Machine Learning Technique for Beam Management in 5G NR RAN at mmWave Frequencies

GUSTAV FAHLÉN & JOEL BILL

MASTER´S THESIS

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Machine Learning Technique for Beam Management in 5G NR RAN at mmWave Frequencies

Gustav Fahlén & Joel Bill
elt14gfa@student.lu.se
elt14jbi@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisors: Ove Edfors, LTH
Irfan Baig, Ericsson Lund

Examiner: Fredrik Rusek

November 28, 2019

Printed in Sweden
Tryckeriet i E-huset, Lund

Popular Science Summary

The mobile industry is looking to achieve much higher data rates when making the transition from the fourth generation (4G) of mobile technology to 5G. One way of accomplishing this is by using the frequency spectrum where the frequencies have a wavelength of millimeter length, the so called millimeter wave spectrum. This spectrum was previously not used because signals on these frequencies are difficult to deal with, but with more advanced techniques it has become a possibility. One of these advanced techniques that enables the usage of this spectrum is called beamforming and is based on focusing the transmitted signals in a more narrow direction than what was done earlier. These focused signals are called beams. The cell tower can modify the signals that it sends out so that they are focused in the direction of where the intended receiving cell phone is positioned. Due to the narrow beams, the cell phone's movements needs to be taken care of and the direction of the transmitted beams needs to be updated. To help the cell tower in selecting which direction to transmit in, the cell phone can report channel measurements on a set of beams and tell on which one it experiences the best signal quality. Therefore, as long as the cell tower provides a set of beams where at least one beam results in the cell phone experiencing good signal quality, the movement of the cell phone can be handled and good signal quality can be sustained. If the cell tower selects wrong beams in the set, it results in the cell phone experiencing non-optimal or bad signal quality.

With this in mind, the goal of this thesis has been to select a set of beams for a moving cell phone to measure signal quality on and that results in the cell phone experiencing good signal quality. The selection of beams has been done by using a machine learning algorithm. Machine learning is a sub-field of artificial intelligence and with the help of machine learning the algorithm could experiment with different beams in the set and observe if the selected beams gave a good result or not. It could then learn from its decisions to get better over time and eventually only select beams that had been proven successful in the past.

The machine learning algorithm's performance was then compared to an already existing algorithm for selecting the beams and the results shows that the machine learning algorithm can reach significantly better performance. The two algorithms have been compared by looking at how good throughput the cell phone experiences, which is a measurement of how well the signal quality is.

Abstract

Ericsson has an interest in investigating if the fast-growing concept known as machine learning can be applied to beam management, in a 5G NR environment using mmWave frequencies. Because of the high path-loss at mmWave frequencies and high throughput demands of 5G NR systems it is crucial to the UE to always stay connected to the most suitable beam, to provide highest possible throughput. To obtain the required fine alignment of each single beam, optimization of beam management operations, such as beam tracking is essential.

The type of machine learning algorithm used is called reinforcement learning. The algorithm will aim to always connect the UE to the most suitable beam - by comparing RSRP values from a selection of beams, that are picked based on the current serving beam. The machine learning algorithm will initially pick a candidate beam set based on baseline (which is explicitly programmed), however after multiple iterations, when the algorithm is considered experienced, decisions will instead be based on machine learning.

The algorithm will be trained from scratch over 50 different seeds i.e. 50 different environments with different properties to increase the reliability of the performance of the machine learning. The performance of the machine learning algorithm will be evaluated by comparing the cell downlink throughput of machine learning and baseline.

When reviewing the result, it is clearly illustrated that reinforcement learning can be applied to beam management in mmWave environment to boost the average cell downlink throughput compared to baseline.

Acknowledgements

First of all we would like to thank the Ericsson office in Lund for letting us write our thesis there. Helpful, supportive and interested colleagues has made this a great place to write a master's thesis at. Secondly, we would like to thank Ove Edfors and Irfan Baig who have been our supervisors at LTH and Ericsson, respectively. We would also like to send out a special thanks to Niclas, Emil, Jonas and Staffan in the Systems team for helping us along the road and showing a great interest in what we were doing. Furthermore we would like to thank all the other thesis students at Ericsson Lund for the great atmosphere and the delicious Friday fika.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

The demand for high speed mobile broadband has been one of the key aspects in the transition from fourth generation (4G) mobile technology to 5G. To meet these higher user throughput demands, the millimeter wave (mmWave) spectrum with its huge bandwidths has been considered as an enabler of the higher data rates requirement specified for 5G. However, mmWave frequencies makes the channel conditions highly vulnerable to propagation loss, especially when the distance between transmitter and receiver increases. To deal with this, beamforming will be an important and necessary tool to use. The basic idea of beamforming is to concentrate the transmission signal from the antenna in the direction of the intended receiver and therefore significantly improving the received signal power. This is done by altering the phase of the signals from each antenna element in a way that the signals add up constructively in the intended direction and destructively in other directions. To establish and retain a suitable beam for an intended receiver, the beams require fine alignment. This is achieved through a set of operations collectively known as beam management which includes beam establishment, beam refinement and beam tracking. Achieving perfectly aligned beams between transmitter and receiver requires an intelligent beam tracking algorithm that in an efficient way can select the most suitable beam based on the mobility of the receiver.

1.2 Purpose

The purpose of this thesis is to investigate how well machine learning, more specifically reinforcement learning, can be used to find the most suitable beams in the beam tracking process. The proposed machine learning algorithm will be evaluated in comparison to an already existing beam tracking method, which further on will be referred to as the baseline algorithm.

1.3 Problem Formulation

Ideally, beam tracking would be solved by an algorithm that always assigns a beam to the user equipment (UE) where it experiences the best possible channel quality. Preferably this should be done while minimizing the number of measurements and only performing channel quality measurements on a small number of beams to reduce the power consumption. However, trying to optimize all these parameters lies outside the scope of this thesis.

The problem in this project was limited to predicting a set of beam candidates that Channel State Information Reference Signals (CSI-RS) should be measured on by the UE. The predicted set should have one or more beams that gives high Reference Signal Received Power (RSRP) when the measurement is reported back to the base station. The beam with the best reported RSRP is then expected to be a good choice to perform a beam switch to. CSI-RS and RSRP are two well known quantities in the standardization of mobile technology done by 3GPP [1].

1.4 Previous Work

In recent years, the interest in machine learning has significantly increased. Mnih et al [2] used a variant of Q-learning in combination with a neural network to train an agent to play seven Atari 2600 games. The method achieved great results, outscoring a human expert on three of the games. Machine learning has also gained positive results in e.g. speech processing and computer vision [3]. Thanks to these successes, the possibility of implementing machine learning in telecommunication has started to be investigated.

Ekman [4] used supervised learning to try to optimize handovers between base stations by finding the target beam with the highest possible RSRP. The report shows that 25 candidate beams were needed to achieve a 90 % beam-hit-ratio, i.e. the best possible beam is selected, and around 15 candidate beams were needed to achieve a 90 % sector-hit-ratio (the best sector is selected).

Similarly, Bonneau [5] investigated if reinforcement learning could be used for handovers in a 5G system. The proposed algorithm tried to optimize the trade-off between signal quality, number of measurements to find a better beam and the number of handovers. Even though promising results were obtained in a small scale system, long computation times prevented the method from being successful on a larger scale.

Klautau et al [6] primarily focused on generating realistic data sets that can be used for deep learning (a subcategory of machine learning) based problems that are related to mmWaves and beam management. The proposed data set was then applied to let deep learning predict the beam selection in a vehicle-to-infrastructure (V2I) 5G environment. However, it was considered to be out of the scope of the paper to investigate the performance of different deep learning architectures.

With the previous mentioned reports in mind, it has not yet been thoroughly investigated how well reinforcement learning can be used to handle the beam tracking procedure. Hence, in this thesis we apply our implementation of reinforcement learning and analyze how well it performs in comparison to already existing methods.

1.5 Delimitations

There are multiple questions and topics that can be discussed in future related studies that were not covered in the scope of this thesis. One delimitation is the concept of wide beams. Beam management in 5G networks will include wider beams that a UE measures channel quality on and connects to. Once a selection of a wide beam has been made, channel quality measurements of more narrow beams that are mapped to the selected wide beam are measured and reported by the UE. The best reported narrow beam is then selected by the base station. However, in this thesis only switches between narrow beams are considered, and thus the step of connecting to a wider beam is left out.

The simulation environment was limited to one cell containing one base station. Therefore handovers between multiple base stations were not considered in this work.

Another delimitation is that the optimal size of the candidate beam set was not explored. To optimize the power consumption, it is possible that fewer beam candidates could be used and still achieve the same results. If the algorithm is intelligent enough it is reasonable to believe that the best beam candidate can be found as many times as the algorithm with more beam candidates.

