OULUN AMMATTIKORKEAKOULU

Daniel Leinonen

SEQUENCE CREATOR TOOL FOR 5G RADIO OVER THE AIR TESTING

A sequence creator software application for state-of-the-art radio test environment

SEQUENCE CREATOR TOOL FOR 5G RADIO OVER THE AIR TESTING

A sequence creator software application for state-of-the-art radio test environment

Daniel Leinonen
Sequence creator tool for 5G radio over the air testing
Spring 2024
Bachelor's Degree in Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Bachelor’s Degree in Information Technology

Author: Daniel Leinonen
Title of the thesis: Sequence creator tool for 5G radio over the air testing
Thesis examiners: Olli Himanka, Esa Leskelä, Keijo Hyttinen
Term and year of thesis completion: Spring 2024 Pages: 51

The subject of the thesis was the development of Sequence Creator Tool (SCT) which is an application that is used to create test sequences in a new state-of-the-art radio test environment at Nokia testing laboratory facilities. The aim was to develop a tool that enables fine granular control of antenna patterns created by the user, creation of test sequences for the new test environment, as well as management of user equipment (UE) attenuation and speed values. The Wall test environments are complex test environments that utilize a large array of antennas on a wall to simulate the movement of UEs, such as mobile phones. The work was commissioned by Nokia Oyj.

The implementation started with gathering of theoretical knowledge about the test environment for which the tool was developed. The initial design for the tool had been documented before the thesis was started, although it was still very much in its infancy. The new tool was intended to offer similar capabilities as the existing Wall Control Tool. With this tool however, it was time consuming to create Wall sequences especially when tens of UEs were used. The objective of the SCT was to simplify and accelerate the creation of custom test sequences for UEs and to automatically handle the conflicts and limitations of test environment control hardware.

Based on the gathered theoretical knowledge, a suitable implementation for the tool was decided and further improved during the development. As a result of the thesis the first version of the SCT was designed and developed for fifth generation (5G) radio testing. The project also emphasized the need for tools that can adapt to the fast evolution of 5G radio testing. Additionally, the project contributed to building personal academic understanding of the challenges and opportunities in the field of 5G radio testing.

Keywords: RF, Software Development, Python, Sequence Creator

OPENING SENTENCE

I would like to express my sincere gratitude to Esa Leskelä and Keijo Hyttinen, who have guided me throughout my bachelor's thesis and acted as liaisons and supervisors for Nokia Oyj. I would also like to express my heartfelt appreciation to my supervisor Tuomo Pylkäs to whom I am indebted for the thesis work position. Thank you for giving me the opportunity to develop my skills in the profession I wish to excel in. Special thanks to my thesis supervisor Olli Himanka for his invaluable guidance and assistance which he provided from start to finish during this thesis work.

Without your support, valuable insights, and constructive feedback, this thesis would not have been possible. Your guidance and mentorship have been instrumental to me throughout this academic journey, and I cannot thank you enough for your time, expertise, and support that made completing this thesis possible.

TABLE OF CONTENTS

CONCEPTS

3G	Third Generation of mobile network technology, which enables advanced mobile services such as internet browsing and video calls.
3GPP	Third Generation Partnership Project, its specifications develop and define protocols and network interfaces used in cellular telecommunications technologies.
4G	Fourth generation, significantly faster data transfer rates and enhanced support for multimedia applications.
5G	Fifth generation, ultra-fast data speeds, low latency, and increased connectivity, enabling the development of advanced technologies such as the Internet of Things and augmented reality.
API	Application Programming Interface, a structured set of rules that define methods and data formats which enables software applications to communicate and exchange data.
CLI	Command-line interface, an interface that allows users to interact with a computer program by entering lines of text, referred to as command-lines.
GUI	Graphical user interface, an interface that enables users to interact with a program through graphical symbols and visual markers.

