns of any unitary matrix in $\mathbf{C}^{M\times M}$, which is defined as the Euclidean space of square and complex-valued M dimensional matrices.

This also holds for $\mathbf{C}^N$ of which row space of $\mathbf{B}$ is a p dimensional subspace. This means that there are p ($p \leq M, p \leq N$) independent rows in $\mathbf{B}$ and $\mathbf{C}^N$ can be spanned by the rows of any unitary matrix in $\mathbf{C}^{N\times N}$

Suppose p ($p \leq M, p \leq N$) column space. A unitary matrix $\mathbf{U}$ can be defined in a way that the p dimensional column space of $\mathbf{B}$ is spanned by a subset of first p columns of $\mathbf{U}$ which together form a matrix $\mathbf{U}_s$. Let the remaining $M_s - p$ columns form a matrix $\mathbf{U}_o$. Hence,

$$\mathbf{U} = [\mathbf{U}_s \mathbf{U}_o] \tag{46}$$

For the unitary matrix $\mathbf{U}$, it can be shown that

1. From $\mathbf{U}^H\mathbf{U} = \mathbf{I}_M$

$$\mathbf{U}_s^H \mathbf{U}_s = \mathbf{I}_p \tag{47}$$
$$\mathbf{U}_s^H \mathbf{U}_o = 0 \tag{48}$$
$$\mathbf{U}_o^H \mathbf{U}_o = \mathbf{I}_{M-p} \tag{49}$$

2. From $\mathbf{U}\mathbf{U}^H = \mathbf{I}_M$

$$\mathbf{U}_s \mathbf{U}_s^H + \mathbf{U}_o \mathbf{U}_o^H = \mathbf{I}_M \tag{50}$$

where $\mathbf{I}$ is a identity matrix and subscripts express the ranks. The relations above show that any vector $\mathbf{u} \in \mathbf{C}^M$ can be decomposed into two vectors $\mathbf{u}_s$ and $\mathbf{u}_o$ from spaces spanned by $\mathbf{U}_s$ and $\mathbf{U}_o$ respectively and that those vectors are orthogonal to each other. Therefore, a given matrix $\mathbf{B}$ must be decomposed in order to obtain these subspaces.

4.2.2 Singular value decomposition

Singular value decomposition (SVD) is a generalization of the eigendecomposition of a positive semidefinite normal matrix. SVD is a tool that is used to decompose the range space of a matrix into two complementary subspaces [16].

The matrix $\mathbf{B}$ defined above can be decomposed as follows:

$$\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^H = \begin{bmatrix}\mathbf{U}_s \mathbf{U}_o\end{bmatrix} \begin{bmatrix}\Sigma_s & 0 \\ 0 & \Sigma_o\end{bmatrix} \begin{bmatrix}\mathbf{V}_s^H \\ \mathbf{V}_o^H\end{bmatrix} \tag{51}$$

where Σ is an $M \times N$ diagonal matrix which contains the singular values ω_{si} of $\mathbf{B}$. These singular values are positive numbers which are ordered in ascending order as follows

$$\omega_{s1} \geq \omega_{s2} \ ... \ \geq \omega_{sp} \geq \omega_{sp+1} = \ ... \ = 0 \tag{52}$$

As seen, only the first p singular values are nonzero. The p columns of $\mathbf{U}_s$ corresponding to the nonzero singular values span the column space of $\mathbf{B}$. These are called left singular vectors. Likewise, the p rows of $\mathbf{V}_s$ are called right singular vectors which span the row space of $\mathbf{B}$. [2]

The idea behind subspace-based methods is to search for directions in which the steering vectors associated with these directions are orthogonal to the noise subspace.

4.2.3 MUSIC

MUSIC (Multiple Signal Classification) is a popular method for super-resolution direction finding based on the eigenvalue decomposition of the sensor covariance matrix. Although the analysis in the following is based on an ideal signal model, it is essential to obtain important insights.

By assuming that the p paths in (35) are not spatially correlated i.e. signal components arrive from different directions, the steering matrix $\mathbf{A}$, containing linearly independent steering vectors, has full column rank. Further, since $\mathbf{X}$ is a diagonal matrix, it has also full rank and it is nonsingular as long as the signal components are not correlated. Nonsingularity means that the matrix is full rank and the inverse of the matrix exists. As explained in Section 3.2.2, the covariance matrix is written as

$$\mathbf{R} = \mathbf{A}\mathbf{R}_{xx}\mathbf{A}^H + \sigma_N^2\mathbf{I}_M \tag{53}$$

Obviously, $\text{rank}(\mathbf{R}_{xx}) = \text{rank}(\mathbf{X}) = p$, thus $\mathbf{R}_{xx}$ is nonsingular [17].

Suppose that eigenvalues of $\mathbf{R}$ are $\{\upsilon_1,...,\upsilon_M\}$ so that

$$|\mathbf{R} - \upsilon_i\mathbf{I}_M| = 0, \tag{54}$$

for all υ_i where $i = 1,2,..M$.

When substituting (53) into (54), it becomes

$$|\mathbf{A}\mathbf{R}_{xx}\mathbf{A}^H + \sigma_N^2\mathbf{I}_M - \upsilon_i\mathbf{I}_M| = 0, \tag{55}$$

for all υ_i where $i = 1,2,..M$. By assuming that $\mathbf{A}\mathbf{R}_{xx}\mathbf{A}^H$ has eigenvalues e_i, then

$$e_i = \upsilon_i - \sigma_N^2, \tag{56}$$

where $i = 1,2,..M$.

If the number of incident signal components p is less than the number of antenna elements M, a full column rank $\mathbf{A}$ and nonsingular $\mathbf{R}_{xx}$ guarantee that the matrix $\mathbf{A}\mathbf{R}_{xx}\mathbf{A}^H$ is positive semidefinite with rank p. This means that the eigenvalues of $\mathbf{A}\mathbf{R}_{xx}\mathbf{A}^H$ are non-negative and $M-p$ eigenvalues are zero. From (56), this means that $M-p$ eigenvalues of $\mathbf{R}$ are equal to the noise variance σ_N^2, thus they are smaller than the first p eigenvalues

$$\upsilon_{p+1} = \upsilon_{p+2} = ... = \upsilon_M = \sigma_N^2 \tag{57}$$

For the eigenvectors $\mathbf{q}_i$ associated with the $M-p$ smallest eigenvalues, it can be written

$$(\mathbf{R} - \sigma\mathbf{I}_M)\mathbf{q}_i = \mathbf{A}\mathbf{R}_{xx}\mathbf{A}^H\mathbf{q}_i + \sigma^2\mathbf{I}_M\mathbf{q}_i - \sigma^2\mathbf{q}_i \tag{58}$$

$$= \mathbf{A}\mathbf{R}_{xx}\mathbf{A}^H\mathbf{q}_i = 0 \tag{59}$$

Since $\mathbf{A}$ is full rank and $\mathbf{R}_{xx}$ is nonsingular, this means that

$$\mathbf{A}^H \mathbf{q}_i = 0 \tag{60}$$

The equation above shows that the eigenvectors associated with the $M - p$ smallest eigenvalues are orthogonal to the p steering vectors that correspond to $\mathbf{A}$, thus

$$\{\mathbf{a}(\theta_1),...,\mathbf{a}(\theta_p)\} \perp \{\mathbf{q}_{p+1},...,\mathbf{q}_M\} \tag{61}$$

The DOAs of the signal components can be found by searching the steering vectors, associated with the received signals, that are orthogonal to the noise subspace [18].

The MUSIC spectrum is constructed as follows

$$P(\theta) = P_{MUSIC}(\theta) = \frac{1}{\mathbf{a}^H(\theta)\mathbf{V}_n\mathbf{V}_n^H\mathbf{a}(\theta)} \tag{62}$$

where $\mathbf{V}_n = \left[\mathbf{q}_{p+1},...,\mathbf{q}_M\right]$ is the matrix containing noise eigenvectors i.e. the noise subspace.

The DOAs are found by scanning over the angular region and searching peaks in the spectrum. There are peaks when the steering vector $\mathbf{a}(\theta)$ is orthogonal to the noise subspace i.e. the value of the denominator approaches zero.

Compared to beamforming techniques, MUSIC has the advantage of much better resolution [9]. However, since MUSIC requires eigendecomposition to find the noise subspace, it has higher computational complexity and requires more storage. Also, because MUSIC assumes uncorrelated signals, the practical use in applications is limited. If the signal components are highly correlated, the covariance matrix is rank-deficient meaning that MUSIC is not able to find the peaks of the signal components [19]. Further, MUSIC can find $M - 1$ peaks from the covariance matrix. This means that if the number of paths $L > M - 1$, MUSIC is not able to find every peak.

4.2.4 Root-MUSIC

The MUSIC algorithm scans over all possible directions in order to estimate the DOAs. However, it is more efficient to estimate the DOAs by finding the roots of the polynomial $J(z)$ which is the denominator of (62) as

$$J(z) = \mathbf{a}^H(\theta)\mathbf{C}\mathbf{a}(\theta), \tag{63}$$

where $\mathbf{C} = \mathbf{V}_n\mathbf{V}_n^H$ [20]. The ULA steering vector is defined as

$$\mathbf{a} = \begin{bmatrix} 1 & z & z^2 & \cdots & z^{M-1} \end{bmatrix}^T \tag{64}$$

and z for ULA is expressed as

$$z = e^{j\varphi \sin\theta}. \tag{65}$$

The roots of $J(z)$ contain the directional information of the signal components. In an ideal model, the roots would be on the unit circle at locations determined by the directions of the components. However, in the presence of noise, the positions of

roots might shift out from the circle. In this case, the p closest roots to the circle are the roots that corresponds to the signal components [21]. The angle of these components can be calculated as follows

$$\theta_l = \arcsin\left[\frac{\lambda}{2\pi\Delta}\arg(z_l)\right], \quad l = 1,2,...,p. \tag{66}$$

The root-MUSIC algorithm simplifies the MUSIC algorithm by finding the roots of a polynomial as opposed to merely plotting the spectrum and searching for peaks. However, [22] shows that when the number of antenna elements is low and SNR is high, MUSIC performs better than root-MUSIC.

4.3 ESPRIT

Most of the DOA estimation algorithms introduced above depend on the precise knowledge of the array steering matrix $\mathbf{A}(\theta)$. It means that $\mathbf{a}(\theta_l)$ for each θ_l must be known. In practice, the steering vectors are obtained by either direct calibration or by analytical means using the information of the position and the response of each element of the array [2]. This can be expensive and time-consuming as well as error sensitive. Besides, these algorithms can be exhaustive since they search through all possible angles i.e. steering vectors to find the DOAs.

ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques) is a subspace-based DOA estimation algorithm that exploits a property called the shift-invariance of the array. By exploiting the shift-invariance, ESPRIT reduces the computational and storage requirements compared to other subspace methods.

4.3.1 Basic principle

ESPRIT assumes that the receiving antenna array is divided into two identical subarrays. These subarrays can overlap which means that some elements might belong to both subarrays.

The individual elements of both subarrays can have arbitrary polarization, directional gain, and phase response if there is an identical twin in its companion subarray [23]. Each twin element pair is assumed to be separated physically by a fixed displacement vector. Therefore, the array possesses a displacement invariance. The displacement invariance means that array elements occur in matched pairs with identical displacement vectors. The displacement invariance leads to the rotational invariance of the signal subspace associated with the spatially displaced subarrays. The invariance property is utilized by ESPRIT to find DOAs.

Assuming p signal component impinging the array with two m element subarrays, the matrix from (35) can be expressed as

$$\hat{\mathbf{H}} = \begin{bmatrix}\hat{\mathbf{H}}_1 \\ \hat{\mathbf{H}}_2\end{bmatrix} = \begin{bmatrix}\mathbf{A} \\ \mathbf{A}\Phi\end{bmatrix}\mathbf{X} + \begin{bmatrix}\hat{\mathbf{N}}_1 \\ \hat{\mathbf{N}}_2\end{bmatrix} = \tilde{\mathbf{A}}\mathbf{X} + \mathbf{N}, \tag{67}$$

where Φ is a rotation operator which is a $p \times p$ diagonal matrix that relates the SRS received by the two subarrays. The rotation operator is caused by the extra

delay in the second subarray due to the fixed displacement Δ between the two subarrays. The goal of the ESPRIT algorithm is to estimate the DOAs by determining $\Phi = diag\,[e^{j\varphi_1},...,e^{j\varphi_p}]$, where $\varphi = -k\Delta \sin\theta_p$ as defined in Section 4.1.1. This requires two steps where the first step is to estimate the signal subspace and the second to estimate the subspace rotation operator.