Finally, the optimal transition for when the algorithm is trained enough, and should increase the number of decisions taken based on previous experiences was not investigated. Instead, the timing of this transition was decided based on obtained observations of how quickly the algorithm learned over time.

1.6 Disposition

Introduction

Gives the reader a short introduction including background, motivation and purpose for the thesis.

Background

Describes the background more in depth and explains some of the key features that are expected to play an important role in the transition from LTE to NR.

Telecommunication Theory

Explains the procedure of signal quality measurements between UE and base station. Also briefly mentions orthogonal frequency-division multiplexing (OFDM).

Machine Learning

Describes the theory behind reinforcement learning and further explains how it is implemented in this thesis.

Method/Simulation Overview

Contains information about the simulation environment, the chosen parameter setup and the available data.

Results

Results of how the algorithm is performing in the different scenarios.

Discussion

Thoughts about how the algorithm possibly could be further improved and future work.

Chapter 2

Background

2.1 5G NR

The applied 5G technology (3GPP Release 15)[1] is based on today's well-known LTE technology, but with the difference that 5G aims to satisfy much higher demands on different categories. For instance, 5G will result in much higher end-user data rate, lower latency and lower power consumption. To meet the requirements set and to allow potential future development the new radio access technology, NR is used. 5G NR features many new and improved LTE technologies such as Massive MIMO, millimeter waves (mmWave) and beam management [7][8, pp. 4-6].

2.2 mmWave

MmWave frequencies (technically in the 30 to 300 GHz spectrum but often referred to the 3GPP Frequency Range 2 (FR2) which includes frequencies from 24.25 GHz to 52.6 GHz) is one of the new features in 5G NR that can contribute to both higher bandwidth and bit rate. However, mmWave require new and more advanced technology to perform on the expected level, because of the short wavelength. The link between two antennas using mmWave frequencies suffer from high path-loss, severe channel intermittency and are blocked by various obstacles. Moreover, to handle these issues directional transmission links are required, which can be achieved from the technique known as beamforming. However, to achieve this, fine beam alignment is required through usage of a couple of operations known as beam management [9].

2.3 Beamforming

Beamforming is a technique used for high frequencies (mmWaves) in NR for directional signal transmission. This is achieved by combining elements in a phased array in such a way that some signals will experience constructive interference, and some will experience destructive interference [10]. By using a combination of elements transmitting signals of different phases, all signals will add up to one focused directional beam. If the direction is set towards a UE, it is a key-enabler to counteract the big path-loss obtained in the antenna links using mmWaves [11].

Different techniques are used to achieve different results of beamforming, namely analog beamforming (ABF), digital beamforming (DBF) and hybrid beamforming (HBF).

2.3.1 Analog Beamforming

In fully implemented ABF, the entire antenna array is connected to one radio frequency (RF) chain. This technique is a simple and effective way to generate high beamforming gains from a large number of antennas, but with less flexibility. Since ABF only use a single RF chain, it only allows one communication beam at a time, hence resulting in decreased throughput [12].

2.3.2 Digital Beamforming

In fully implemented DBF, every antenna element is connected to one separate RF chain. This technique offers high performance and high degree of freedom, but with a drawback. Each RF chain requires separate FFT/IFFT blocks, digital-to-analog converters and analog-to-digital converters, which increases cost and complexity of the system significantly [11].

2.3.3 Hybrid Beamforming

HBF is a combination of digital and analog components, which provides the possibility to use multiple RF-chains, but much fewer than the number of antenna elements. Therefor this technique will provide higher performance than ABF, and with less complexity as DBF [13].

2.4 Beam Management

To be able to make a fine alignment of receiving and transmitting beams, a set of operations called beam management is performed. This set of operations consists of beam establishment, beam refinement and beam tracking and the goal is to always obtain the optimal beam for the UE, i.e. the beam with the best channel quality [9].

2.4.1 Beam Establishment

Beam establishment includes the procedures that describe how a beam pair is initially established in the downlink and uplink transmission directions. A connection is then set and if communication continues it can be assumed same beam will be used to transmit data [8], pp 332.

2.4.2 Beam Refinement

Beam establishment is applied for wide beams, however after establishment it is preferred to refine the beam shape. For instance, this is done to make the beam more narrow compared to the wide beams used for initial beam establishment [8], pp 243.

2.4.3 Beam tracking

Beam tracking is an operation that handles beam switches both along the vertical and horizontal axis by utilizing a two-dimensional antenna array [14]. The main purpose of beam tracking is to find the best possible serving beam among a set of narrow beams, which is decided based on reference signals [8], pp 243.

Chapter 3

Telecommunication Theory

This chapter explains the theory behind the signal quality measurements that takes place in wireless communication systems. Both how it works in present 4G systems and how it is expected to be in the next generation of wireless networks is explained. The theory is provided for the reader to get a better understanding of how our proposed solution works in detail.

3.1 OFDM

The transmission scheme used in downlink in LTE is OFDM. In uplink, DFT-spread OFDM is used which is also based on OFDM techniques [15], pp 31. Because of its robustness to time dispersion and usage of both time and frequency domain when defining signal and channel structure, it was found to be a suitable candidate for 5G as well [8], pp 61. In OFDM, a frequency band is divided into multiple orthogonal smaller frequency bands called subcarriers that each carry parts of the transmitted data [16]. Furthermore, the subcarriers are separated in such a way that at each sample point in the frequency domain only one of the subcarriers has a non-zero value, which is illustrated in Figure 3.1. This makes the subcarriers independent and they do not influence one another, they are so called orthogonal. Besides the unwanted phenomenon called inter carrier interference, the problem that a delayed OFDM symbol can overlap with an adjacent symbol also exists. This occurs in the time domain and is called inter symbol interference. It is countered by having a guard interval, or time gap, between the symbols. The guard interval is filled with a copy of the last part of the symbol and is called Cyclic Prefix (CP).

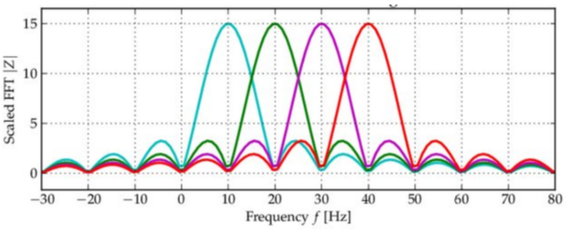

Figure 3.1: OFDM subcarriers [17].

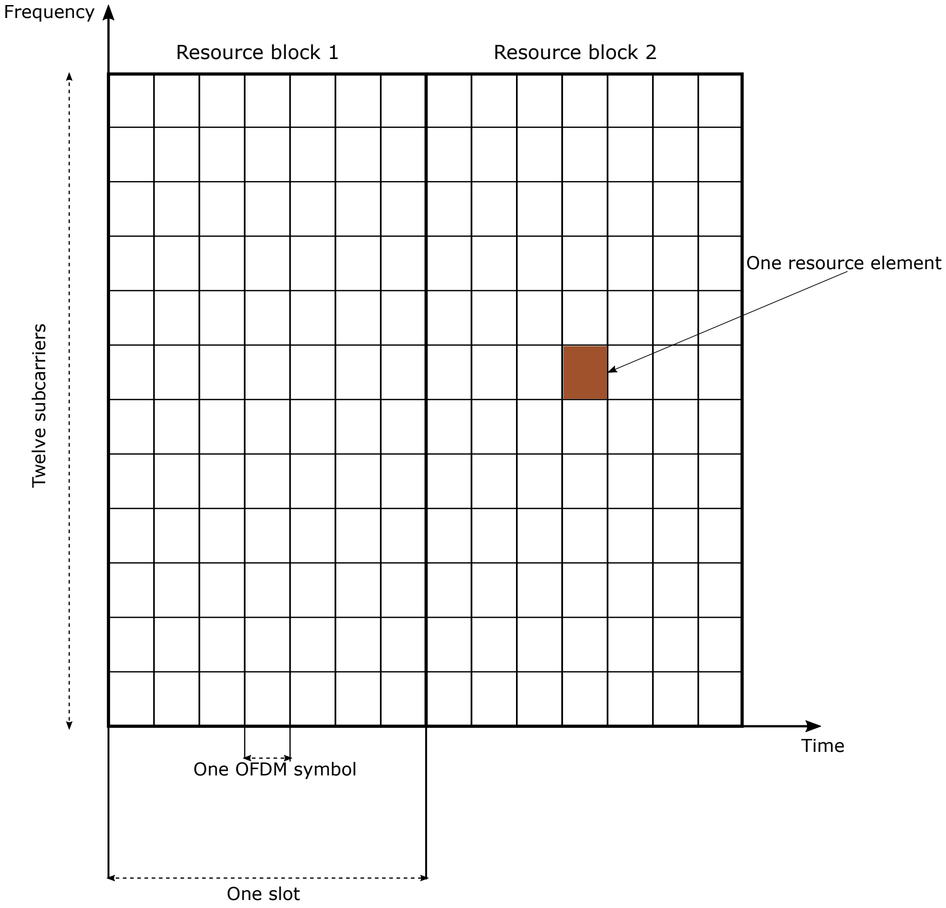

Figure 3.2: The LTE physical time-frequency resource.