LTE	LTE, industry jargon for a specific type of 4G used in mobile networks.
MIMO	Multiple Input Multiple Output, a 5G radio technology that uses multiple antennas to improve data throughput, reliability, and performance with multiple simultaneous data streams.
NSA	Non-standalone, used alongside the concept of 5G to define a 5G network architecture that utilizes the existing 4G network.
OTAVA	Over The Air Validation Area, a product development laboratory at Nokia Oulu.
RF	Radio frequency, a measurement representing the oscillation rate of electromagnetic radiation.
SA	Stand-alone, used alongside the concept of 5G to define a pure 5G network architecture that provides true 5G speeds with the 5G core network.
SCT	Sequence Creator Tool. The topic of the thesis; a new tool used for test sequence creation.
UE	User equipment, such as smartphones used in test environments to test radio capabilities and features.

1 INTRODUCTION

The subject of the thesis was the development of the Sequence Creator Tool for 5G radio over the air testing, which is an application that can be used to create fine-tuned sequences in a state-of-the-art radio test environment at Nokia. It is aimed to streamline the process of generating precise test sequences to aid engineers in their daily work. The work was commissioned by Nokia Oyj.

The testing environments at Nokia are used to test new radio software and hardware to evaluate their performance and to ensure there are no issues related to the new software and hardware before they are released to the customers. Evaluating the 5G radios requires testing how well they perform in actual field conditions. This assessment is essential to guarantee that 5G networks live up to their commitment of providing enhanced speeds and reliable connectivity. Valuable field tests can be very laborious and expensive to arrange thus the Wall testing environments were introduced to create test cases in a controlled environment. At the Nokia testing laboratory facilities, there are numerous different kinds of testing environments created for different purposes. One of these testing environments is a wall that is lined with antennas which are connected to UEs, such as mobile phones. These test environments, called Walls, allow the creation of test cases that simulate real-world scenarios to gather valuable information about the performance of the radios.

Test sequences are sets of predefined actions or scenarios designed to evaluate the performance of 5G radios systematically. These sequences simulate various real-world scenarios, such as signal strength variations, interference, handovers, and mobility, to assess how the radios handle different conditions. By subjecting 5G radios to these controlled test sequences, engineers can identify potential issues, optimize performance, and ensure that the radios function optimally in diverse circumstances, ultimately validating the reliability and efficiency of the software and hardware used in the radios. These kinds of sequences can be created with the tool specifically designed for the previously mentioned Wall testing environments.

The aim of the thesis was to develop a tool that enables fine granular control over antenna patterns created by the user, creation of sequences on antenna rows in the new Wall environment, as well as management of UE attenuation, step and speed values based on input data. An important feature that was taken into consideration when developing the tool was the rapid creation of tests. The tool was to be intuitive to operate and the output was to be quickly available, even for intricate test sequences. The tool in question is the Sequence Creator Tool; a new tool that can be used to create test sequences with simplicity in mind to aid testers who need to create sequences quickly.

This thesis will discuss wireless network technologies and the testing environments used at Nokia and their purpose in producing high-end products in a competitive global market for 5G radios. The tool that is currently used to create test sequences, its strengths, and topics of improvement will also be glanced at. This previously created tool resulted in the plan to create a new tool to be built alongside the existing one to support radio testing. The thesis will explain the reasons why the new tool was created, how this tool is used as well as the comprehensive long-term plans for the development of this tool to ultimately make it a powerful multipurpose tool.

2 TECHNOLOGY TESTED IN TESTING ENVIRONMENTS

Testing the 5G radio involves evaluating the performance of these new radios in real-world conditions. This testing is crucial to ensure that 5G networks deliver on their promise of faster speeds and more reliable connectivity. To carry out this kind of comprehensive testing Nokia has gone to great lengths to ensure high product standards.

2.1 4G and LTE

Fourth generation communication system (4G) is a wireless network technology system that allows communication between devices through radio waves. 4G is a highly advanced protocol that sends and receives data, succeeding third generation (3G) and the protocols used before that. It was developed to replace 3G as it offers a much faster and reliable connection. [1.] 4G is fully based on Internet Protocol, which has a standardized way of formatting data packets and performing actions to send the data across networks to the correct recipients [2].