4.3.2 Signal subspace estimation

Because ESPRIT assumes two subarrays, the signal subspace is spanned by two sets of vectors $\mathbf{E}_1$ and $\mathbf{E}_2$ which are ideally spanned by the columns of $\mathbf{A}$. The signal subspace is obtained similarly as in (53):

$$\mathbf{R} = \tilde{\mathbf{A}}\mathbf{R}_{xx}\tilde{\mathbf{A}}^H, \tag{68}$$

where both $\tilde{\mathbf{A}}$ and $\mathbf{R}_{xx}$ are assumed to have a full rank p. Assume that the signal subspace is spanned as $\mathbf{E}_x = [\mathbf{e}_1,...,\mathbf{e}_p]$. Because the signal covariance matrix $\mathbf{R}_{xx}$ has full rank, $\mathbf{E}_x$ spans the same space as $\tilde{\mathbf{A}}$. This results to a fact that there must exist a unique nonsingular matrix $\mathbf{T}$ such that

$$\mathbf{E}_x = \tilde{\mathbf{A}}\mathbf{T} \tag{69}$$

Further, $\mathbf{E}_x$ can be decomposed into $\mathbf{E}_1$ and $\mathbf{E}_2$ of the two subarrays so that

$$\mathbf{E}_x = \begin{bmatrix}\mathbf{E}_1 \\ \mathbf{E}_2\end{bmatrix} = \begin{bmatrix}\tilde{\mathbf{A}}\mathbf{T} \\ \tilde{\mathbf{A}}\Phi\mathbf{T}\end{bmatrix} \tag{70}$$

The two subarrays span the same signal subspace and have the same dimension, which is caused by an identical configuration. Thus, a nonsingular $p \times p$ matrix Ψ can be derived such that

$$\mathbf{E}_1\Psi = \mathbf{E}_2 \Rightarrow \mathbf{A}\mathbf{T}\Psi = \mathbf{A}\Phi\mathbf{T} \tag{71}$$

and

$$\Psi = \mathbf{T}^{-1}\Phi\mathbf{T} \tag{72}$$

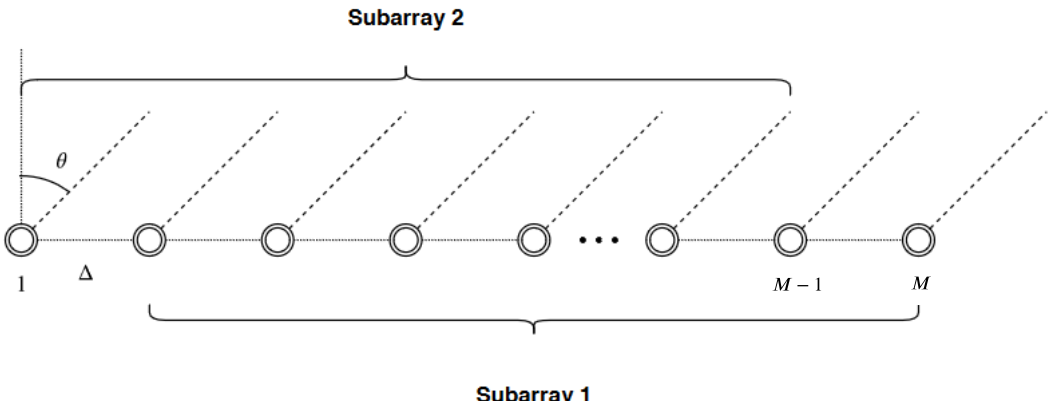

Figure 8: Array structure with 2 subarrays for ESPRIT

From the equations above, it can be seen that Ψ and Φ are related via an eigenvalue-preserving similarity transformation. Therefore, Ψ and Φ share the same eigenvalues which lie in the diagonal elements of the matrices. This means that Ψ rotates the signal subspace matrix $\mathbf{E}_1$ corresponding to the first subarray to the signal subspace matrix $\mathbf{E}_2$ corresponding to the second subarray. Hence, the shift-invariance property is expressed in terms of signal eigenvectors that span the signal subspace. This expresses that ESPRIT tries to find the subspace rotating operator Ψ and then its eigenvalues that contain the desired DOA information instead of finding directly the spatial rotation Φ.

4.3.3 Subspace rotation operator estimation

In a real-world scenario, the processing data is corrupted by noise. Noise causes that (71) cannot be solved exactly, since $\mathbf{E}_1$ and $\mathbf{E}_2$ do not span exactly the same subspace. Hence, two methods can be used to obtain a suitable estimate for Ψ: least squares (LS) and total least squares (TLS). This gives ESPRIT two different versions.

Least squares and total least squares can be explained by considering (71) in a matrix model $\mathbf{AX} = \mathbf{B}$. In order to estimate $\mathbf{A}$, the LS method assumes that $\mathbf{A}$ is known and all errors lies in the noise in $\mathbf{B}$. Thus, the solution of LS is written as

$$\hat{\mathbf{X}} = \left[\mathbf{A}\mathbf{A}^H\right]^{-1}\mathbf{A}^H\mathbf{B}, \tag{73}$$

where $\hat{\mathbf{X}}$ is the estimate of $\mathbf{X}$. However, the LS method does not take into account the fact that $\mathbf{A}$ is also determined from the received data which contains noise and disturbances. This means that $\mathbf{A}$ may have errors too. This fact is considered in the TLS method.

In the TLS method, the criterion can be stated as finding residual matrices $\mathbf{R}_A$ and $\mathbf{R}_B$ of minimum Euclidean norm such that

$$[\mathbf{A} + \mathbf{R}_A]\,\hat{\mathbf{X}} = \mathbf{B} + \mathbf{R}_B \tag{74}$$

The computation procedures of TLS is not covered in this thesis but it can be found for example in [24].

4.3.4 Standard ESPRIT DOA estimation

The previous sections presented the basics of the ESPRIT method. This section will introduce the DOA estimation steps by using the standard ESPRIT method. It will be assumed that the system uses ULA as an example with M elements.

Let's define the antenna array so that the two subarrays have maximum overlap, as depicted in Figure 8. This means that each subarray contains $m = M - 1$ elements. The selection matrices can be formed by $\mathbf{J}_1$ which contains the first $m = M - 1$ rows of $\mathbf{A}$ and $\mathbf{J}_2$ which contains the last $m = M - 1$ rows of $\mathbf{A}$. This can be expressed as

$$\mathbf{J}_1 = [\mathbf{I}_m\ 0] \in R^{m\times M} \text{ and } \mathbf{J}_2 = [0\ \mathbf{I}_m] \in R^{m\times M}, \tag{75}$$

where $\mathbf{I}_m$ is an $m \times m$ identity matrix.

Due to the fixed displacement between the first and second subarrays, the array steering vectors of the second subarray are just scaled versions of the array steering vectors of the first subarray. For a single steering vector, this can be expressed as

$$\mathbf{J}_1\mathbf{a}(\varphi_l)e^{j\varphi_l} = \mathbf{J}_2\mathbf{a}(\varphi_l), \quad 1 \leq l \leq p \tag{76}$$

For the steering matrix, containing the steering vectors, the shift invariance property can be expressed in a matrix form as

$$\mathbf{J}_1\mathbf{A}\Phi = \mathbf{J}_2\mathbf{A}, \tag{77}$$

where $\Phi = diag\,[e^{j\varphi_1},...,e^{j\varphi_p}]$ is a unitary diagonal $p \times p$ matrix. The diagonal elements contains the desired DOA information in their phases φ_l. The shift invariance property of $\mathbf{A}$ is the base of ESPRIT type methods. However, as discussed earlier, ESPRIT tries to estimate the subspace rotating operator Ψ instead of finding Φ which are directly related to each other.

By following the steps from section 4.2.2, the SVD of $\mathbf{R}$, without considering the noise, is written as

$$\mathbf{R} = \mathbf{A}\mathbf{R}_{xx}\mathbf{A}^H = \mathbf{U}_s\Lambda_p\mathbf{U}_s^H, \tag{78}$$

The equation above expresses that the columns of $\mathbf{A}$ span the p-dimensional signal subspace which is

$$S = \text{Range}\{\mathbf{A}\} = \text{Range}\{\mathbf{U}_s\} \tag{79}$$

Therefore, there exist a nonsingular $p \times p$ matrix $\mathbf{T}_A$ such that $\mathbf{A} = \mathbf{U}_s\mathbf{T}_A$. Thus, the shift invariance property of (77) can be expressed with the eigenvectors of $\mathbf{U}_s$, which span the signal subspace:

$$\mathbf{J}_1\mathbf{U}_s\mathbf{T}_A\Phi = \mathbf{J}_2\mathbf{U}_s\mathbf{T}_A \Leftrightarrow \mathbf{J}_1\mathbf{U}_s\Psi = \mathbf{J}_2\mathbf{U}_s, \tag{80}$$

where $\Psi = \mathbf{T}_A\Phi\mathbf{T}_A^{-1}$ is a nonsingular $p \times p$ matrix signal subspace rotating operator. As a result, Ψ and its eigenvalues containing DOA information can be found when the signal subspace, represented by $\mathbf{U}_s$, is estimated.

Since in practice, the unitary matrix $\mathbf{U}_s$ is computed from the sample covariance matrix defined in (37), (80) is written as

$$\mathbf{J}_1\mathbf{U}_s\Psi \approx \mathbf{J}_2\mathbf{U}_s \tag{81}$$

Thus, (81) may not yield exact solution. In addition, in order to find all the DOAs, the size of the subarrays $M-1$ need be at least equal to the total amount of paths L in the channel. The estimation of the subspace rotating operation Ψ is computed by solving the equation (81) by using either LS or TLS as explained in the precious sections. For example, if LS is used, the rotating operator can be estimated as

$$\hat{\Psi} = \left[\mathbf{J}_1\mathbf{U}_s\,(\mathbf{J}_1\mathbf{U}_s)^H\right]^{-1}(\mathbf{J}_1\mathbf{U}_s)^H\mathbf{J}_2\mathbf{U}_s \tag{82}$$

where $\mathbf{J}_1$ and $\mathbf{J}_2$ are known.

When the estimation $\hat{\Psi}$ is computed, the wanted DOA information can be extracted from it. The eigenvalues of $\hat{\Psi}$ can be computed by its eigendecomposition since

$$\hat{\Psi} = \mathbf{T}\Phi\mathbf{T}^{-1} \text{ where } \Phi = diag\left[e^{j\varphi_1},...,e^{j\varphi_p}\right] \tag{83}$$

Hence, the eigenvalue γ_l of Ψ represents estimates of the phase factors $e^{j\varphi_l}$. When the estimates of the phase shifts φ_l are computed, the corresponding DOAs θ_l can be obtained with the relationships

$$\varphi_l = \arg(\gamma_l) \text{ and } \theta_l = \arcsin -\frac{\lambda}{2\pi\Delta}\varphi_l, \quad 1 \leq l \leq p \tag{84}$$

The strongest DOA corresponds to the eigenvector with the highest eigenvalue γ_l. Thus, by only selecting the eigenvector with the highest eigenvalue from $\mathbf{U}_s$, the strongest DOA can be estimated.

5 Neural networks

A neural network (NN) is a computing system used to extract knowledge from a dataset. This knowledge can be used to give predictions and recognize patterns from the dataset. For example, neural networks are used to forecast stock markets, give a medical diagnosis and recognize human faces [25]. One of the distinct characteristics of a neural network is the ability to learn, generalize and adapt to changing situations from experience and examples. A neural network mimics some of the learning activities of the human brain and the design behind neural networks was motivated by the structure of a real brain [25]. However, the structure of a neural network is much simpler, since it contains fewer components and operates on a more abstract level.

The structure of a neural network can be divided into macro and micro levels. This chapter will first introduce the macro and then micro level of a neural network by defining fundamental components and explain how data flows through different networks. Following this, different types of neural networks will be introduced and the differences explained. The last part of this chapter will discuss the suitability of neural networks in cellular communication systems.