3.2 Time-Frequency Structure

The LTE transmission resources can be visualized in a time-frequency grid, see Figure 3.2. In the time domain, transmissions are organized into frames of length 10 ms. Each frame is divided into ten subframes of length 1 ms. A subframe is further divided into two slots of length 0.5 ms. Finally, a slot is divided into a number of OFDM symbols which is the smallest unit in the time domain. A slot consists of seven OFDM symbols if normal CP is used in the OFDM symbols, or six if extended CP is used. In the frequency domain, the smallest unit is a subcarrier. The subcarrier spacing in LTE is 15 kHz. This was chosen because it was found to offer a good balance between frequency errors and unnecessarily large overhead from the cyclic prefix [8], pp 61. A resource element, consisting of one subcarrier during one OFDM symbol, is the smallest physical resource in LTE. Resource elements are grouped into resource blocks which consists of 12 subcarriers (frequency domain) and one slot (time domain). Finally, the minimum scheduling unit consists of two consecutive resource blocks within one subframe which is called a resource-block pair [15], pp 78.

A couple of things differ in NR. Unlike LTE with carrier frequencies up to only approximately 3 GHz, NR needs to support carrier frequencies varying from sub-1 GHz up to mmWave frequencies. Having one fixed subcarrier spacing for all of these different deployment scenarios is not possible, and therefore a range of spacings are supported. Changing the subcarrier spacing also leads to changes in the cyclic prefix (which also changes the time of an OFDM symbol) due to the nature of OFDM. The supported spacing in NR and corresponding cyclic prefix are shown in Table 3.1 [8], pp 105.

Subcarrier Spacing (kHz)	Cyclic Prefix (μs)
15	4.7
30	2.3
60	1.2
120	0.59
240	0.29

Table 3.1: Subcarrier spacing supported by NR.

In the time domain, NR uses frames of length 10 ms and subframes of length 1 ms just like LTE does. A subframe is then divided into slots consisting of 14 OFDM symbols each. Since the length of an OFDM symbol varies with the subcarrier spacing, so does the duration of a slot. The different slot durations are illustrated in Figure 3.3.

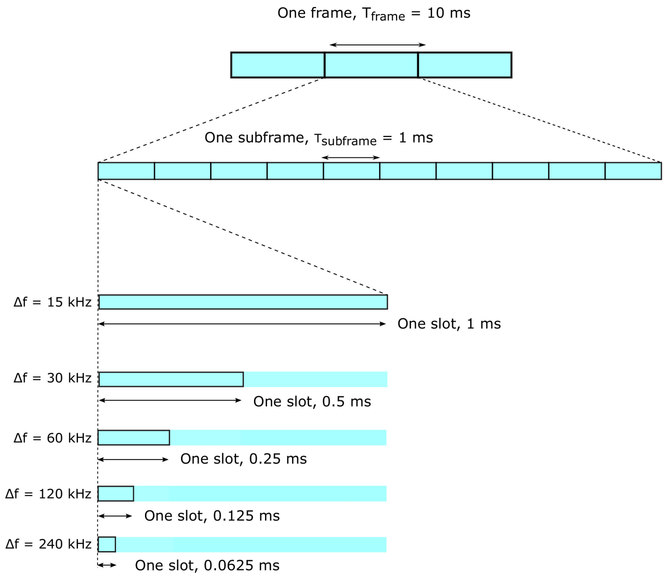

Figure 3.3: Frames, subframes and slots in NR.

In the NR frequency domain structure, just like in LTE, a resource element consists of one subcarrier during one OFDM symbol and is the smallest physical resource. However, unlike LTE a resource block is 12 consecutive subcarriers in the frequency domain. This is different from the LTE definition where it consists of 12 subcarriers in the frequency domain and one slot in the time domain. The reason why an NR resource block is defined in the frequency domain only is the flexibility in time duration for different transmissions which was not the case in the original LTE release [8], pp 109.

3.3 Reference Signals

For a UE to measure channel quality there are downlink reference signals that occupy specific resource elements in the time-frequency grid. In the first release of LTE, this was done with cell-specific reference signals (CRS). The structure of a single CRS is illustrated in Figure 3.4. A CRS occupy the first and seventh subcarrier during the first OFDM symbol and the fourth and tenth subcarrier during the fifth OFDM symbol. These resource elements are called reference symbols and have predefined values. When a UE tries to estimate the downlink power it measures the power of the CRS and reports the result to the base station. More specifically, it measures the linear average received power over the CRS specified resource elements. The measured result is called Reference Signal Received Power (RSRP) and is reported in the logarithmic scale power unit dBm.

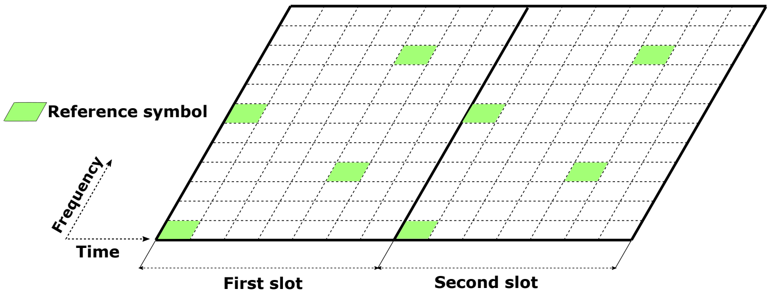

Figure 3.4: Structure of CRS within a resource block pair.

In LTE release 10 (released in 2011) the channel-state-information reference signal (CSI-RS) was introduced. CSI-RS will also be used in NR. It was initially introduced to support more advanced multiple-input and multiple-output (MIMO) techniques which was something that CRS could not do. CSI-RS usually occupies one, two or four resource elements depending on how many antenna ports that are used [8], pp 135. Multiple CSI-RS can be used to measure signal quality over a number of channels, or beams, where each CSI-RS is connected to a specific beam. The beams can be seen as focused, directed streams of data using beamforming techniques. Just like in LTE, it is the linear average over the power contributions of the resource elements that carry CSI reference signals that is measured. The UE measures the RSRP on each of the beams in the beam set and reports back to the base station. The measured RSRP values are used as support for beam management decision making.

Chapter 4

Machine Learning Approach

This chapter will begin with an introduction to machine learning and different common machine learning concepts. Focus will later shift to one of the main concepts of this thesis, namely reinforcement learning. Reinforcement learning can be done using several different types of algorithms, however this thesis will only investigate the algorithm called Q-learning.

4.1 Introduction to Machine Learning

Machine learning is said to be a subset of artificial intelligence with the ability to learn and improve based on experience without having to be explicitly programmed. This means that the computer can predict and decide what action to take depending on patterns or by reading big amounts of data. Machine learning is usually divided into three different sub-fields, supervised learning, unsupervised learning and reinforcement learning.

4.1.1 Supervised Learning

Supervised learning is based on mapping single inputs to outputs from big amounts of input and output data. The mapped data will be analyzed to produce a function with the ability to map new data.

4.1.2 Unsupervised Learning

Unlike supervised learning, unsupervised learning is used when only the input data is known. Since the learner is only given the inputs is the purpose of unsupervised learning to find underlying patterns among the input values itself. Usually this method will not provide as good result as supervised Learning, mainly since the outcome is unknown and it is thereby impossible to determine how accurate the found method is.

4.1.3 Reinforcement Learning

The goal of reinforcement learning is to find what action to take from different states based on a reward system. Reinforcement learning will fully trust the so-called reinforcement agent to make the best action based on earlier cumulative rewards.

4.2 Reinforcement Learning

The procedure of reinforcement learning can be described by Figure 4.1. The agent decides what action to take based on earlier experiences or sometimes random, based on a stochastic variable. An action (A_t) will affect the current state/position (S_t) in the environment followed by an announcement to the agent that the current state has been updated with (S_{t+1}). Moreover, the agent will also be announced with a reward value, that is either positive or negative based on the outcome of the current action. The reward R_{t+1} is thereby determined in association with S_t, A_t and S_{t+1}. All parameters are named based on the variable 't', which is a discrete time variable.

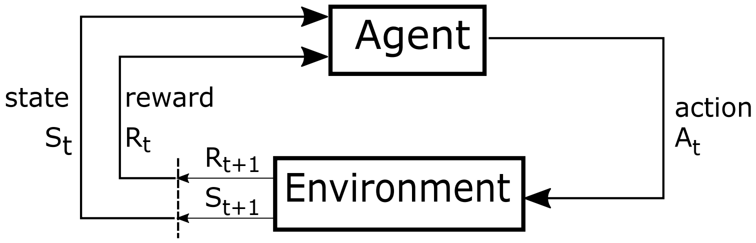

Figure 4.1: Block-scheme over Reinforcement learning process.