Before 4G became a widely used system the ITU Radiocommunication Sector (ITU-R) issued new standards and requirements in International Mobile Telecommunications-Advanced Standard to upgrade mobile networks to reach speeds of at least 100 megabits per second along with other requirements [1,3]. With the available technology at that time this was not possible, hence 3G was improved and Long-Term Evolution (LTE) became a successor of 3G [2].

4G LTE is a specific type of 4G used in mobile networks hence the term LTE should not be used interchangeably with 4G. The main characteristic that differentiates 4G from 3G is the data rates achieved. [4.] LTE can be seen as improved and very fast 3G, a means to achieve true 4G speeds rather than being a standalone technology. This new technology greatly improved the user experience with better speed, coverage, and latency but still it did not fulfil the requirements set by ITU-R to be considered the next generation. As a new technology gets released it does not by default make it available to everyone. As an example, when 4G was released, the technology was not supported by majority of the devices at the time, which meant that 4G remained unusable to many people. On the other hand, infrastructure that provides LTE connectivity was and still is widespread, which enabled most people around the world to use LTE instead. [5,6.]

The same phenomenon can be seen happening with 5G. Although the new technological innovation exists, access to 5G remains scarce as most devices do not support 5G yet, hence LTE still plays a vital role in the industry. As LTE connectivity is almost universally available, it can be utilized during the migration time from 4G to 5G to provide network connectivity to older devices still dependable on older technology. [7.]

2.2 5G

5G can be deployed in two different ways: non-standalone (NSA) and standalone (SA). NSA 5G uses the existing 4G LTE network infrastructure to provide 5G services. This means that 5G devices can connect to both 4G LTE and 5G networks, and the 5G service will be delivered over the 4G LTE network when 5G is not available. NSA 5G uses the 4G core network while SA 5G uses 5G core. This network infrastructure is illustrated in figure 1. Albeit SA 5G can offer more reliable and faster connections, NSA 5G has some significant advantages, hence the two different deployment modes were defined in 3GPP specifications. [8.]

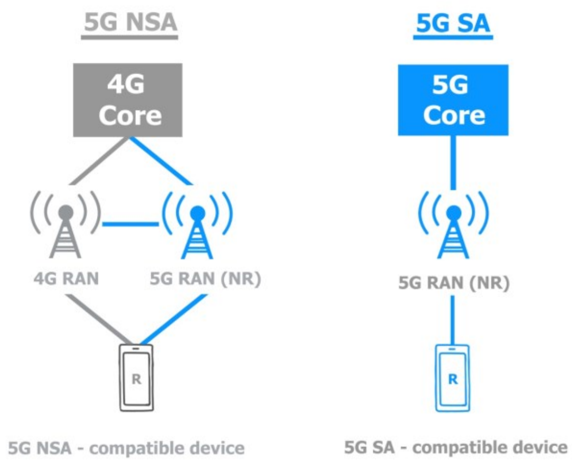

Figure 1: Simplified 5G NSA and 5G SA architecture

5G has been fused with LTE to create NSA 5G to aid with the scarcity of 5G infrastructure and user accessibility. NSA 5G can be quickly deployed by using the already existing LTE infrastructure, making the new technology available without having to replace the network as a whole. This naturally lowers the costs for service providers as a completely new architecture is not a necessity for NSA 5G. [9.]

NSA 5G acts as the first stage of 5G as it cannot offer all the same capabilities as SA 5G, such as low latency that is one of the biggest advantages of 5G. NSA also requires much more energy as it is dependent on 5G RAN as well as 4G LTE infrastructure. [10.] Gradual steps are taken to move to SA 5G, which will simplify the network and device architecture and introduce even more significant advancement to 5G network data speed, latency, and coverage [11].

SA 5G is a fully independent 5G network that does not rely on the legacy 4G LTE network. This means that in SA 5G networks 5G devices can only connect via 5G, and the 5G service will always be delivered over the 5G network. Advantages of SA 5G include higher data speed and extremely low latency. [12,13.]