5.1 Macro structure

The macro-level views the network as a whole. It defines the arrangement of nodes and the number of layers in a network. As the brain consists of billions of highly connected neurons, the basic operating unit in a neural network is a neuron-like node. A network that is formed by a single node is called a perceptron. The macro-level also contains the physical organization of nodes by defining the relationships between them.

A typical neural network consists of multiple nodes located in multiple layers. Nodes can use parallelism, where they operate in parallel within the same layer. Parallelism is combined with serial operations, where multiple layers are located in sequence. The first layer of the network is called the input layer and the last layer is called the output layer. Layers between input and output layers are called hidden layers. The example network shown in Figure 9 has an input layer, one hidden layer, and an output layer.

In a neural network, each node, except nodes in the input layer, receives and processes inputs from other nodes of different layers [26]. The processed input is then available as the output of the node and is fed into the next layer or if the node lies in the output layer, the output is the output of the network. The connections between layers define how data flows through the network. The data flow in the network can be unidirectional or bidirectional. In unidirectional connections data flows in a forward direction through the network from the input layer to the output layer without any loops or back-connections. These networks are called feedforward neural networks. Bidirectional connections are usually used with recurrent neural networks. Bidirectional recurrent networks connect two hidden layers running in opposite directions to a single output. This allows the network to receive information

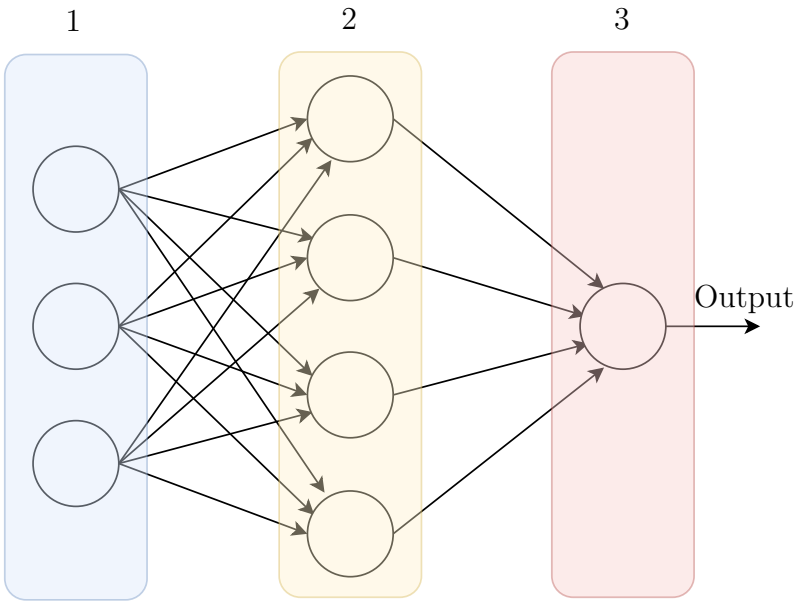

Figure 9: The example structure of a MLP. 1. An input layer, 2. A hidden layer, 3. An output layer

from both past and future states. A network can also be fully-connected, which means that each node is connected to every node in the following layer. By using different data flow directions and different numbers of hidden layers with various activation functions, different types of neural networks can be defined [25].

- **Multilayer perceptron** - MLP is the basic type of neural network. It belongs to the class of feedforward networks since the connections between nodes are always unidirectional and data flows in a forward direction [27]. It contains input and output layers and at least one hidden layer. The data flows from the input layer to the output layer through hidden layers. Each node in hidden and output layers uses usually a non-linear activation function such as sigmoid or ReLU function but also linear functions can be used. The input layer takes the raw data as an input and feeds it to the next layer through its activation functions. The nodes in the following layers make similar operations until the output layer outputs the result.

  The popularity of MLP is decreasing due to moderate performance and low convergence efficiency which originate from the usual fully-connected structure [27]. However, MLP is suitable for classification problems where the output of the network is discrete or categorical and the input data is labeled. MLP is also suitable for regression prediction problems, where the network model is used to predict a continuous variable such as house prices.

- **Convolutional neural network** - CNN is a class of feedforward networks that are commonly used to analyze visual imagery. As the name indicates, some of the hidden layers are called convolutional layers that employ a mathematical operation called convolution. The input data for the network is usually a multi-

dimensional array such as image and the network consist of input and output layer and a series of convolutional layers. The convolutional layer receives input and then transforms the input to output using convolution operators. Each layer uses different filters to find patterns from given input by using a convolution operation. The convolution operation tells how well a filter matches the input. For example, if a filter is designed to detect horizontal edges, by convolving the input matrix with the filter matrix, horizontal edges can be detected.

CNNs has a few advantages over MLPs [27]. Because CNNs use convolution layers, the networks can have different types of data as input. This support of size varying inputs allows CNNs to fit well to applications that have high multidimensional inputs such as image processing. Also, since the same filter matrix is used to convolve the whole input matrix, the parameters needed are reduced, which mitigates the risk of overfitting. CNNs are used in image processing, since

- **Recurrent neural network** - Recurrent Neural Network is designed to recognize sequential characteristics of input data and use patterns to forecast what is the next likely scenario. This is achieved by using feedback loops that allow information to persist. It means that outputs of some layers are fed back to previous layers as inputs meaning that connections between nodes are bidirectional. RNNs are used in sequence prediction problems such as language translations or forecasting events in temporal datasets i.e. stock market changes, weather forecast and travel time.

5.2 Micro structure

A neural network imitates the human brain by using large numbers of simple interconnected nodes located in different layers [25]. A node takes data from other nodes as inputs x_1, x_2,..., and performs simple mathematical operations to this data by using an activation function. After the operations, the processed data is passed to subsequent nodes.

Nodes are connected to each other, either positively or negatively. The connections have known strengths called weights w_1, w_2,..., which indicates the importance of the respective inputs to the output. An activation function determines the output signal y by using the weights and other parameters. By varying these parameters different models of decision-making can be obtained.

Modern neural network models often use non-linear and differentiable activation functions. Non-linearity allows creating complex mappings between inputs and outputs which is necessary for learning and modeling complex data such as audio, video, and images. The aim is to produce a non-linear decision boundary using non-linear combinations of the weights and inputs. Differentiability allows backpropagating the error of the model when training to optimize the parameters for the actual data, which will be discussed later in this chapter.

One basic activation function uses an arbitrary threshold value to determine output value of a node. The output is binary, either 0 or 1. If the weighted sum $\sum_j w_j x_j$ is greater than the threshold t of the node, the output is 1 and if less or equal the output is 0.

$$y = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq t \\ 1 & \text{if } \sum_j w_j x_j > t \end{cases} \tag{85}$$

Usually the threshold is replaced by an additional parameter called bias b which is used to adjust the output along with the weighted sum of the inputs.

$$y = \begin{cases} 0 & \text{if } \sum_j w_j x_j + b \leq 0 \\ 1 & \text{otherwise} \end{cases} \tag{86}$$

The threshold activation function is quite harsh, since a small change in the input of a node can sometimes cause the output to completely flip, say from 0 to 1. This property can be problematic in some cases. To overcome these problems, different activation function are defined

One popular non-linear function is the sigmoid activation function seen in Figure 11. The sigmoid function is defined as

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \tag{87}$$

where $z = b + \sum_j w_j x_j$. The standard sigmoid function is somewhat similar to the threshold model but with the addition of a region of uncertainty [26]. It means that sigmoid function can have values between 0 and 1 which can be used to measure uncertainty in the data. Thus, the output can be considered as a probability of being 1 or 0. In other words, an output of 0.5 means that the probability that the target output is 1 is the same as being 0, while an output of 0.8 means the target 1 is more likely. The output of the sigmoid function can be interpreted as a probability since output range from 0 to 1 is the same as the probability range.

However, the standard sigmoid function might be slow to compute since it requires the exponential function. In many cases, an approximation of the results will be sufficient. Hence, a hard sigmoid function is used to lower the computational

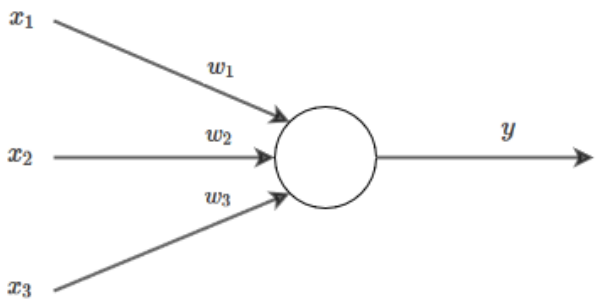

Figure 10: A node with 3 inputs and 1 output

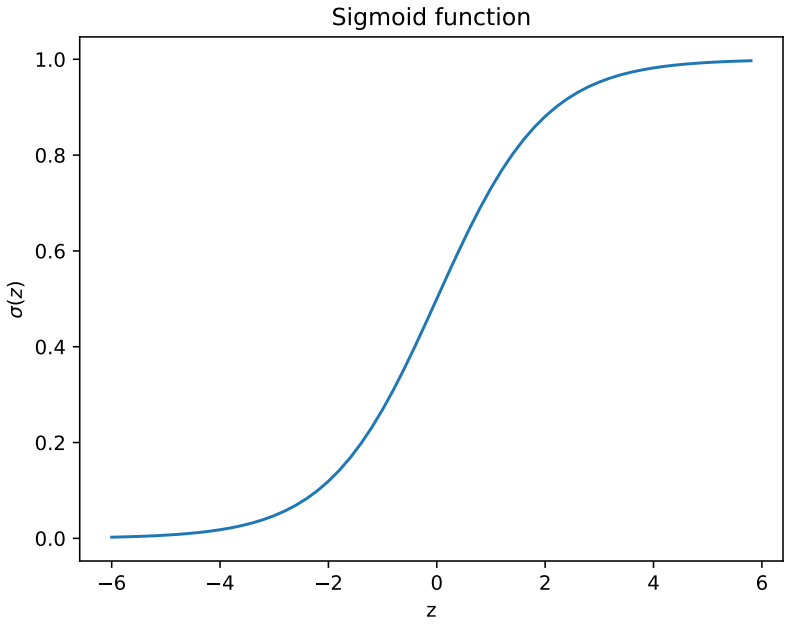

Figure 11: The "S"-shaped curve of a sigmoid function

complexity. The hard sigmoid function is faster and lighter to calculate than the standard sigmoid because it requires less computing by being a generalization of the sigmoid activation function. The hard sigmoid function is non-smooth and almost linear as seen in Figure 12.

Another popular non-linear activation function is the Rectified Linear Unit (ReLU) defined as

$$f(z) = \max(0, z) \tag{88}$$

where z is the input of the node. The ReLU function is zero for all negative inputs and when the input is greater than zero, the function returns the input. The ReLU function has multiple advantages over the sigmoid function. First, the ReLU function is trivial to implement compared to the sigmoid, since ReLU requires only a comparison function [28]. Also, the derivative of the ReLU function is easy to calculate due to the shape of the function. Second, the neural network is more sparse when using the ReLU function. Sparsity means that most of the weights are zero which leads to an increase in space and time efficiency since only some of the nodes are active [29]. Thus, ReLU is less expensive than Sigmoid. Third, since the ReLU function appears and acts as a linear function, it is easier to optimize [29].

5.3 Functionality

The design of a neural network depends on the corresponding problem, which is desired to be solved. The first step is to choose the correct network type, number of hidden layers and number of nodes in each layer. Also, activation functions

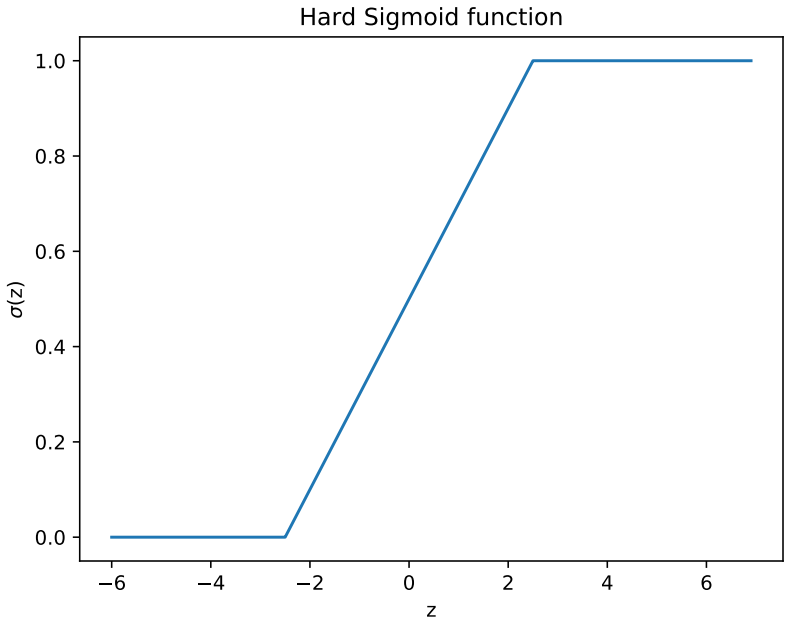

Figure 12: The hard sigmoid function

and connections between nodes need to be defined. These variables are called hyperparameters, which determine the structure of a network.