4.2.1 Q-learning

The Q-learning algorithm is used to obtain a high performance by validating each action taken. The validation is based on the accumulated previous outcomes of the current action (A_t) from the current state (S_t), best possible outcome from the next state (S_{t+1}) and a reward-value, that is set based on validation of the outcome of current action (A_t) from current state (S_t). The Q-learning algorithm uses (4.1) each time an action is performed to validate ((S_t), (A_t)) and to obtain a Q-value. Each calculated Q-value is then used to update an element in a two-dimensional matrix, called the Q-table. Multiple Q-table updates needs to be performed to get a reliable Q-table that consists of trustworthy values where all ((S_t), (A_t)) combinations have been explored. Therefore, the training will initially start with what is known as 'exploration' to explore the environment and learn which actions from which state are considered beneficial or not for a good performance. After some training the Q-learning algorithm will start to make a transmission towards more 'exploitation' of Q-table to perform only the known beneficial actions to obtain a high performance. To optimize the Q-learning algorithm to solve a specific problem, the behavior of (4.1) can be improved by adjustment of the learning rate (α) and the discount factor (γ). The importance of these values will be explained in the following sections.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \cdot (r_t + \gamma \cdot \max_A Q(S_{t+1}, A) - Q(S_t, A_t)) \tag{4.1}$$

Learning rate, α

The learning rate can be set to a value between zero to one and determines to what extent new information should override old information. After rewriting (4.1) the term (1 - α) $\cdot$ $Q(S_t, A_t)$ can be found, illustrating that if $\alpha = 1$ only the most recent information will be considered, and $\alpha = 0$ results in not absorbing any new information. An incorrectly chosen α can have bad consequences on the Q-table and result in that the Q-table never converges. If the Q-table never converge it will continue to try to learn without a stop and without reaching a final goal. To obtain the best possible result in this thesis different learning rates have been tried out during simulations, namely
$\alpha = [0.1; 0.2; \ldots; 0.9]$.

Discount Factor, γ

The discount factor determines the importance of a good or bad decision from S_{t+1} by adopting a value between zero to one. (4.1) states mathematically that based on the value of γ, $\max_A Q(S_{t+1}, A)$ will have various impact on the calculated Q-value. If $\gamma = 1$ the agent will strive to reach a high reward in a long-term perspective, while $\gamma = 0$ will make the agent extremely short-sighted and only consider current rewards. For beam tracking as in this thesis, it is the instantaneous current data throughput a beam provides that is the most relevant. Therefore, $\gamma = 0$ is chosen for all simulations, to completely neglect the fact that a future decision will affect the decisions regarding the current data throughput.

Epsilon, ϵ

Epsilon is variable created to balance the relationship between when to use exploitation and exploration. As described above, exploitation and exploration are two different ways to make decisions, hence to interact with the environment. Exploration is important to make sure that most states are visited at least once and to allow the Q-table to change if environment changes. ϵ is assumed to be a dynamical value between 1 to 0 and it is compared to random number. If the random number is smaller than ϵ will exploration be used. During training ϵ successively decreases with a specified decay rate, to make a fair transition from more exploration to more exploitation. In this study ϵ was initially set to one with a decay rate = 0.9998. ϵ got reduced by the decay factor according to (4.2), every time a positive reward was obtained. The decay rate of 0.9998 was chosen based on the simulation time. It was essential to not reach a too low epsilon too fast to do continuous exploration at the same time as only exploitation should be applied in the end of the simulation. Furthermore, ϵ was also the reference for which phase of the training the Q-algorithm was set to at the moment, but this will be further explained in chapter 5.

$$\epsilon = \epsilon \cdot decayrate \tag{4.2}$$

Reward, r_t

In this thesis the reward is calculated according to (4.3). A positive reward is therefore only obtained when one of the beams from the beam candidate set has better RSRP than the current serving beam.

$$r_t = \begin{cases} RSRP_{probingBeam} - RSRP_{currentBeam}, & \text{if } r_t > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

Q-table

The Q-table spans up the state-action space and is of the dimensions number of possible states × number of possible actions. Each time a new Q-value is calculated for a specific state-action pair, the new Q-value will replace the old Q-value in accordance with (4.1) in the Q-table at position $(S_t,\, A_t)$. The initial Q-table applied in this thesis is illustrated in Figure 4.2 and consists of 64 actions × 64 states:

- State: A state in this thesis is referred to a narrow beam.
- Action: An action in this thesis is referred to a switch between two narrow beams.

The number of states corresponds to the size of the applied beam grid, which consists of 64 beams in total. To be able to perform a beam switch to any beam, the number of actions needs to be the same as the number of states, i.e., 64 possible actions.

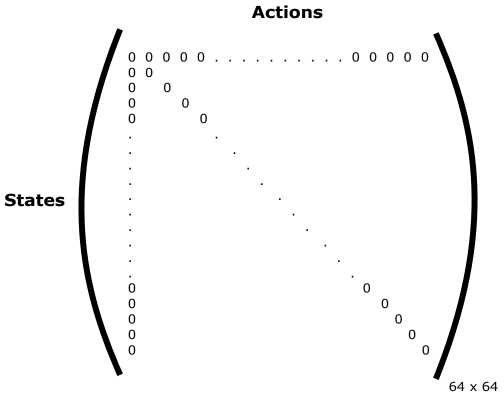

Figure 4.2: Initial Q-table with dimensions corresponding to beam grid.

Initially, before training has started, all elements in the Q-table are set to zero, since no actions from any state should be favored to be performed more often. In the Q-table in Figure 4.2 each row refer to one state and each column to one action. If the user is connected to a narrow beam S_t and the machine learning algorithm picks a candidate beams set consisting of the following six actions $[A1_t$, $A2_t$, $A3_t$, $A4_t$, $A5_t$, $A6_t]$ the Q-table will be updated at $(S_t,A1_t)$, $(S_t,A2_t)$, $(S_t,A3_t)$, $(S_t,A4_t)$, $(S_t,A5_t)$ and $(S_t, A6_t)$. The number of actions chosen are based on the number of beams being probed in baseline to be able to make a fair comparison between the machine learning algorithm and baseline.

Chapter 5
Method

This chapter describes the procedure of applying machine learning to beam management, more precisely beam tracking, in an Ericsson 5G NR simulator.

5.1 Data overview

The simulator offered much freedom regarding what logs that could be used to get access to different data. The following logs seemed interesting:

- Cell downlink (DL) throughput
- Bit Error Rate (BER)
- Current serving beam index + corresponding RSRP-value
- User position, direction and movement speed

To be able to investigate the result of the machine learning algorithm and also to compare the result to the baseline algorithm, a vital step was to find a log providing a relevant Key Performance Index (KPI). Cell downlink throughput was considered as the most promising KPI, mainly because of its relevance to this thesis, but also because of its easy accessibility in the simulator. BER could most likely also have been used, but was not further investigated in this thesis.

Even though cell DL throughput was a suitable parameter to evaluate the result on, it could not be applied to train the Q-table. When training the Q-table it was required to be able to evaluate the connection of each user at each beam. Hence, RSRP turned out as a suitable replacement, which was reachable for each user at each beam.

Logs regarding user position, direction and movement speed was also accessible in the simulator but theses were never included in the machine learning algorithm itself. However, those logs helped to increase the knowledge of movement patterns of UEs and to create our own movement patterns for UEs.

5.2 Method and algorithm overview

The goal of the thesis is to implement a machine learning algorithm, or more precisely a Q-learning algorithm in the Ericsson simulator to improve its ability to pick more suitable beams for a candidate beam set.

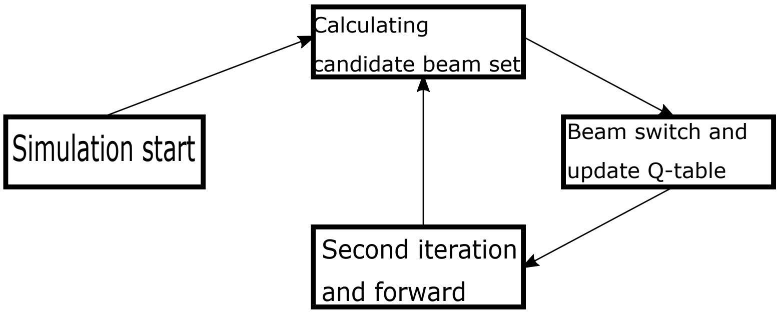

Figure 5.1: Block-scheme showing the structure of the implemented machine learning algorithm.