The target 5G architecture option aims to enhance performance and simplification. SA 5G offers a simplified device architecture, increasing efficiency and making it more adaptable to meet the requirements of modern communication. [14.] SA 5G technology is designed to deliver ultra-low latency, reducing communication delay to near-instant levels which is crucial for applications like real-time gaming and autonomous vehicles. These features of 5G also allow a wider range of use cases, from IoT sensors to automatic vehicles. [15.]

SA 5G especially benefits applications that require high performance to support strenuous features. The other mode, NSA 5G is a good option when quicker deployment outweighs the benefits of having high speed and low latency when this kind of performance is not necessary. Both architectures have their benefits and drawbacks that are listed in detail in table 1 that goes over the characteristics of these technologies. For example, when building wireless network infrastructure, network providers must also take distances into careful consideration. As noted earlier, it is important to mention that 5G covers a shorter distance in comparison to both LTE and the commonly used 3G, especially in more remote rural locations. [16.]

Table 1: Characteristics of NSA and SA 5G technologies [17]

Feature	**Non-Standalone 5G (NSA)**	**Standalone 5G (SA)**
Network Architecture	Utilizes existing 4G infrastruc-ture	Requires a dedicated 5G core network
Core Network	Shares core network with 4G	Utilizes 5G core network
Data Speed	Lower speeds than full 5G	Very high true 5G speeds
Latency	Lower latency than 4G, but not optimal	Achieves true 5G ultra-low la-tency
Initial Implementation	Faster to deploy, leveraging 4G	Requires new 5G infrastruc-ture
Independence	Reliant on 4G for some func-tions	Fully independent 5G archi-tecture

2.3 MIMO

An essential technology used in 5G radios is Massive Multiple Input Multiple Output (MIMO) which is one of the fundamental pieces of technology used in 5G. This is a more advanced version to standard MIMO as it has the capability for multi-user support meaning it can concurrently support more UEs such as smartphones and other devices. [18.]

Massive MIMO is a technology used in modern wireless communication systems which utilizes a large array of antennas. Each antenna can transmit or receive signals simultaneously, enabling data streams to multiple users as well as better signal quality and reliability. This technology improves downlink and uplink network connections by utilizing spatial multiplexing, null forming, and beamforming. [19,20.] This technology is illustrated below in figure 2, where there is a comparison between legacy and MIMO antennas.

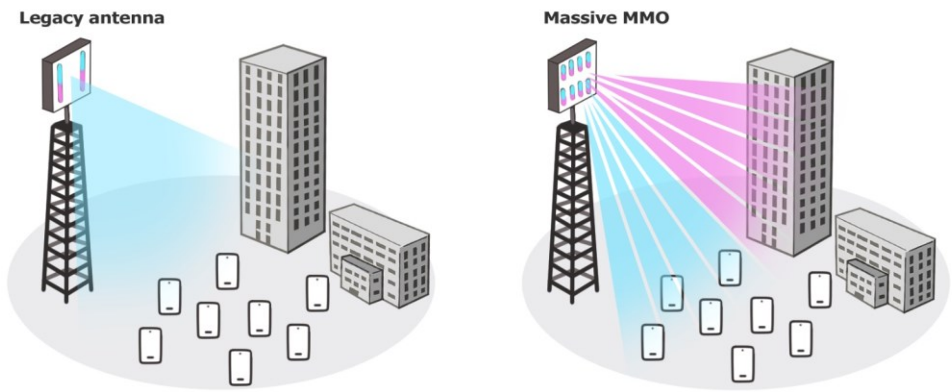

Figure 2: Legacy and Massive MIMO antenna comparison

Spatial multiplexing is a technique where multiple data streams are transmitted concurrently using multiple antennas, thus increasing the data throughput of the system. Spatial multiplexing takes advantage of the different propagation paths created by the antennas to transmit unique data streams, allowing for higher capacity making it a key component in advanced wireless technologies like 4G and 5G. [21.] The multiplexed data streams can go to one device or to different devices. Instances where the data streams are directed to one device are referred to as single-user MIMO (SU-MIMO), while instances involving multiple devices are called multi-user MIMO (MU-MIMO). Null forming is a variant of beamforming that aims to reduce or cut beam gain in certain directions where interfering transceivers are located, resulting in less interference in signal from multiple cell towers. [19.] Simply expressed, massive MIMO is a technology that uses lots of antennas to create a lightspeed highway for communication i.e., faster download speeds, video calls without any lag, and better connectivity overall [21].