Once the hyperparameters are determined, the network model needs to be trained and tested. Training means that the weights and biases of activation functions are adjusted to receive accurate estimations. Testing, in turn, means that the performance of the network is measured with a separate dataset. The test dataset is independent of the training dataset, but it should follow the same probability distribution. If the distribution differs, the problem is called dataset shift. The data shift causes the model to yield inaccurate predictions for the test dataset [30]. Also, the test dataset should not contain the same samples as the training dataset. If there are multiple overlapping samples in the datasets, the model can not be trusted, since the model would be using the testing data for training [31]. Training and testing on the same dataset may yield extremely misleading results.

Before the network can be trained, the weights and biases are initialized. This is required for the first iteration of the training phase. When the initialization is done, the training data is fed to the network. The training data contains the correct targets i.e the desired values for the responses associated with the inputs. These targets can be then compared against estimates of the outputs given by the network using a certain metric. One popular metric is the loss function. The loss function indicates how good the estimates are compared to the targets. Thus, the smaller the output of the loss function, the better the model is for the problem in question.

One way to calculate the loss is to use a quadratic cost function also known as

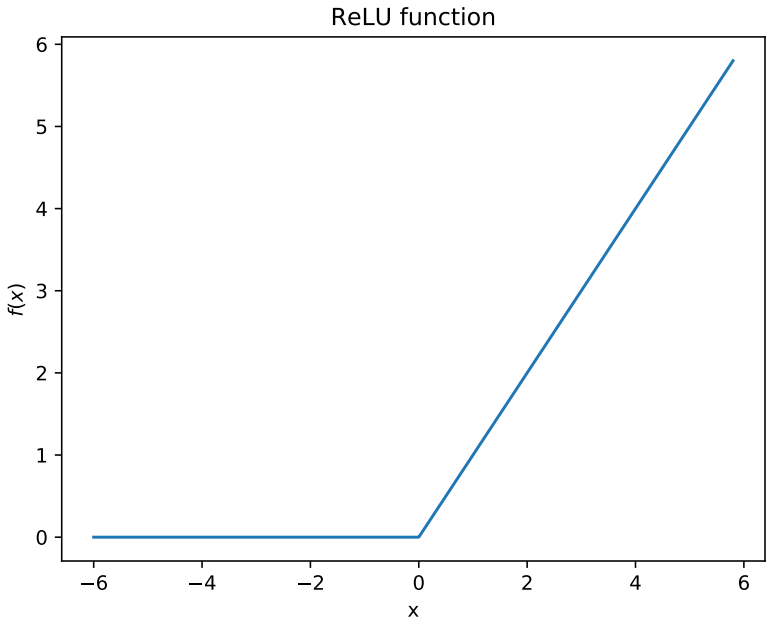

Figure 13: Plot of the ReLU function

the mean squared error (MSE) which is defined as follows:

$$L(y, \tilde{y}) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2, \tag{89}$$

where y denotes the targets, m is the total number of training inputs and $\hat{y}$ is the outputs of the network i.e. the estimates. The idea of training is to minimize the value of the loss function through multiple iterations. In each iteration, an example from the training data is fed to the input layer of the network. After this, the weights and biases are adjusted so that the value of the loss function decreases. There are multiple different methods to find the optimal weights and biases which minimize the loss function [26]. One popular method is the gradient descent.

The gradient descent is an optimization function used to update the weights and biases. Optimization is done by iteratively moving in the direction of steepest descent defined as the negative of the gradient. Since weights and biases have an individual impact on the final predictions, partial derivatives are used to minimize the loss function. For example, weights are updated as follows

$$\mathbf{w}' = \mathbf{w} - \eta \frac{\partial L}{\partial \mathbf{w}}. \tag{90}$$

where L is the loss function and $\mathbf{w}'$ denotes the updated weights. Further, η is the learning rate controlling the rate at which the model learns i.e. the step size of moving in the direction in which the gradient indicates. This operation is performed to each weight in $\mathbf{w}$ following the chain rule. The same process is performed to biases and additional parameters and eventually every parameter which minimizes the loss function are found [27]. The training phase is showed with each step in the Table 1.

Table 1: Steps of the training phase

Step	Operation
1	Initialize weights and biases with random values and calculate the loss function
2	Calculate the gradient (change in the loss function when the weights and biases are changed by a small value)
3	Adjust the weights and biases with the gradients to reach the optimal values that minimizes the loss function
4	Use the new weights for prediction and calculate new loss
5	Repeat steps 2-4 until further adjustments to weights do not significantly reduce the loss function

When the network is trained, the testing data is fed to the network. Testing data contains the targets, but unlike in the training phase, the targets are not given to the network. After testing, the estimates are compared to the targets. The testing error i.e. the error between the targets and the testing outputs is usually larger than the training error. This might indicate that the model is overfitted. Overfitting occurs when the model learns the detail and noise in the training data to the extent that it negatively affects the performance of the model on testing data [29]. This means that the model has picked up and learned the noise or random fluctuations as concepts from the training data. These models usually have high variance and low bias. The problem with overfitting is that these concepts do not apply to the testing data, thus, the error increases. Overfitting can be mitigated with different methods such as using larger and more versatile datasets, regularization and early stopping [29].

The opposite of overfitting is underfitting. Underfitting means that the model has not learned the problem, thus, it performs poorly on the training and testing data. These models, as opposite to overfitting, usually have high bias and low variance. Underfitting usually results from too little learning or the simplicity of the network i.e. there are not enough layers and nodes in the network [29].

Overfitting and underfitting can be explained with the bias-variance tradeoff where models with lower bias have higher variance and vice-versa. One would want to choose the model which both accurately captures the regularities in the training data, but also generalizes well to unseen data. This kind of ideal model would have low bias and low variance as seen in Figure 14. However, the ideal model is typically difficult to find since it is almost impossible to have low bias and low variance at the same time [32].

The design of the neural network is usually not straightforward, especially if domain knowledge is not known. This is due to the structure and number of hyperparameters. These factors should be optimized to receive accurate results. Further, the computational complexity of the network should be considered when designing a network. If the network uses a considerably large amount of resources, it might not be suitable for certain applications. It should be noted that it might be impossible to find an inexpensive network in terms of computational complexity for

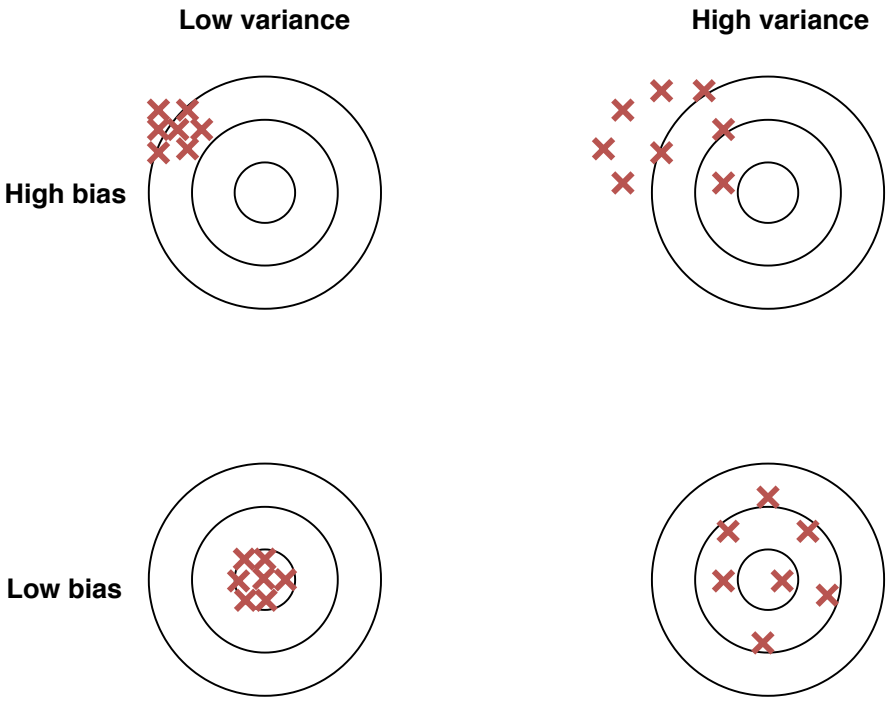

Figure 14: Bias-variance tradeoff

a difficult problem.

5.4 Suitability for wireless networks

The increasing number of mobile devices and the rising popularity of mobile applications and services pose unprecedented demands on mobile and wireless networking infrastructure. Also, the exploding mobile traffic volumes and the demand for the network to support real-time extraction of fine-grained analytics and agile management of network resources bring new challenges for mobile networks. However, conventional communication theories face several inherent limitations in fulfilling the massive data and high data rate communication with low latency in complex scenarios. Wang *et al.* [33] listed some of these limitations:

- *Channel modeling in complex scenario* - The wireless channel plays a large role when a communication system is being designed. Usually, the design of a system depends heavily on practical channel conditions or is based on channel models that characterize real environments. However, these models are inaccurate with complex scenarios with many imperfections and nonlinearities. For example, when a system uses multiple antenna elements, the channel properties become more complex which means that the channel is difficult to model and thus remains unknown.

- *Fast and effective signal processing* - Higher traffic volumes bring computational challenges for traditional receiver processing algorithms such as channel

estimation and detection algorithms. When the volume of data increases, the computation complexity increases, thus, the traditional algorithms form a computational bottleneck in real-time systems.

- *Tradition structure of a communication system* - Conventional communication system consists of a series of artificially defined signal processing blocks such as coding, modulation, and detection. Each block is optimized separately, thus an optimal performance in the entire communication task cannot be guaranteed.

Due to these and other limitations, machine learning has been seen as an interesting technique among researchers in the field of mobile networks.

The use of machine learning in mobile networks is not a new concept. Machine learning has been applied to a variety of different parts of a mobile network including channel modeling and estimation, encoding and decoding, etc. However, it is unused commercially, since conventional machine learning algorithms have limited learning capabilities. Researches believe that by using advanced machine learning techniques such as neural networks, it is possible to improve the performance of machine learning techniques in complex scenarios [33]. These improvements are based on several facts that resolve the limitations of a traditional network.

Neural networks have been proven to be a universal function approximators which have efficient learning capabilities despite complex channel conditions. By training a neural network, it is possible to optimize the end-to-end performance instead of requiring well-defined mathematical models or algorithms which are based on information theories [33]. This means that the neural network can extract high-level features from the mobile data, even if the data is noisy and created by heterogeneous sources.

Also, neural networks can handle large datasets that are common in communication networks with multiple users. Large datasets are handled with distributed and parallel computing architectures, which ensure computation speed and processing capacity [27].

Lastly, as opposed to conventional communication systems, a neural network-based system can break the traditional block structure to achieve performance improvements since they can be trained to optimize end-to-end performance [33].

6 Neural network based DOA estimation

The traditional DOA estimation algorithms face challenges in real-time multiuser environments. These challenges are related to the noisy multipath channel, where the resolution of the DOAs of signal components is small. As explained in Chapter 4, the traditional algorithms can resolve only $M - 1$ DOAs. Also, if the noise level is high, it can be challenging to find the strongest signal component. Since the DOA estimation is used in downlink beamforming, these limitations might decrease the overall performance of the communication system. Additionally, subspace-methods are known to require very complicated computations. This is due to that they are involved in the computation of matrix inversion and SVD of complex matrices, and one - or even multidimensional nonlinear optimizations [10]. Hence, it might be reasonable to estimate the DOAs with a neural network to increase efficiency.

This thesis aims to examine if a neural network is a feasible solution for the DOA estimation when using channel estimations as an input. The examination is done by creating a neural network and experimenting it with different hyperparameters and then comparing it to the traditional algorithm.