CSI-RS reports are sent on each beam in the candidate beam set to the UE and are used to make measurements and to calculate RSRP. The RSRP values are compared to decide which candidate beam that can provide highest channel quality. The Q-table will get updated when CSI-RS measurements have been sent back to the base station. An overview of the algorithm can be seen in Figure 5.1.

5.2.1 Different phases

Depending on the value of ϵ the algorithm will be in either training phase, transition between training and test phase or in test phase.

1. The training phase is mainly exploiting the baseline algorithm. Until epsilon ϵ is decreased to 0.25 the machine learning algorithm will only select one beam to the candidate beam set, while the baseline algorithm will select the rest of the beams. $\epsilon = 1.0$ to 0.25

2. During the transition a combination of baseline algorithm and machine learning algorithm will be used: $\epsilon = 0.25$ to 0.05

3. When ϵ has decreased to 0.05 the machine learning algorithm is said to be in the test phase. At this stage one beam will be chosen randomly from a closest beam set to the candidate beam set, while the rest of the beams will be decided based on exploiting the Q-table. $\epsilon = 0.05$ to 0.0

The different values of ϵ that are set as transition points were decided based on own experience after observing the result after running multiple simulations.

5.2.2 Simulation start

Initially, ϵ is set to one, which establishes the start of training phase. To maintain a decent cell DL throughput the baseline algorithm is responsible for most beam selections in this stage.

5.2.3 Calculating candidate beam set

To help the machine learning algorithm to only consider the most reasonable beams a closest beam set is created to select beams from. The closest beam set consists of the current serving beam plus the 19 closest beams. The closest beams are the beams that have the shortest distance to the serving beam where they hit the ground. The reason 19 closest beams were chosen was to decrease the required time of simulation to reach a good reliable Q-table. The candidate beam set will hold six beams in total and the current phase decides how these beams are selected:

- training phase: {four beams chosen by baseline + one beam chosen by machine learning + serving beam}
- transition phase: {two beams chosen by baseline + three beams chosen by machine learning + serving beam}
- test phase: {five beams chosen by machine learning + serving beam}

A beam chosen by machine learning can either be decided based on the Q-table or picked randomly from closest beam set. This decision is based on if a random variable becomes greater than or less than the current value of ϵ. Initially most beams chosen by machine learning will be picked out on random, but it strives to exploit Q-table more with time. However, one random beam from closest beam set will always be added to the candidate beam set to allow the Q-table in all phases to have the possibility to adapt to new behavior among UEs.

5.2.4 Beam switch and update Q-table

When the candidate beam set is full, CSI-RS reports will be sent on each beam and a RSRP value will be calculated for each beam. The RSRP value for each respective beam in the candidate beam set will be compared to the RSRP value of the serving beam to decide if a switch of serving beam would be beneficial or disadvantageous. A positive RSRP difference according to (4.3) will trigger a switch of serving beam and decrease ϵ according to (4.2). The Q-table will also get updated, in line with (4.1) at each beam/action in the candidate beam set. The state corresponds to the serving beam and each beam in the candidate beam set corresponds to an action.

5.2.5 Second iteration and forward

The different sections will be repeated multiple times each second according to the block-scheme in Figure 5.1. The value of ϵ will follow from each loop.

Chapter 6

Simulation Overview

In this chapter, an overview of the simulation environment will be presented. The available data, configuration parameters for the network and UE behavior is discussed and explained.

6.1 Simulation environment

In this thesis an Ericsson simulator has been used to simulate data traffic in a 5G system. The simulator is a powerful tool that supports multiple 5G features such as mmWave frequencies, beamforming and beam management procedures. It also made it possible to perform simulations over many simulation seeds and iterate a certain parameter of interest over many values. A simulation seed is one version of all the possible random parameter configurations that can take place in the simulation. Each parameter's behavior is determined by the seed, and for any seed the parameters are always determined in the same way so that the seed is reproducible. The learning rate variable in the Q-learning algorithm was iterated over values between 0.1 and 0.9 to identify which learning rate that gave best performance. Running the simulations over multiple seeds, instead of just one, added further credibility to the results since the proposed algorithm was tested in different surroundings that were influenced by the randomness of the seed. The simulations were run in parallel on powerful servers so that the simulation process would not take an unreasonably long time.

All the simulations were run twice with the exact same simulator parameters. The only thing that set them apart was that in one of the cases our proposed machine learning algorithm was used for beam tracking, and in the other case the already existing baseline algorithm was used.

The generated data was stored in log files that were post-processed in MATLAB. In the next section the simulator parameters are listed and explained in more detail together with a visualization of the simulation area.

6.2 Simulator parameters

- Simulation time: 400 s
- seeds: 50
- throughput log sample period: 0.01 s
- carrier frequency: 28 GHz
- deployment scenario: 1 cell with 1 base station
- cell radius: 100 m
- antenna:
  - height: 23 m
  - zenith angle: 23°
  - number of narrow beams: 64
  - number of antenna elements: 128
- UE:
  - always has data to transmit
  - number of UEs: 1 initial UE and then UEs arrive at an intensity of 20 per second until a maximum of 10 UEs are in the system.
  - height: 1.5 m
  - movement pattern:
    - straight mover:
      - speed: 4 m/s
      - spawn randomly in the cell
      - move straight in a random direction until a circle that circumscribes the hexagonal simulation area is reached
      - bounce on the circle border and continue the movement in the new direction
    - road mover:
      - speed: 10 m/s
      - spawn at coordinates within a predefined area
      - move horizontally (west-to-east mover) or move vertically (north-to-south mover) until a circle that circumscribes the hexagonal simulation area is reached
      - turn around 180 degrees and continue the movement in the opposite direction

Some of the parameters needs further explanation. The antenna zenith angle sets how much the antenna should tilt towards the ground, where an angle of zero degrees would mean that the antenna is directed horizontally. The number of antenna elements tells how many individual antennas that are used to beamform the signal.

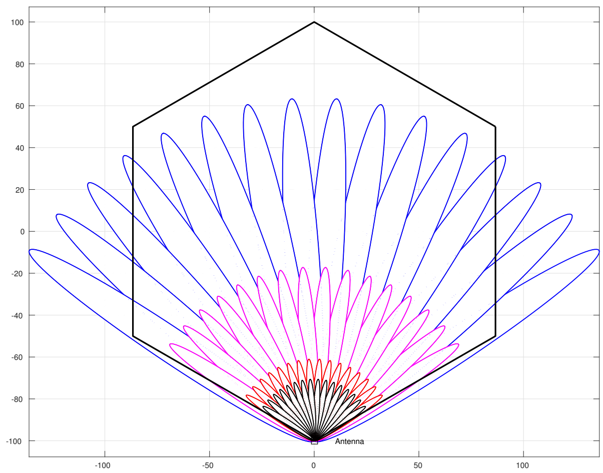

Figure 6.1: Illustration of the base station's position and conceptual beams drawn from it.

The simulation environment consists of UEs spawning in a hexagonal cell with a cell radius of 100 m, i.e. the distance from the center of the hexagon to its six corners is 100 m. Figure 6.1 illustrates the base station's position in the simulation area. It also shows the different beams that can be formed and where they hit the ground. It should be noted that the beams' width in the figure does not exactly correspond to the beams' width in the simulator. Nevertheless, the figure can still be useful to the reader to get a clearer picture of what the beam setup looks like. Especially useful is how the figure illustrates the four layers of beams that are formed and the fact that beams can overlap each other.

When a simulation starts, UEs that spawns begin to move according to their movement pattern. At the same time they try to establish a connection for data transmission with the base station. Once this initial access procedure is finished the UE is connected to one of the base station's beams. The beam tracking algorithm then tries to ensure that the moving UE always is connected to a beam with good channel quality. The simulations were run in two scenarios where the UEs movement patterns were different. With this setup, the proposed algorithm could be compared in the two scenarios to see if having UEs with a more predictable movement path had any impact on the algorithm's performance.

6.2.1 Straight movers only

In the first simulation scenario only straight movers exists. This type of UE spawns at a position in the cell that is determined by the randomness of the simulation seed. The UE continues to move until the simulation ends. Figure 6.2 shows two UEs starting position and movement in one of the seeds.

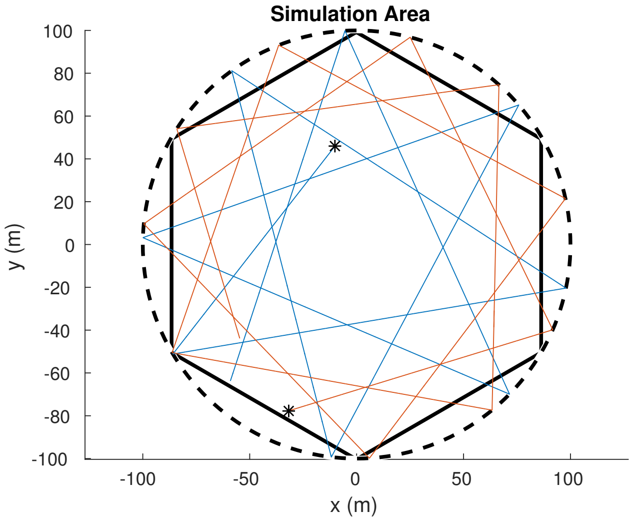

Figure 6.2: Movement pattern of UEs of the straight mover type.