2.4 Beamforming

Beamforming is a technology used in 5G radios to improve the quality and speed of wireless communication. Beamforming can be illustrated with a simple depiction. In a traditional audio system, the speakers would play the music equally in all directions, regardless of the position of listeners in the room. In a beamforming audio system, the speakers can direct the sound waves towards a precise position, creating a more focused and clear audio experience. By shaping and steering the beam to a single point, music can be heard more clearly and from a farther distance [22]. Beamforming in a 5G radio works in a similar way, except instead of sound waves, it uses radio waves to communicate. Multiple antennas work together to focus their signals on a specific direction where the receiver is located, minimizing the radio waves that are sent in other directions thus resulting in a stronger and more efficient signal sent directly to the intended recipient. [23.]

Previously, the transmission and reception of signals was mainly accomplished with the utilization of dish antennas, which are equipped with the ability to mechanically steer the signals to the desired direction. These antennas however have major drawbacks, such as being slow to steer, considerable physical size and the limitation of having only one preferred radiation pattern or data stream. [24.] A newer solution to combat these drawbacks introduced electrically steered phased array antennas.

Phased array antennas consist of antenna elements that merge their radiation patterns to shape a main lobe directing energy to the desired location. Concurrently, the antenna strategically counteracts signals in undesired directions, creating nulls and side lobes. The array optimizes main lobe energy while minimizing side lobe radiation and enables the manipulation of radiation direction via phase adjustments to each antenna element. This technology offers several advantages over the traditional antennas, such as enhanced long-term reliability, fast steering, and the capability to support multiple beams. [25,26.] Figure 3 illustrates the effects of a phased array antenna system that can manipulate the radiation from each individual antenna to electrically create an offset to direct the signal to the desired direction.

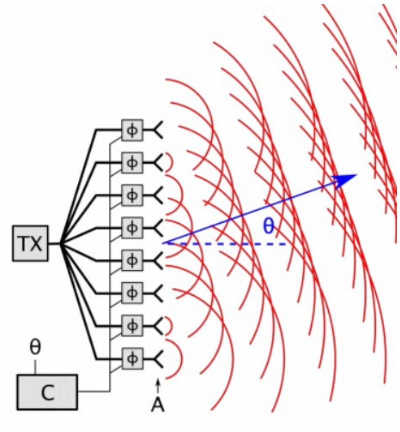

Figure 3: Phased array antenna system [27]

By using beamforming, 5G antennas can communicate more effectively with devices that are farther away, moving at high speeds, or located in crowded areas with lots of other devices. Overall, 5G beamforming technology helps to improve the performance, coverage, and capacity of wireless communication networks, making it possible to support more devices and applications. [23.]

3 TESTING ENVIRONMENTS AT NOKIA OTAVA PRODUCT DEVELOPMENT LABORATORY

Testing environments at Nokia OTAVA (Over The Air Validation Area) are used to evaluate the performance of the LTE and 5G radios in terms of speed, latency, and overall reliability. This includes measuring network throughput, response time, and error rates. Coverage and capacity are thoroughly tested to ensure that the radios can provide service to the entire area that they are designed to serve and to make sure that radios can handle a high volume of traffic without slowing down or dropping connections. Low latency and interference are significant features that 5G promises to deliver hence these are pivotal subject when testing.

In the highly competitive global market, customers require more than just great performance from radios. Electricity is a significant expense item for operators and with the new generation of radios, the electricity consumption of the radios has been decreased. Low power consumption of the radios is a key element in being energy-efficient which presents in a wide range of benefits, such as reduction in site energy consumption which the customers naturally desire. Overall, 5G radio testing is an essential part of the development and deployment of 5G networks, and it ensures that customers can enjoy the full benefits of this new technology. It is important to stress the need for comprehensive testing to ensure that 5G networks are reliable, secure, and perform as expected under a variety of conditions.