This chapter introduces the neural network used in the estimation. First, the performance indicators used in the comparison are defined. Then, the chosen network type is justified. Lastly, the architecture of the chosen network is described by defining the different hyperparameters.

6.1 Performance indicators

Since the DOA estimation is linked to the downlink precoding scheme meaning that the downlink beamforming weights can be calculated from the estimated DOA.

The neural network is compared to the traditional algorithm by using predefined performance indicators. These indicators should be fair for both methods and should consider the relationship between DOA estimation and downlink beamforming. By examining the operation environment of the traditional method, two performance indicators can be defined.

- *Error* - The error measures the difference between the estimated DOA and the correct one. The estimated DOA must be close to the correct one to minimize the power loss in the downlink transmission. However, it might not be necessary that the estimated DOA is exactly the correct DOA since some of the power of the signal lies in adjacent angles. The accuracy is measured by using Mean Absolute Error (MAE). MAE is an estimator that measures the average of the errors i.e. the difference between the estimated and the correct DOAs. The MAE is calculated as follows

$$\epsilon_l = |\theta_l - \hat{\theta}_l|, \tag{91}$$

  where θ_l is the correct angle and $\hat{\theta}_l$ is the estimated angle. The accuracy also covers the ability to resolve signals impinging from close angles i.e., the minimum gap between two incidence angles that can be distinguished.

- *Computational complexity* - The computational complexity measures how efficiently the estimation is given. It is closely related to energy consumption and decision latency. Both of these are important for the DOA estimation in a cellular communication system functioning in real-time with limited resources. Computational complexity is difficult to measure, thus in this thesis, the complexity measurement is done by calculating the Big O notation O. The Big O notation describes the limiting behavior of a function. This can be used to classify algorithms according to the required running time or space when the input size increases. The letter O relates to the order of the function which is being measured. The big O usually describes the upper bound on the growth rate of the function. There are also several other notations related to big O which describe other kinds of bounds on asymptotic growth rates.

6.2 Neural network type justification

The DOA estimation problem is a regression problem where multiple input variables in one data point are mapped into one output variable. This means that the channel estimations should be mapped into the DOA by a network model. Given the performance indicators, the performance of the network model needs to be at least as good as the performance of the traditional algorithm to be a feasible solution for the DOA estimation. This means that the error and the complexity of the neural network should be at least as small as the error of the traditional method. Therefore, the network should not be too deep or too wide. Also, the activation functions should be simple in terms of computational complexity measured with the big O notation.

The different neural network types are explained in Chapter 5. Since these network types differ from each other having different advantages and disadvantages, it is reasonable to justify the most suitable type for the DOA estimation problem.

6.2.1 Convolutional neural network in DOA estimation

As explained in Chapter 5, CNN differs from other networks by being able to process matrices as inputs. This means that it is efficient at processing large data sets with multiple dimensions. This advantage is exploited in image processing and image recognition where the network recognizes patterns in 2D data. These patterns are also found in 1D data such as natural language processing where subsequent words depend on each other.

The CNN might be a suitable network type for the DOA estimation in terms of accuracy. However, CNNs often consume substantial storage and computational resources which make them unsuitable for real-time systems with limited hardware resources [34]. Hence, a CNN seems unpractical for DOA estimation.

6.2.2 Recurrent neural network in DOA estimation

RNN uses recurrent connections to extract and utilize empirical auto-correlations in sequential data. This enables that the network can capture correlation across long

intervals of time in the input sequence [35]. Due to these properties, RNN is used in speech recognition and natural language processing, as explained in Chapter 5.

When considering a channel with only a LOS path, the subsequent DOAs depend on each other. This means that when the DOA is estimated for a user, the next estimation should be close to the previous estimation. This is due to the relatively slow motion of the users. However, when the channel contains multiple paths, the correlation among subsequent DOAs decreases. This is because the changes in the channel properties and noise occur more or less randomly from the receiver perspective. Therefore, the path, which contains the strongest signal, might switch randomly, which might be problematic for RNN.

RNN should be a feasible solution for the LOS channel DOA estimation. However, since the channel considered in this thesis contains multiple paths, it is difficult to utilize the advantages of a RNN.

6.2.3 Multilayer perceptron in DOA estimation

Due to the properties of the channel estimation data and the performance indicators, a MLP network seems the most suitable solution for the DOA estimation problem. This is because simple implementations and flexibility of a MLP network make it a universal and efficient solution for regression problems [27]. Besides, MLP networks have been successfully deployed in a variety of applications in wireless communications [36], which makes it the easiest network type to implement for the DOA estimation. It is also logically reasonable to start with the easiest option to approach the DOA estimation problem.

6.3 Introduction of the network model

The network which was implemented for the DOA estimation problem was a MLP network. This section introduces the architecture of the used network and how the network was trained and tested.

6.3.1 Architecture of the network

Following the performance indicators, the network was first built with an arbitrary hyperparameter set and then optimized by finding an appropriate hyperparameter set via trial and error. The optimization was done by first minimizing the number of layers and then minimizing the number of nodes in each layer. The set with the lowest error — computational complexity combination was chosen. The resulting hyperparameter set is shown in Table 2.

The network contains an input layer, an output layer, and two hidden layers. The number of hidden layers was chosen to balance between accuracy and computational complexity. If there were more layers in the network, the increase in the accuracy would not be remarkable. Also, the network would have been more complex since there would be more arithmetic operations. On the other hand, if there were fewer layers, the accuracy would decrease significantly.

The input layer contains 384 nodes, which is due to the size of the input data. The hidden layers contain 30 and 20 nodes, respectively, which were chosen via a trial and error. The output layer contains one node since each channel estimation corresponds to one DOA estimation.

The neural network uses ReLU activation functions in every layer, except in the output layer. ReLU was chosen due to the sparsity properties and trivial operations which make it more computationally efficient to compute compared to other activation functions such as the sigmoid function. ReLU also yielded sufficiently accurate results. The output layer uses hard sigmoid as an activation function seen in Figure 12. Hard sigmoid had the best performance among the activation functions in the testing phase.

6.3.2 Implementation

The neural network was implemented using Python (v3.7) with a Tensorflow library. Tensorflow is a machine learning library developed by Google which enables fast implementations of neural networks. Further, analysis of the data and models was done by using scikit-learn and Numpy libraries.

The input data of the network contains multiple channel estimates of a single user defined in (35). Each row in the input data contains one channel estimate. Each channel estimate consists of complex values and the number of these values depends on the number of antennas in the receiving system and the SRS configuration. The SRS configuration defines how many time- and frequency PRBs are needed for a single SRS transmission for a user where each PRB contains 6 SRS symbols. Furthermore, the complex values are divided into real and imaginary parts. This means that each complex value is presented with two values. Therefore, with M antenna elements and N PRBs, each row in the input data contains $M \times N \times 6 \times 2$ symbols.

Before the training phase, the input data were randomly divided into training and testing subsets. The training data contained most of the data with the targets. The training phase trained the network model for different scenarios, which are introduced

Table 2: The structure of the network

Hyperparameter	**Network**
Number of input nodes	384
Number of output nodes	1
Number of hidden layers	2
Number of nodes in 1st hidden layer	30
Number of nodes in 2nd hidden layer	20
Activation function in the hidden layers	ReLU
Activation function in the output layer	Hard sigmoid
Training:	
Optimizer	Adadelta
Loss function	Mean Squared Error
Metrics	Accuracy

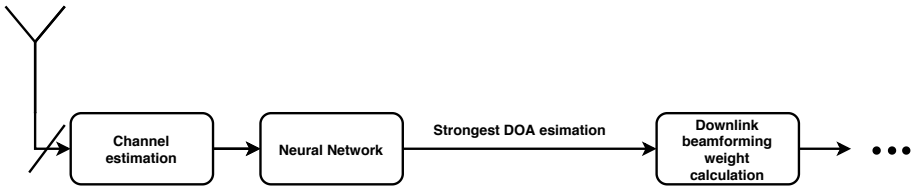

Figure 15: Neural network processing flow of DOA based downlink beamforming

in the next chapter.

In the training phase, the training data was fed to the network. With this training data, the network trained itself by optimizing the weights following Table 1. The optimization was done by using a MSE loss function presented in (89). Instead of using the gradient descent optimizer presented in Chapter 5, the network model used an optimizer called Adadelta. One difference between gradient descent and Adadelta is that Adadelta adapts the learning rate. The learning rate is based on a moving window of gradient updates. This way Adadelta continues learning even when many updates have been done.

Once the training phase was over, the testing data set was fed to the network. By measuring the accuracy with MAE, the absolute error of the network was calculated. Also, the MAE for the traditional algorithm was calculated and compared to the network.

7 Results

In this chapter, the neural network, introduced in the previous chapter, is compared to the standard ESPRIT method. The comparison is done with the performance indicators defined in Section 6.1 and in different channel scenarios i.e. channel properties. The channel properties varied the number of paths and SNR values. By using different channel properties, the overall performance of the neural network and ESPRIT was measured.

This chapter first introduces the simulation environment, where the datasets for the neural network were generated. Next, the results from each scenario are presented. Lastly, this chapter analyses the results.

7.1 Simulation environment

The channel estimates, which are used as input for the neural network, were generated with a Matlab simulation. An LTE (Long Term Evolution) communication network was simulated, which allowed to generate large datasets with different channel properties in a short time. This was required for the training phase for the neural network.

In the simulation environment, it was possible to define different channel properties. These channel properties were defined as

- Number of paths
- Delay of each path
- DOA of each path
- Power of each path
- Noise power of the channel

The training phase required a large set of data from different channel scenarios. This means that the training data had to include enough variation in respect to the channel properties. In practice, the data sets were generated with different channel properties. However, data generation with the simulation was time-consuming, hence, it was not possible to generate enough data to include different cell and SRS configurations. Thus, the simulations were run under one configuration set.

The cell configuration defined a single user with one transmit antenna element. The user transmitted a SRS to the base station through the channel using TDD. The base station contained a smart ULA antenna array with 8 receiving antenna elements. These elements were divided into two different slant types. This means that the antenna array contained 4 dual-polarized elements. However, ULA with 4 antenna elements can be problematic for the neural network and ESPRIT. This is due to the high variance value in the antenna elements. Increasing the number of elements would decrease the variance, but the simulation supported only 4 ULA dual-polarized elements. Also, the elements were divided into 2 subarrays with maximum overlap,

Table 3: Power delay profile of the simulation

Path #	Delay (ms)	Power (dB)
1	0	0
2	[200, 300]	[-1, -2]
3	[320, 600]	[-2, -4]
4	[900, 2000]	[-5, -8]
5	[2200, 4000]	[-9, -11]
6	[4200, 5000]	[-12, -14]
7	[5000, 6000]	[-15, -20]

that is, both subarrays contained $m = 4 - 1 = 3$ elements. Further, the carrier frequency was set to 1900 MHz and the element spacing was 0.475m. The user was stationary and the noise was AWGN. On the other hand, the SRS configuration defined a fixed sequence for the user, which means that the used sequence was the same in every simulation. The SRS was transmitted by using 4 PRBs. Further, the channel attenuation was set to 35 dB.

These settings ensured that the data points were derived from the same probability distribution and the correlation between data points was negligible.

7.1.1 Data generation

The data for the neural network was generated using the simulation defined above. First, the training data was generated under different channel properties defined above according to Table 3. The training data generation was divided into 5 different simulation parts as in Table 4. In each simulation, the number of paths was fixed and the SNR value was varied from -5dB to 5dB. The power delay profile was set according to 3GPP Spatial Channel Model Extension (SCME) [37] with random variance. This means that the power and delay of each path were set following the 3GPP specifications but some randomness was added. Also, the DOA of each path was randomized independently in each simulation. The randomness decreased the correlation between the data points. Due to the correlation, the network model might have yielded misleading results since it might have overfitted the training. The size of the training data was 30 000 data points in each simulation. This means that the total amount of training data was 180 000 data points.

The testing data was generated independently by using the same simulation. The testing data was generated in wider a SNR range and it also contained a number of paths outside the training data. This was done in order the measure the generalization of the model of the network i.e. how well the model adapts to new, previously unseen data. The generalization is important so that the model works with different channel properties.

The DOA estimations for the traditional method were generated in the same simulations as the testing data. This ensures that the traditional method and the neural network use the same channel estimations so that the results can be compared fairly.