6.2.2 Straight movers and road movers

In the second simulation scenario both straight movers and road movers exists. This scenario is supposed to represent a real-world scenario where UEs both move in random directions and along straight paths. The road movers' movement pattern can be compared to how vehicles move along a road, repetitive and predictable. Figure 6.3 shows starting position and movement of road movers moving from west to east and north to south, as well as a straight mover, in one of the seeds.

6.3 Simulation errors

A total of 50 seeds $\cdot$ 9 iterations $\cdot$ 2 scenarios $=$ 900 simulations were run when testing the machine learning algorithm. When testing the baseline algorithm 50 seeds $\cdot$ 2 scenarios $=$ 100 simulations were run (this test did not require any iterating parameter). For some unknown reason six out of the 1000 simulations were terminated before they could finish and no log files were stored from them. Five of the errors happened when the machine learning algorithm was tested and one error occurred when the baseline algorithm was tested. The interrupted simulations were distributed over four seeds, and out of fairness to the comparisons between the two beam tracking algorithms, and also between the two simulation scenarios, these four seeds were discarded when the results were processed.

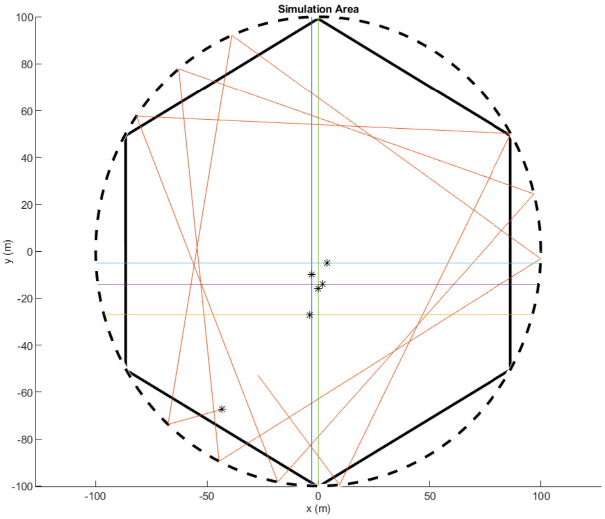

Figure 6.3: Movement pattern of UEs of the straight mover type.

6.4 Post-processing

The logged data was processed in MATLAB to be able to present the results in a clear way. Besides MATLAB's many useful tools two additional methods were used, namely cumulative moving average and linear interpolation.

6.4.1 Cumulative moving average

Cumulative moving average is a calculation method to analyze time series of data by averaging all of the previous data points up until the current data point. Viewed simplistically it can be seen as smoothing the data. This is especially useful when dealing with throughput data that can fluctuate a lot, even during short time periods. The equation for calculating the moving average is

$$CMA_n = \frac{x_1 + x_2 + ... + x_n}{n}. \tag{6.1}$$

6.4.2 Linear Interpolation

Due to the fact that UEs perform beam switches at different times (and therefore stores the RSRP values at different times) depending on which algorithm that is used, interpolation of the RSRP values is required to compare the two algorithms in a fair way. The interpolation of each UE's RSRP values for a seed was done by

$$y = y_0 \cdot (1 - \frac{x - x_0}{x_1 - x_0}) + y_1 \cdot (1 - \frac{x - x_0}{x_1 - x_0}). \tag{6.2}$$

This was done for all UEs in a seed, and by taking an average of all the UEs' interpolated RSRP values, an average RSRP over time for a seed was given. The same procedure is done for all the seeds and an average RSRP over time for all the seeds is then calculated. The same calculations are made both for Q-learning and for baseline. Finally, the two interpolations are compared and an RSRP difference at a specific time can be calculated.

Chapter 7

Results

This chapter presents the results from the simulations and shows the performance of the proposed algorithm compared to the baseline algorithm in the two simulation scenarios.

7.1 Optimal learning rate

The number of seeds where each learning rate value was the most optimal one for the straight- and road mover scenario is shown in Table 7.1. The most optimal learning rate for each seed was decided by comparing every learning rate's sum of the cell downlink throughput data for that seed.

Straight mover scenario		Road mover scenario	
Learning rate	Best choice	Learning rate	Best choice
0.1	1	0.1	4
0.2	6	0.2	6
0.3	7	0.3	8
0.4	8	0.4	5
0.5	4	0.5	4
0.6	5	0.6	7
0.7	4	0.7	5
0.8	5	0.8	5
0.9	6	0.9	2

Table 7.1: The number of times each learning rate was the most optimal for a seed in the two scenarios.

The optimal learning rate value varied for the different seeds and every value was the most optimal one for at least one seed. However, by looking at Table 7.1 it can be concluded that the values 0.4 and 0.3 were the optimal values most number of times for the straight mover scenario and the road mover scenario, respectively.

By varying the learning rate for each seed, an optimal performance can be reached for the Q-learning algorithm. However, knowing beforehand which learning rate to use for which simulation seed is not possible. Therefore, the performance for when a fixed learning rate value was used on all the seeds is also calculated. When the results are presented in the following sections, both the optimal performance and the performance for the fixed learning rate is compared to the baseline algorithm.

7.2 Straight mover scenario

The results for the straight mover scenario will be divided into two sub-chapters, namely throughput results and RSRP results.

7.2.1 Throughput results

Figure 7.1 shows the averaged cell downlink throughput for the Q-learning algorithm when using optimal learning rate, Q-learning with learning rate set to 0.4 and the baseline algorithm. As can be seen in the figure, the throughput of the Q-learning algorithm closely follows the baseline algorithm in the early stages of the simulation. This is expected, because the Q-learning algorithm uses the baseline algorithm to select candidate beams in the early training phase. However, an increase in throughput for the Q-learning algorithm compared to the baseline algorithm is noticed after a short while when candidate beams are selected based on the results of previous selected beams. After the initial training phase, the Q-learning algorithm's throughput stays superior during the rest of the simulation. The Q-learning algorithm even increases the throughput difference compared to the baseline algorithm the longer the simulation runs. This is the case for both the optimal and the fixed learning rate versions of the algorithm. As expected, the optimal learning rate's throughput is better than when a fixed learning rate is used.

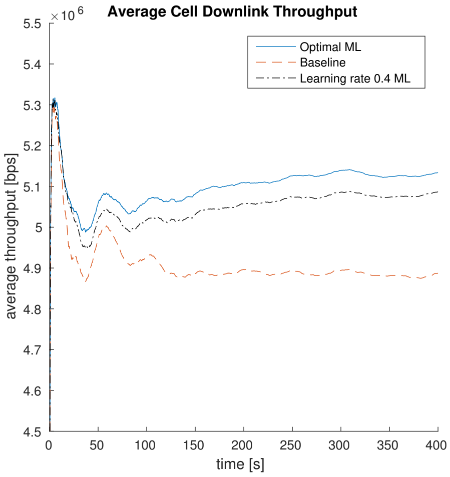

Figure 7.1: Averaged cell downlink throughput for the straight mover scenario.

The relative cell downlink throughput increase compared to the baseline algorithm can be seen in Figure 7.2. The two plots are calculated by dividing the throughput for the optimal and the fixed learning rate versions with the throughput for the baseline algorithm. The figure shows that at the end of the simulation, when the algorithm has had time to learn from previous decisions, a five percent increase in cell downlink throughput is achieved for the optimal Q-learning algorithm. For the algorithm with fixed learning rate the throughput increase is slightly lower, but still around four percent better than the baseline algorithm.

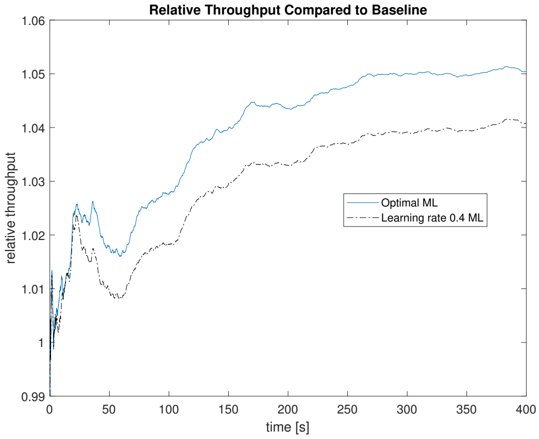

Figure 7.2: Relative cell downlink throughput for the straight mover scenario.