Table 4: Training data

Simulation #	Number of paths	SNR (dB)	Size (data points)
1	1	[-5 .. 5]	30 000
2	2	[-5 .. 5]	30 000
3	3	[-5 .. 5]	30 000
4	4	[-5 .. 5]	30 000
5	5	[-5 .. 5]	30 000

7.2 Comparison of the neural network and the traditional method

The testing phase can be divided into two parts. These parts measured the impact of SNR and the number of paths to the DOA estimation. The first part contains three different scenarios and in each scenario, the number of paths was fixed and the SNR value was varied between -5 dB and 5 dB. The second part fixed the SNR value to 0 dB and varied the number of paths. This part also examined if the neural network has the same limitations as ESPRIT in terms of the number of DOAs it can distinguish. Also, the generalization of the network is examined by setting the path number and the SNR value to the range which was not included in the training data. Additionally, the comparison contains the computational complexity measurements between the traditional method and neural network with a single data point i.e. channel estimation.

7.2.1 DOA estimation with different SNR values

In the first scenario seen in Figure 16, where there was only one propagation path, the DOA estimation was more accurate for ESPRIT. With low SNR values, the MAE of ESPRIT was a bit over 3 degrees. However, when the SNR value was increased to 4 dB, the MAE decreases to about 1 degree. For the neural network, the curve of the MAE was similar, although the MAE was about 1 degree higher for every SNR value. For low SNR, the neural network was not able to yield sufficient estimations. For high SNR values, the estimations of the neural network were mediocre.

When there were two paths in the channel, the neural network and ESPRIT performance were the same, seen in Figure 17. With low SNR values, the MAE was lower for the neural network. As the SNR value increased, the MAE of the neural network and ESPRIT decrease with the same rate. Overall the performance was slightly better for the neural network.

When increasing the number of paths to 3, it can be seen clearly that the performance of ESPRIT is decreasing drastically. This can be seen in Figure 18. For low SNR values, the MAE increased to 6 degrees and even though increasing the SNR value, the MAE did not decrease. However, the performance of the neural network stays at the same level as with fewer paths. The MAE is not excellent for the neural network, but moderate. This means that the neural network is not as dependent on the number of antenna elements as the ESPRIT and can operate with

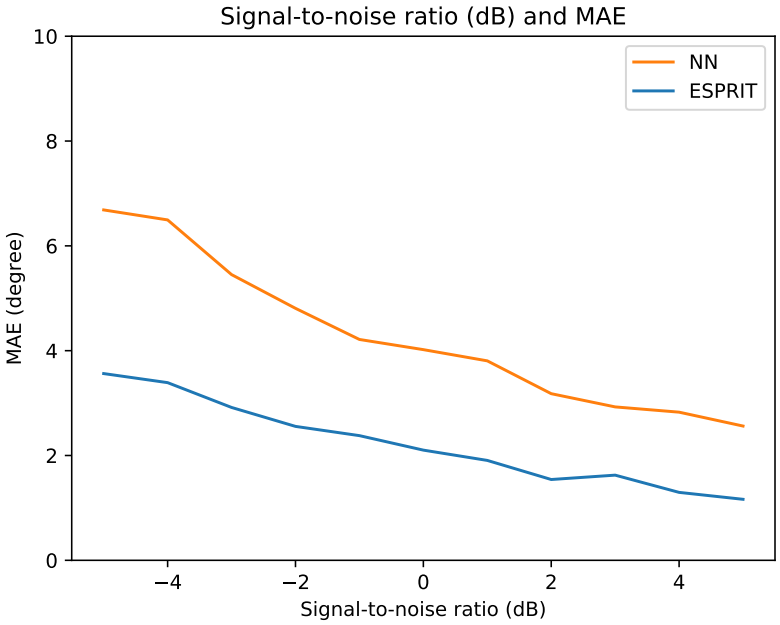

Figure 16: MAE of DOA estimation with 1 path

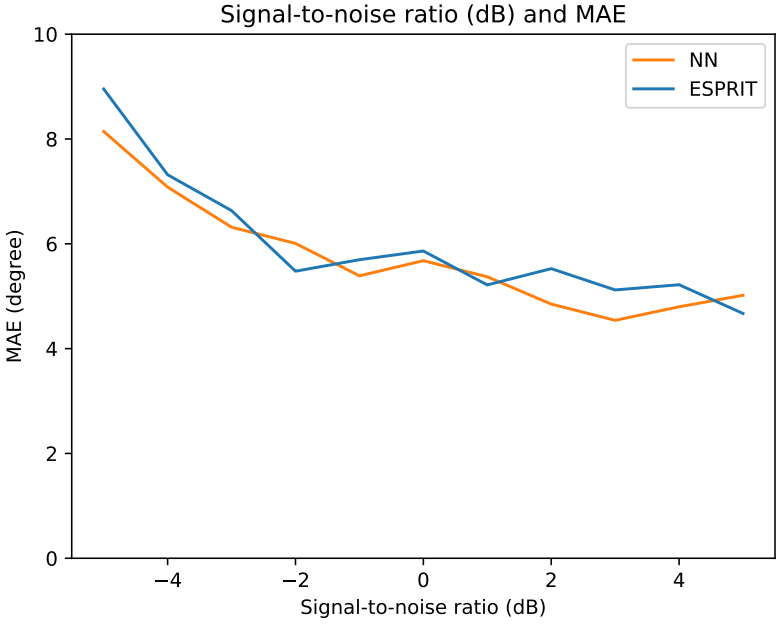

Figure 17: MAE of DOA estimation with 2 path

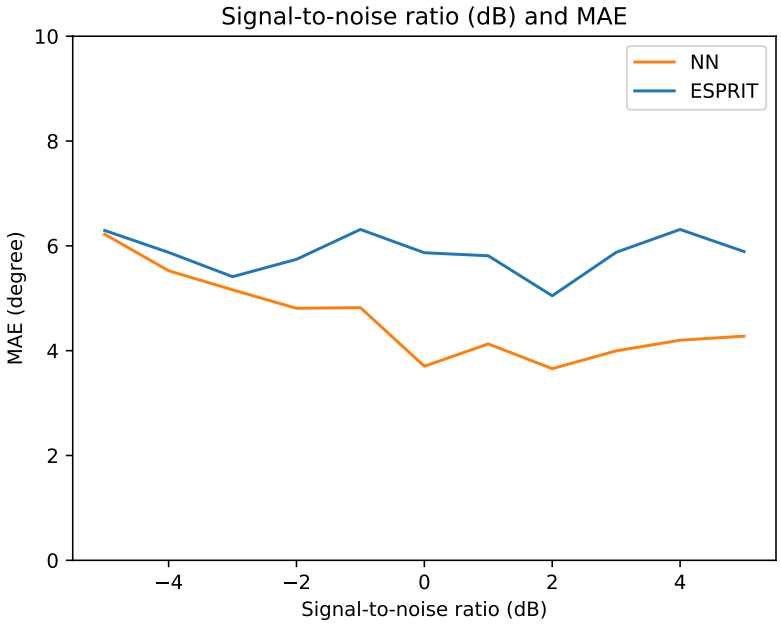

Figure 18: MAE of DOA estimation with 3 path

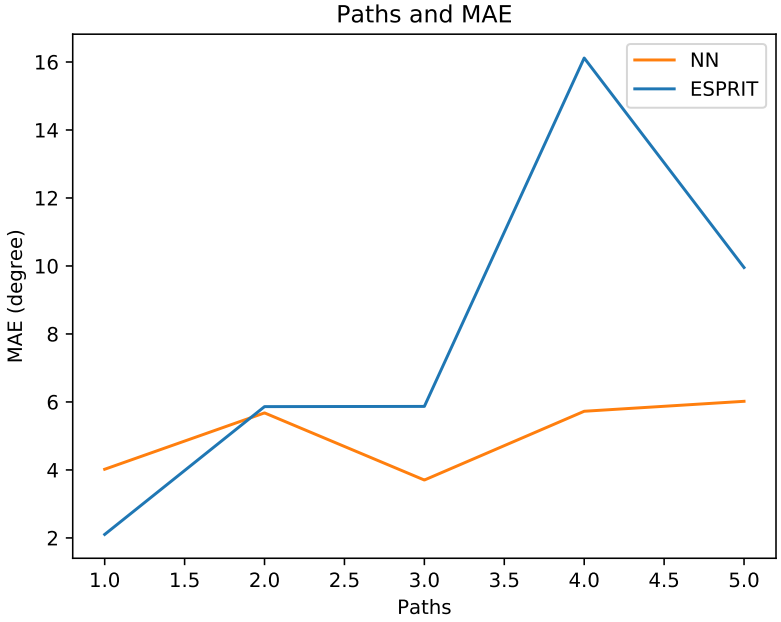

Figure 19: MAE of DOA estimation with SNR value of 0 dB

fewer elements.

7.2.2 Extrapolation of the number of paths

The second scenario measured the effect of extending the number of paths outside the ESPRIT definition zone to examine the limitation of the neural network.

From Figure 19, it can be seen that the number of paths has a strong effect on ESPRIT. When the amount of paths is small, ESPRIT performs better than the neural network. However, when the number of paths is over 3, the performance of ESPRIT decreases heavily.

On the other hand, the neural network performs more steadily than ESPRIT. The number of paths does not have a high effect on the estimations. The MAE of the neural network is constantly between 4 and 6 degrees. This means that the neural network is more reliable. It is more robust against the changes in the channel. However, the MAE is never excellent, meaning that the error between the correct angle and the estimated angle might be too high for cellular systems. Nevertheless, in most cases, the MAE of the neural network is better than the MAE of ESPRIT.

7.2.3 Generalization

The generalization was tested by running the simulation with channel properties that were not used in the training data. The testing data contained simulations, which were done by using lower SNR values than -5 dB and by using more than 5 paths.

Figure 20 shows the performance of both methods when the number of paths is 1 with low SNR values. It can be seen that when the MAE started to increase dramatically for the neural network and ESPRIT when the SNR value exceeded -5 dB. The MAE was lower for ESPRIT from 5 dB to -5 dB. For smaller than -5 dB SNR values, the estimations were more or less random.

When the same was tested with 3 paths seen in Figure 21, MAE started to increase at -5 dB for the neural network and at -10 dB for ESPRIT. However, ESPRIT was not able to yield accurate estimations at any point.

When the network model was tested with 6 paths seen in Figure 22, the accuracy of the neural network decreased slightly compared to the accuracy with the scenarios which were included in the training data. However, when the model was tested with 7 paths, the model was not able to resolve the correct DOA at all. This was seen in Figure 23

These observations indicate that the network model is not able to generalize properly for different channel properties. The performance was similar when the SNR value was extremely low compared to ESPRIT. When testing with path numbers which were not included in the training data, the performance of the network model decreased considerably. This might be due to overfitting or due to the large differences between the channel estimation data with different channel properties. The lack of ability to generalization is a serious problem. This is since in real-world the channel model is even more complex meaning that it is impossible to train the network for every scenario.

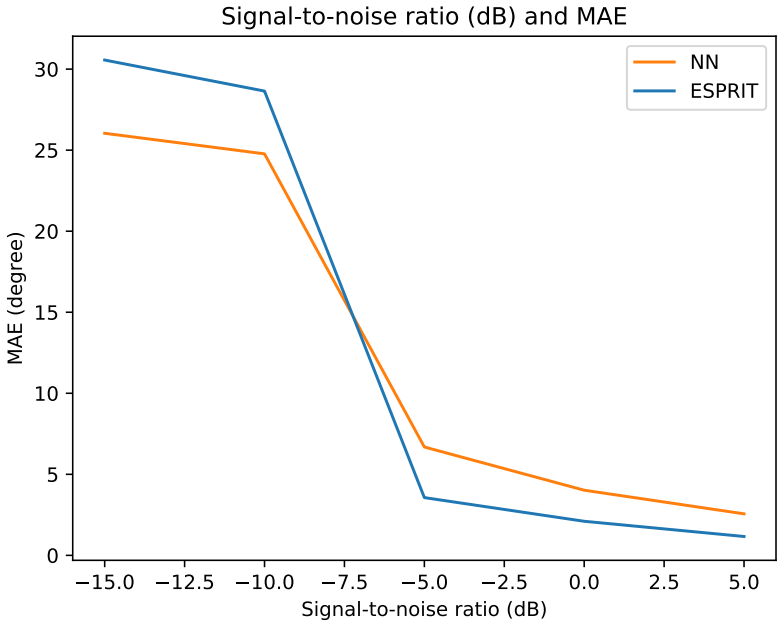

Figure 20: MAE of DOA estimation with 1 path and high noise

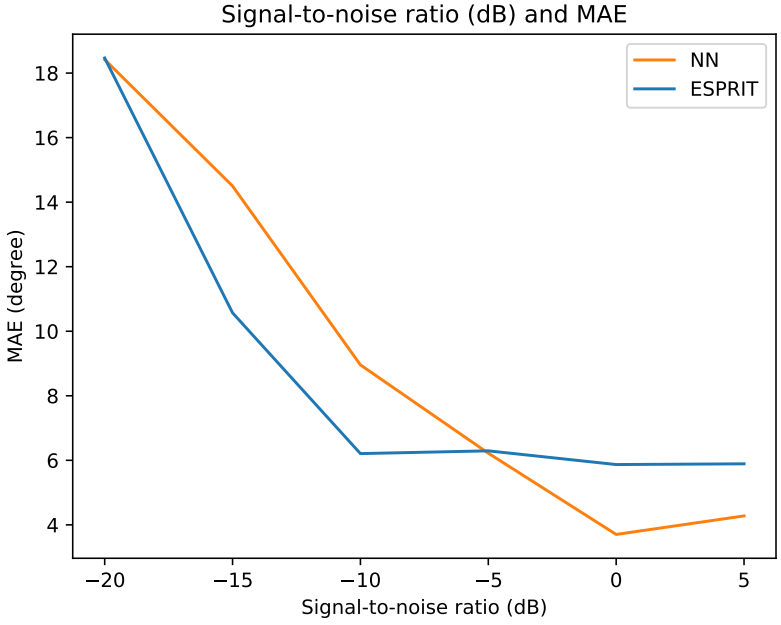

Figure 21: MAE of DOA estimation with 3 path and high noise

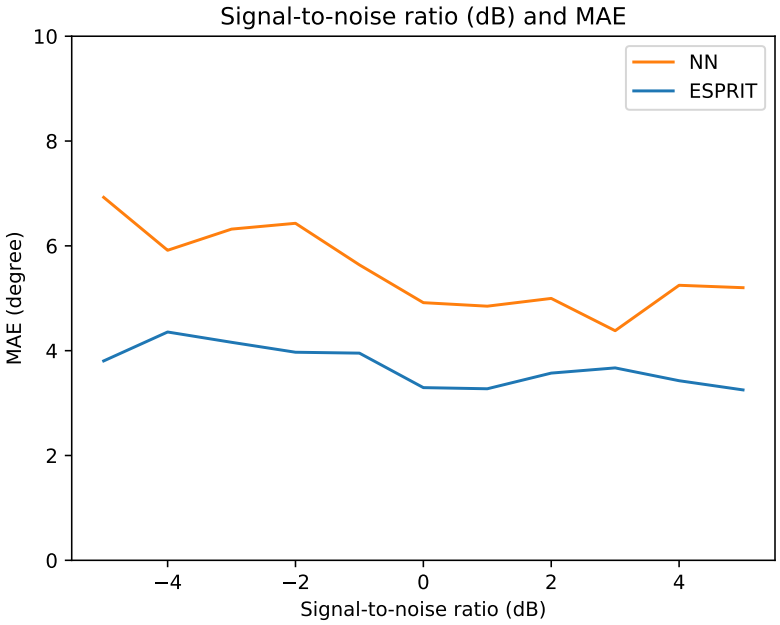

Figure 22: MAE of DOA estimation with 6 path

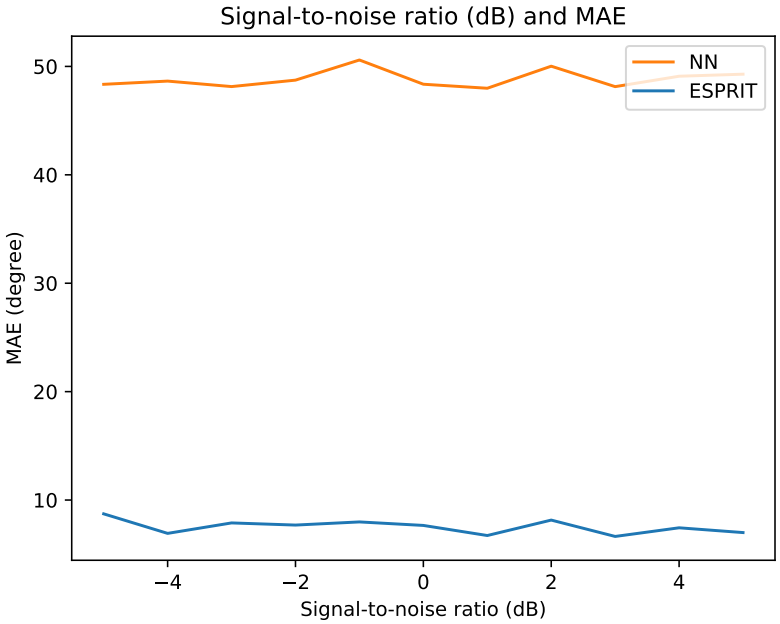

Figure 23: MAE of DOA estimation with 7 path

Table 5: The computational complexity of ESPRIT

Input to the algorithm θ, $a(\theta)$		
No	**Operation Performed**	**Complexity**
1	Averaging the channel estimation matrix	$O(M)$
2	Estimation of Correlation matrix	$O(M^3)$
3	SVD of correlation matrix	$O(M^3)$
4	Selecting the eigenvector corresponding to the highest eigenvalue	$O(M)$
5	Using LS to solve the Ψ	
6	Obtain estimate of Ψ	
	Total	$O(M) + O(M^3) + O(M^3) + O(M)$

Table 6: The computational complexity of the neural network

Input to the algorithm: $n = M \times N \times 6 \times 2$ data points of a channel estimation.		
No	**Operation Performed**	**Complexity**
1	Feeding n data points to the nodes in the first hidden layer	$h_1O(n)$
2	Feeding h_1 data points to the nodes in the second hidden layer	$h_2O(h_1)$
3	Feeding h_2 data points to the node in the output layer	$mO(h_2)$
	Total	$h_1O(n) + h_2O(h_1) + mO(h_2)$

7.2.4 Computational complexity

There are many differences in the steps of the DOA estimations between the traditional method and the neural network. This causes them to perform differently since they contain different amounts of operations. The computational complexity comparison is done with a single channel estimate, thus the complexity of a single DOA estimation is measured.

The neural network contained multiple activation functions in parallel and serial. This means that the DOA estimation had multiple multiplications, summations, and evaluations. On the other hand, the traditional method required covariance matrix computation, SVD and a LS method calculation.

By following the Table 6, the complexity of the neural network with $n = 384$ inputs, $h_1 = 30$ and $h_2 = 20$ nodes in the hidden layers and $m = 1$ output can be calculated as follows. In a fully-connected layer, all inputs are connected to all outputs. The first layer contains n inputs and h_1 outputs. Hence, the weights $\mathbf{W}$ can be stored in a $n \times h_1$ matrix. The fully-connected layer performs a computation of

$$y = \text{matmul}(x, \mathbf{W}) \tag{92}$$

where matmul stands for matrix multiplication function which is simply multiple dot

products. In (92), x is a vector of n input values, $\mathbf{W}$ is the $n \times h_1$ matrix, which contains the weights of the layer. The output y is a vector of h_1 which contains the output values of the layer. In addition, the ReLU activation function is a single operation and it is applied to the output of the layer. Therefore, feeding a node with k inputs requires k multiplications, $k+1$ additions and 1 evaluation of the ReLU function. This is proportional to k and therefore $O(k)$. Feeding the network takes first feeding h_1 nodes with n inputs each, hence $h_1O(n)$. Next, the second hidden layer with h_2 nodes takes h_1 inputs each, hence $h_2O(h_1)$. Lastly, the output layer with m nodes takes h_2 inputs, hence $m(h_2)$. Thus,

$$\text{Feed time} = h_1O(n) + h_2O(h_1) + mO(h_2) \tag{93}$$

The calculation of the complexity of the traditional method is more complex as compared to the neural network. The complexity comes from the different steps to calculate the DOAs of each signal component. The steps can be seen in Table 5. These steps consider only the most dominant steps meaning that the steps that consume most computations are presented.

The first step is the covariance matrix calculation. The matrix is calculated by using the same channel estimates which were used as the data points of the neural network. However, to minimize the complexity, the channel estimates are averaged over the PRBs. Since a SRS is sent over N PRBs, each antenna element calculates the mean over every PRB and sets this mean as a channel estimate per PRB. This means that each element forms a vector of size N where the value of each vector element is the mean. Hence, the averaged channel estimation matrix contains $N \times 6$ additions and 1 division, thus $6N + 1$ operations for each element. This means that the complexity is MN. Lastly, the covariance matrix is further averaged so that the final output of the M element antenna array is an $M \times M$ average channel estimation matrix. The calculation of a covariance matrix can be seen in Figure 24

Once the channel estimation is done, the covariance matrix is performed with a matrix multiplication between the averaged channel estimation matrix and its Hermitian version. According to [38], multiplication of two square matrices $\mathbf{A} \times \mathbf{A}$, $\mathbf{A} \in \mathbb{C}^M$ has a complexity of $O(M^3)$.

The next step is the eigendecomposition, which was done by using SVD. Measuring the complexity of SVD is not straightforward since the SVD can be done by using different algorithms. The choice of which algorithm to use depends on the use case and how many singular values are desired to be computed. It is assumed, that the system wants to calculate every singular value, thus, the complexity is $O(M^3)$ for matrix $\mathbf{R} \in \mathbb{C}^M$ [39].

The final step is to obtain the rotating operator Ψ. This is done with the LS method introduced in Section 4.3.3. Since only the eigenvector corresponding to the highest eigenvalue is relevant, the input of the LS method is a vector size of $M \times 1$. Therefore the complexity is $O(M)$.

As can be seen, the neural network has a complexity order of $NM \times h_1 \times h_2$ whereas the ESPRIT has an order of M^3. This means that as the number of antenna elements increases, the complexity of ESPRIT grows as a cubic. The growth of the complexity of the neural network is linear if the number of PRBs and the number of

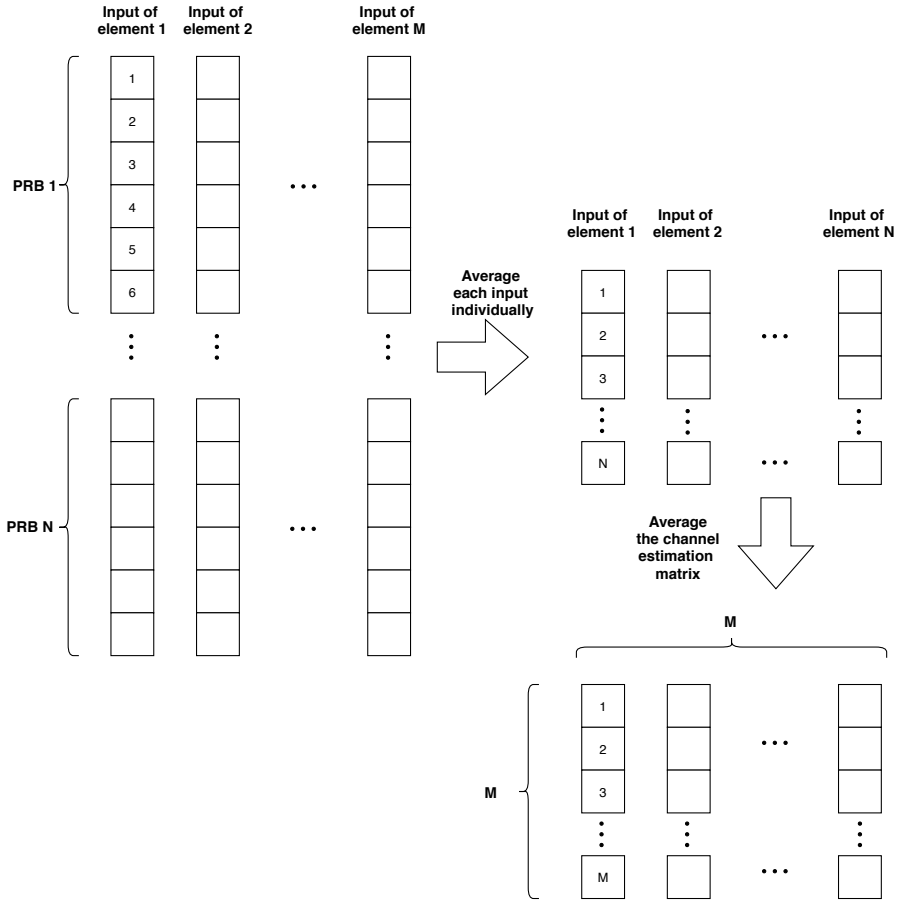

Figure 24: Averaging the channel estimation

nodes in each hidden layer is constant. Therefore, if the number of antenna elements is large, the ESPRIT requires more resources to estimate the DOA.