Another way to illustrate the throughput increase for the Q-learning algorithm is through a CDF plot, see Figure 7.3. The CDF plot shows that around 90% of the throughput values takes a value less than or equal 5.6 Mbps for the optimal Q-learning algorithm. For the fixed learning rate the 90% limit is just slightly lower, while the baseline algorithm's 90% limit is significantly worse at around 5.2 Mbps. Worth noting is that unlike figures 7.1 and 7.2, the CDF is not calculated by averaging the throughput data. Instead it uses the actual logged data values. The CDF plot confirms the Q-learning algorithm's throughput increase observed in figures 7.1 and 7.2.

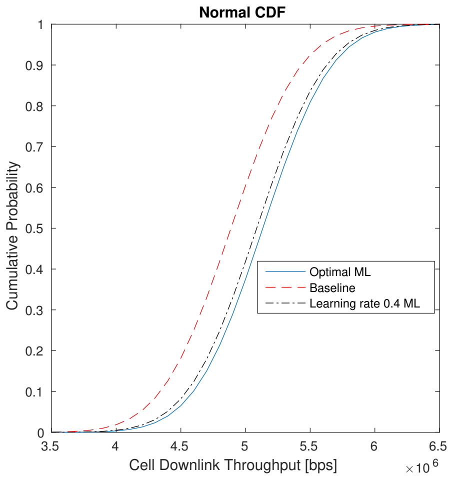

Figure 7.3: Cumulative distribution function of the cell downlink throughput for the straight mover scenario.

7.2.2 RSRP results

The improved performance of the Q-learning algorithm can also be illustrated by looking at the measured RSRP values that are reported by the UEs. In Figure 7.4, the percentage of all RSRP values that were greater than a specified threshold is shown. The threshold was set to -120 dBm. The bar diagram shows that around 81% of the RSRP values reached the threshold for the Q-learning algorithm. For the baseline algorithm only around 79% of them reached the threshold.

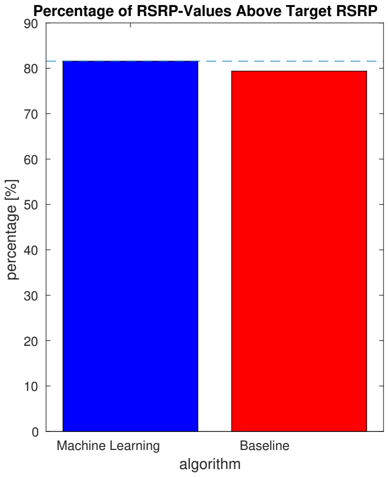

Figure 7.4: Percentage of RSRP values above -120 dBm in straight mover scenario.

Finally, an average interpolated RSRP difference is shown in Figure 7.5. The interpolation method that was used is described in Section 6.4.2. The sample points where the two algorithms are compared are every fifth second, which gives 80 RSRP difference bars. If the bars have a positive value, the RSRP difference is in favor of the Q-learning algorithm and if they have a negative value the RSRP difference is in favor of the baseline algorithm. The absolute value on the y-axis tells how much better one algorithm performed at a specific time compared to the other algorithm. The figure clearly shows that the average interpolated RSRP was higher for the Q-learning algorithm in a majority of the sample times. Furthermore, at the sample times where the average interpolated RSRP difference was in favor of the Q-learning algorithm a greater RSRP difference could be reached (around 2.5 dBm) compared to when the average interpolated RSRP difference was in favor of the baseline algorithm (around 0.7 dBm).

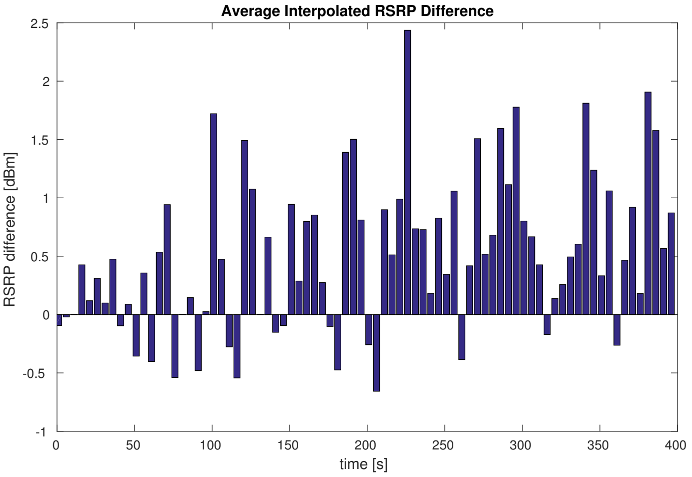

Figure 7.5: Difference between interpolated values in machine learning and baseline over all measured RSRP values in straight mover scenario.

7.3 Road mover scenario

In this section, the results for the road mover scenario are presented. Just like in the section for the straight mover scenario, the results will be divided into two sub-chapters called throughput results and RSRP results.

7.3.1 Throughput results

Figure 7.6 shows the averaged cell downlink throughput for the Q-learning algorithm when using optimal learning rate, Q-learning with learning rate set to 0.3 and the baseline algorithm. Similar to the straight mover scenario, the Q-learning algorithm's performance follows the baseline algorithm in beginning of the simulation. An increase in throughput for the Q-learning algorithm compared to the baseline algorithm is then noticed after a short while and the superior performance of the Q-learning algorithm stays consistent throughout the rest of the simulation. Furthermore, the throughput difference compared to the baseline algorithm is increased the longer the simulation runs. This is true for both when an optimal and a fixed learning rate is used.

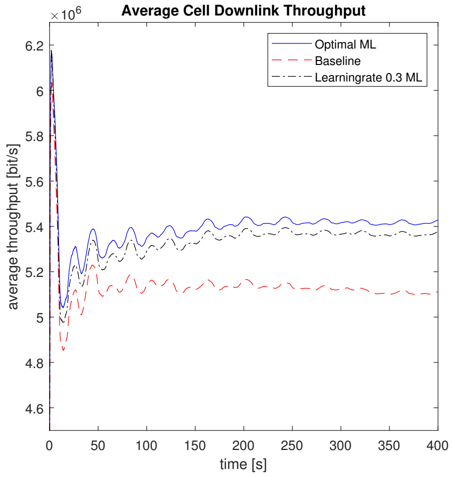

Figure 7.6: Averaged cell downlink throughput for the road mover scenario.

The relative cell downlink throughput increase compared to the baseline algorithm can be seen in Figure 7.7. This shows that a throughput increase over six percent is achieved at the end of the simulation for the optimal Q-learning algorithm. The increase for the Q-learning algorithm with fixed learning rate is over five percent at the end of the simulation.

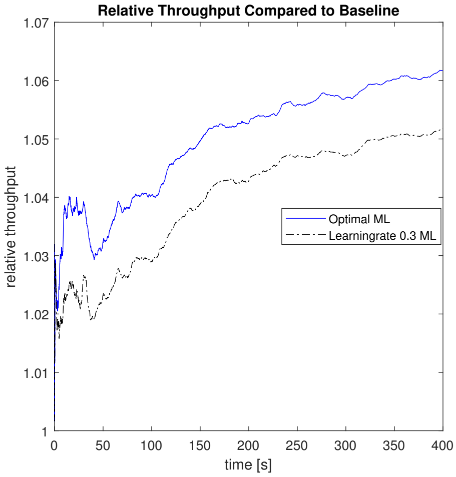

Figure 7.7: Relative cell downlink throughput for the road mover scenario.

Figure 7.8 shows the CDF plot of the cell downlink throughput. The figure shows that around 90% of the throughput values takes a value less than or equal 6.4 Mbps for the optimal Q-learning algorithm. The fixed learning rate follows closely behind the optimal version. Both perform better than the baseline algorithm whose 90% limit equals around 6.0 Mbps. Just like in the straight mover scenario, the Q-learning algorithm achieves a distinctive throughput increase compared with the baseline algorithm. The CDF plot confirms the increase that was observed in figures 7.6 and 7.7.

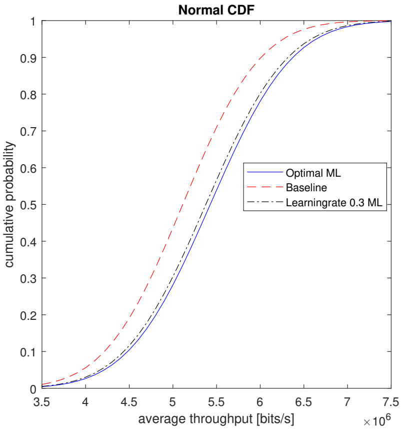

Figure 7.8: Cumulative distribution function of the cell downlink throughput for the road mover scenario.

7.3.2 RSRP results

In Figure 7.9 the percentage of all RSRP values that were greater than the threshold -120 dBm is shown. The bar diagram shows that around 84% of the RSRP values reached the threshold for the Q-learning algorithm. For the baseline algorithm only around 82% of the RSRP values reached it.