As seen in Figure 25, the neural network is less expensive when $M \geq 38$. Since the number of antenna elements can be large in massive MIMO systems to receive accurate estimations, it can be stated that the neural network is less complex in MIMO systems. Further, it should be noted that it is possible to also average the channel estimation for the neural network to decrease the number of inputs for the input layer. However, by averaging the channel estimations, there is a loss in the information, thus, the accuracy would decrease. Since the accuracy was already moderate for the neural network, averaging the channel estimations was not recommendable.

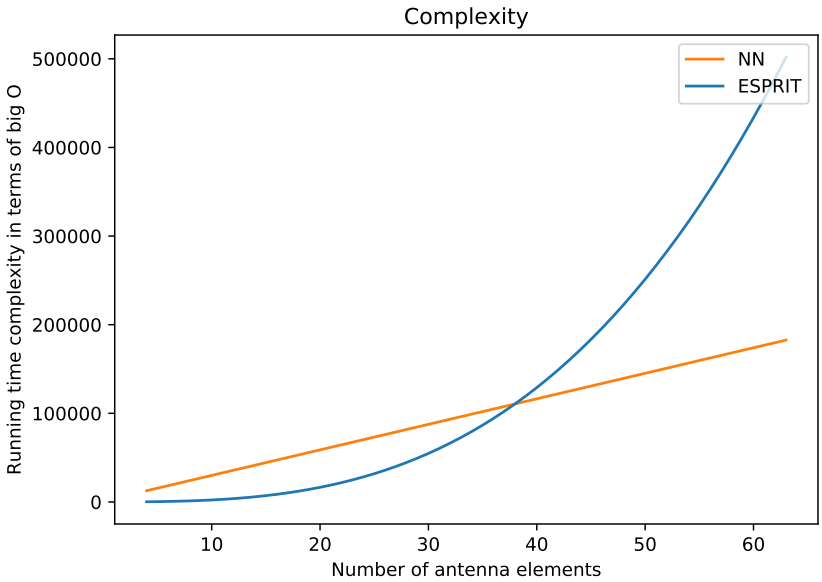

Figure 25: Running time complexity

8 Conclusions

The objective of the thesis was to examine if a neural network could be used to solve the DOA problem for downlink beamforming. The DOA was calculated by utilizing channel estimation information done with SRS. The chosen structure of the neural network was introduced in Chapter 6. This network was chosen by following the performance indicators defined in the same chapter. In Chapter 7, the neural network was compared to the traditional DOA estimation method, ESPRIT. In this chapter, the main results are summarized from Chapter 7. In the end, the chapter gives a view of what kind of future work could be done related to the subject of this thesis.

Traditional methods for DOA can be exhausting for the network and may have limitations corresponding to the number of DOAs they can distinguish. Accordingly, it was examined how neural networks can be used in the DOA estimation problem.

The examination was approached by first defining the crucial minimum requirements for the network. The requirements contained the accuracy of the estimation and the computational complexity of the network. Also, the generalization of the network model was taken into account. By following these requirements, an FNN type of network was chosen and the structure and the hyperparameters of the network were optimized by using sets of training data and trial and error. After the training, the network was tested with testing data.

The optimized network was tested in parallel with ESPRIT. The testing and training datasets were generated with a Matlab simulation. The testing data contained data that was not seen by the network before to test the generalization of the network model. The accuracy was measured by using Mean Absolute Error between the correct DOA and estimated DOA. The computational complexity was computed by calculating the growth rate of each step in both methods with the big O notation. This only gave a rough estimation of the complexity.

The observed results showed that the neural network was able to estimate the DOA with moderate accuracy. The accuracy was better than the accuracy of ESPRIT in some channel conditions but worse in others. Further, the generalization of the network model was not satisfying which is a serious problem in terms of replacing traditional methods in DOA estimation. However, with a large number of antenna elements, the neural network was less complex than ESPRIT. This is important since massive MIMO systems use multiple antenna elements.

These results indicated that the DOA estimation problem can be solved by a neural network. However, to receive sufficiently good results for a cellular network, further research is required related to the optimization and generalization of the network model.

To validate the benefits of a neural network, more testing is required for accuracy and generalization. Due to the limitations of the simulation environment, the number of used antenna elements in this thesis was unusual for a 5G cellular network. This caused the DOA information to be limited for the neural network and ESPRIT. By increasing the number of elements in the antenna array, the accuracy of the network should increase. This would allow averaging the channel estimations to decrease the

number of inputs for the network and thus, lower the complexity.

The generalization problem should be investigated in detail. The training data could be more diverse to make the network model more universal. Also, the use of regularization or dropout functions in the network could lower possible overfitting. Also, the network should be tested with different cell and SRS configurations and the performance of a neural network should be also empirically studied on real-life hardware.

Another subject for future research could be the use of raw symbol data rather than channel estimation data. This would lower the complexity even more since heavy calculations for the channel estimation would not be required. The problem with this approach is that the raw symbol data consist of multiple users and thus there should be a solution for user separation in the neural network. This user separation solution might, however, be exhausting for the system.

The last future research subject is to use different neural network types, such as CNN and RNN. A CNN might decrease the error of the network model but it can be difficult to manage the complexity on an appropriate level. On the other hand, RNN might decrease the error and complexity of the network model but this needs proper testing.

References

[1] M. A. Jensen and J. W. Wallace, "MIMO wireless channel modeling and experimental characterization," in *Space-time processing for MIMO communications*. Wiley Online Library, 2005, pp. 1–39.

[2] Z. Chen, Y. Yu, and G. Gokeda, *Introduction to direction-of-arrival estimation*. Artech House, 2010.

[3] E. C. Jordan and K. G. Balmain, *Electromagnetic waves and radiating systems*. Prentice-Hall, 1968.

[4] M. S. Neiman, "The principle of reciprocity in antenna theory," *Proceedings of the IRE*, vol. 31, no. 12, pp. 666–671, Dec 1943.

[5] E. Dahlamn, S. Parkvall, and J. Sköld, *4G: LTE/LTE-Advanced for Mobile Broadband*. Oxford: Academic Press., 2011.

[6] P. W. Chan, E. S. Lo, R. R. Wang, E. K. Au, V. K. Lau, R. S. Cheng, W. H. Mow, R. D. Murch, and K. B. Letaief, "The evolution path of 4G networks: FDD or TDD?" *IEEE Communications Magazine*, vol. 44, no. 12, pp. 42–50, Dec 2006.

[7] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017. [Online]. Available: http://www.nowpublishers.com/article/Details/SIG-093

[8] E. Dahlman, S. Parkvall, and J. Sköld, "4G LTE-Advanced pro and the road to 5G (third edition)," E. Dahlman, S. Parkvall, and J. Sköld, Eds. Academic Press, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128045756000017

[9] R. Zhang, S. Wang, X. Lu, W. Duan, and L. Cai, "Two-dimensional DOA estimation for multipath propagation characterization using the array response of PN-sequences," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 341–356, Jan 2016.

[10] C. L. Ying and P. S. C. Francois, "Downlink beamforming method," 2004.

[11] T. Asté, P. Forster, L. Fety, and S. Mayrargue, "Downlink beamforming avoiding DOA estimation for cellular mobile communications," in *IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 6, 1998, pp. VI–3313.

[12] M. Li and Y. Lu, "Maximum likelihood DOA estimation in unknown colored noise fields," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1079–1090, July 2008.

[13] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[14] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[15] Y. Jiang, P. Stoica, Z. Wang, and J. Li, "Capon beamforming in the presence of steering vector errors and coherent signals," in *Proceedings of the 11th Annual Workshop on Adaptive Sensor Array Processing, Lexington, MA, USA*, 2003, pp. 11–13.

[16] G. H. Golub and C. Reinsch, *Singular Value Decomposition and Least Squares Solutions*. Berlin, Heidelberg: Springer, 1971, pp. 134–151. [Online]. Available: https://doi.org/10.1007/978-3-662-39778-7_10

[17] J. He and Z.-F. Fu, "2 - mathematics for modal analysis," in *Modal Analysis*, J. H. and Z.-F. F., Eds. Oxford: Butterworth-Heinemann, 2001, pp. 12 – 48.

[18] J. Litva and T. Lo, *Digital Beamforming in Wireless Communications*, ser. Artech House mobile communications series. Artech House, 1996.

[19] B. Sharma, G. Singh, and I. Sarkar, "Study of DOA estimation using MUSIC algorithm," *International Journal of Engineering Science*, vol. 6, no. 7, pp. 594–603, 2015.

[20] A. Zekeriya, H. K. Hwang, G. Marshall, and A. Yakovlev, "Sensitivity analysis for direction of arrival estimation using a root-MUSIC algorithm," *Engineering Letters*, vol. 16, 08 2008.

[21] W. F. Keith, "Utilizing waveform features for adaptive beamforming and direction finding with narrowband signals," *Lincoln Laboratory Journal*, vol. 10, no. 2, 1997.

[22] P. W. Njeri, D. B. O. Konditi, and K. L. Philip, "Performance analysis of MUSIC, root-MUSIC and ESPRIT DOA estimation algorithm," *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 8, no. 1, pp. 209–216, 2014.

[23] M. P. Priyadarshini and R. Vinutha, "Comparative performance analysis of MUSIC and ESPRIT on ULA," in *2012 International Conference on Radar, Communication and Computing (ICRCC)*, Dec 2012, pp. 120–124.

[24] G. H. Golub and C. F. Van Loan, "An analysis of the total least squares problem," *SIAM journal on numerical analysis*, vol. 17, no. 6, pp. 883–893, 1980.

[25] A. J. Maren, C. T. Harston, and R. M. Pap, *Handbook of neural computing applications*. Academic Press, 1990.

[26] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015. [Online]. Available: http://neuralnetworksanddeeplearning.com

[27] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[28] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 315–323. [Online]. Available: http://proceedings.mlr.press/v15/glorot11a.html

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

[30] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, "Dataset shift in machine learning," 2009.

[31] S. Saeb, L. Lonini, A. Jayaraman, D. C. Mohr, and K. P. Kording, "Voodoo machine learning for clinical predictions," *bioRxiv*, 2016. [Online]. Available: https://www.biorxiv.org/content/early/2016/06/19/059774

[32] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[33] T. Wang, C. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Communications*, vol. 14, pp. 92–111, 10 2017.

[34] C. Liu, Y. Wu, Y. Lin, and S. Chien, "Computation-performance optimization of convolutional neural networks with redundant kernel removal," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2018, pp. 1–5.

[35] Z. Ye, A. Gilman, Q. Peng, K. Levick, P. Cosman, and L. Milstein, "Comparison of Neural Network Architectures for Spectrum Sensing," *arXiv e-prints*, p. arXiv:1907.07321, 2019.

[36] L. S. Ezema and C. I. Ani, "Artificial neural network approach to mobile location estimation in GSM network," *International Journal of Electronics and Telecommunications*, vol. 63, no. 1, pp. 39–44, Mar. 2017. [Online]. Available: https://doi.org/10.1515/eletel-2017-0006

[37] *Universal Mobile Telecommunications System (UMTS);LTE; Universal Terrestrial Radio Access (UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRA); Verification of radiated multi-antenna reception performance of User Equipment (UE)*, ETSI, 2014.

[38] D. S. Watkins, *Fundamentals of matrix computations*. John Wiley & Sons, 2004, vol. 64.

[39] H. Qian. (2015) Counting the floating point operations (FLOPS). [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/50608-counting-the-floating-point-operations-flops