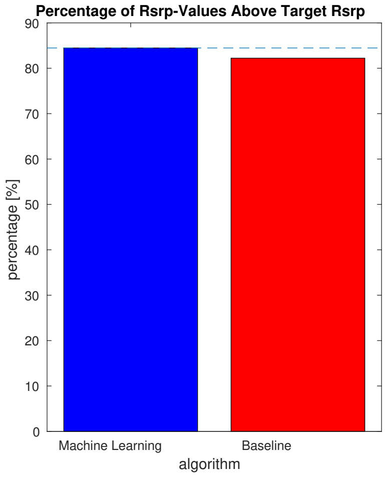

Figure 7.9: Percentage of RSRP values above -120 dBm in road mover scenario.

Finally, Figure 7.10 shows an average interpolated RSRP difference. The interpolation method that was used is described in Section 6.4.2. The sample points where the two algorithms are compared are every fifth second, which gives 80 RSRP difference bars. If the bars have a positive value, the RSRP difference is in favor of the Q-learning algorithm and if they have a negative value the RSRP difference is in favor of the baseline algorithm. The absolute value on the y-axis tells how much better one algorithm performed at a specific time compared with the other algorithm. Similarly to the straight mover scenario, the average interpolated RSRP was higher for the Q-learning algorithm in most of the sample times. Another result that matches the result obtained in the straight mover scenario is that a greater RSRP difference could be reached in favor of the Q-learning algorithm, around 2.5 dBm, compared with around 1.8 dBm for the baseline algorithm.

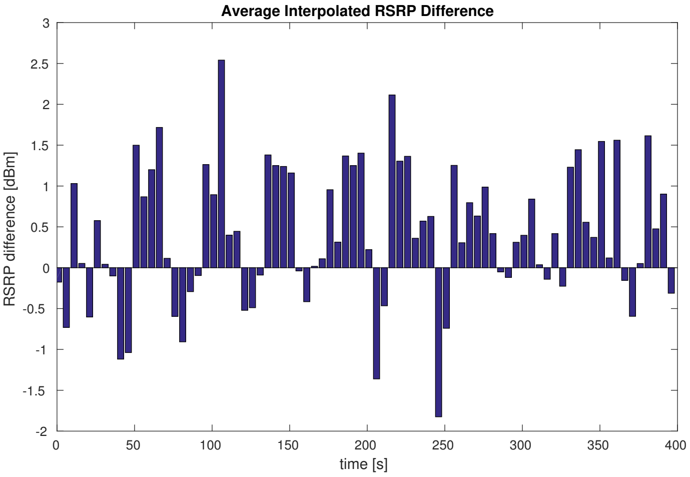

Figure 7.10: Difference between interpolated values in machine learning and baseline over all measured RSRP values in road mover scenario.

Chapter 8 Discussion

8.1 Overall Results

The overall results show that the machine learning algorithm is better than the baseline. The throughput diagrams in the results chapter illustrates a clear picture of higher throughput both when using the same learning rate for all seeds and when the optimal learning rate for each corresponding seed is being used. The RSRP related diagrams shows a clear picture of improved RSRP when using the machine learning algorithm.

The RSRP results can in this report be considered as a quality indicator, applied to verify the throughput results achieved. All RSRP related diagrams illustrates an average improvement of RSRP, which thereby strengthen the reasonability of improved throughput. Achieving higher RSRP value is a clear sign of improved beam selection. However, an additional interesting aspect to consider would be the number of switches that are performed during a simulation. Each beam switch is power consuming, (this is not discussed earlier) hence would a greater number of beam switches from one of the algorithms impair its overall result.

Another indicator that could have been investigated was CQI. Unfortunately, could not the machine learning algorithm reach CQI values for beam switch and it would have been extremely time consuming to allow it. RSRP is based on CQI, thus would CQI be a more sensible indicator to apply.

The simulated throughput values reliability is tough to verify. Cell throughput varies a lot dependent on the number of users is the system and the type of users in terms of traffic model and if a user desires constant data flow or not. Therefor is a comparison to reality not possible to make. But, on the other hand is RSRP fairer to make comparisons with. RSRP values higher than -100 dBm are considered good and will provide a steady signal without any noticeable disturbance. Unfortunately do the simulator seldom provide signals of this quality – neither baseline nor machine learning algorithm. After analysis of the simulation environment is our conclusion that none of the beams transmitted from a base station in the simulator will seldom be able to provide RSRP that meets real RSRP requirements. Because of this behaviour was it concluded required to decrease the target RSRP to match the outputs of the simulator. The target RSRP was decreased to -120 dBm, which is illustrated in Figure 7.4 and Figure 7.9.

8.2 Future work

For future works more aspects than beam selection could have been investigated to improve the results further. One aspect that was not covered in the thesis is how the size of the candidate beam set could have affected the result. If fewer beams in the candidate beam set could be used and still maintain an improved cell DL throughput compared to baseline it would be a sign of better performance. In addition to this, a deep neural network could have been applied instead of one layer reinforcement learning to possibly find more advanced patterns of which beam to connect to depending on previous actions. Another aspect is the relationship between the value of epsilon at a specific time and the phase of the machine learning algorithm. It was never deeper investigated how and when epsilon should be decreased and when the transition between different phases should occur. A last aspect that was not investigated was to train the Q-table based on user DL throughput instead of RSRP. Since the performance of the algorithm is determined with respect to throughput, it would be reasonable to try to do training based on the same KPI. This was not possible in the simulation environment that was used in this thesis, but could possibly be something to consider in future works related to this.

References

[1] 3rd Generation Partnership Project (3GPP), "Study on New Radio (NR) access technology (Release 15)", 3GPP TR 38.912 V15.0.0, Technical Specification, 2018, pp 20-23.

[2] V. Mnih, "Playing Atari with Deep Reinforcement Learning", arxiv.org, 2013. [Online]. Available: `https://arxiv.org/abs/1312.5602v1`. [Accessed: 2019-06-02].

[3] Y. LeCun, Y. Bengio, G. Hinton, "Deep Learning", Nature 521, 2015, pp 436.

[4] B. Ekman, "Machine Learning for Beam Based Mobility Optimization in NR", Dissertation, 2017.

[5] M. Bonneau, "Reinforcement Learning for 5G Handover", Dissertation, 2017.

[6] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang and R. W. Heath, "5G MIMO Data for Machine Learning: Application to Beam-Selection Using Deep Learning," 2018 Information Theory and Applications Workshop (ITA), San Diego, CA, 2018, pp. 1-9. doi: 10.1109/ITA.2018.8503086

[7] Ericsson. "World's first 5G NR radio.", URL `https://www.ericsson.com/en/networks/offerings/5g/5g-nr-radio`

[8] E. Dahlman, S. Parkvall, J. Sköld, "5G NR: The next generation Wireless Access Technology", Academic press, 2018.

[9] M. Giordani, M. Polese, A. Roy, D. Castor and M. Zorzi, "A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies", IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 173-196, Firstquarter 2019. doi: 10.1109/COMST.2018.2869411

[10] Ericsson. "Beamforming, from cell-centric to user-centric.", URL `https://www.ericsson.com/en/networks/trending/hot-topics/5g-radio-access/beamforming`

[11] W. Roh et al., “Millimeter-wave beamforming as an enabling technology for 5G cellular communications: theoretical feasibility and prototype results”, IEEE Comm. Magazine, vol. 52, no. 2, pp. 106–113, Feb. 2014.

[12] J. Palacios, D. De Donno and J. Widmer, "Tracking mm-Wave channel dynamics: Fast beam training strategies under mobility," IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, Atlanta, GA, 2017, pp. 1-9. doi: 10.1109/INFOCOM.2017.8056991

[13] F. Sohrabi and W. Yu, "Hybrid Digital and Analog Beamforming Design for Large-Scale Antenna Arrays," in IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 3, pp. 501-513, April 2016. doi: 10.1109/JSTSP.2016.2520912

[14] Ericsson. "Beamforming, from cell-centric to user-centric", URL https://www.ericsson.com/en/networks/trending/hot-topics/5g-radio-access/beamforming

[15] E. Dahlman, S. Parkvall, J. Sköld, "4G, LTE-Advanced and the road to 5G", third ed., Academic Press, 2016.

[16] Buon Kiong Lau, "ETTN15:3 Modern Wireless Systems - LTE and Beyond, Physical Transmission Resources", Department of Electrical and Information Technology, LTH, Lund University. Lecture notes 3, 2018. [Online]. Available: https://www.eit.lth.se/course/ettn15

[17] El –Samie, “Orthogonal Frequency Division Multiplexing”, Image Encryption 2013.

Printed by Tryckeriet i E-huset, Lund 2020

LUND
UNIVERSITY

Series of Master’s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2020-742
http://www.eit.lth.se