Testing environments used at Nokia include, but are not limited to, radio frequency (RF) shielded Chambers, RF measurement Walls and the new spearhead test environment project for which the tool was created. These testing environments are used for comprehensive radio performance evaluation and feature testing before the latest software and hardware are shipped to the customers.

3.1 RF shielded chambers

RF shielded chambers are enclosures made of conductive material designed block electromagnetic interference and radio RF signals from entering or leaving the chamber [28]. This is achieved with conductive materials, such as copper, steel, and aluminium to form a Faraday cage, an electrically conductive shield that prevents electromagnetic fields from penetrating the enclosure [3]. RF shielded test environments at Nokia are lined on all external surfaces with a metallic layer to shield against RF interference. Additionally in the chambers there is absorbent material that absorbs and distributes electromagnetic energy so the chamber will not internally reflect transmitted RF signals to interfere with the accuracy of the tests [29].

Containers are padded with insulating material on the inside to prevent any signals from escaping outside of the environment as well as to prevent signals from entering from outside of the environment which could interfere with the test results. The inner walls of a chamber are padded with radiation absorbent material, that prevents the signals sent from the radio from reflecting to the sensors which would produce false results. [30.]

RF shielded chambers at Nokia are compact testing environments that are generally used for testing specific components or modules of the new software releases for different radios. They are ideal for catching issues related to individual features. Due to the compact size of the chambers, they are more affordable than the larger testing environments and they can be closely packed, thus saving a lot of space [31]. These chambers are especially suitable for testing simple features on radios when a small number of antennas in the test environment is sufficient for the tests run on the radio. These environments are also a great fit for radio antenna testing, as these products are not as complex as radios or basebands. Figure 4 provides an example of how these chambers are often set up.

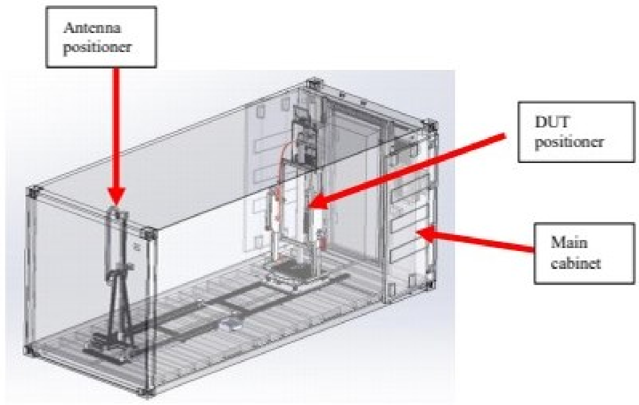

Figure 4: RF shielded chamber

Benefits of using a compact testing environment include faster feedback loops, since the test suite is smaller, and lower resource requirements for acquiring more of these testing environments as fewer components are needed. However, these testing environments may not provide a complete picture of the overall functionality of the software or hardware and some defects, either in software or hardware may escape all the way to the end-user if comprehensive testing is not carried out. The physical size of RF chambers may restrict the size of the equipment or devices that can be tested inside. This limitation could be problematic for testing large-scale 5G equipment.

In summary, RF chambers offer controlled and shielded environments that yield reproducible 5G testing results. However, they come with the trade-offs of limited space, and reduced capabilities compared to larger testing environments, making them suitable for only a limited set of testing scenarios. For other use cases, alternative testing environments may be more practical and effective.

3.2 RF measurement Walls

RF measurement Walls, such as Chambers, are enclosed testing environments used in 5G radio testing. These testing environments have an adjustable radio mount and a wall lined with antennas which are connected to UEs. This architecture is illustrated in a simplified way in figure 5. The rows of antennas on these Wall environments make these testing environments highly controllable and allow engineers to create test cases that simulate real-world scenarios to ensure more accurate results which help gather valuable information about the performance of the radios. This capability makes the Wall a more comprehensive testing environment in comparison with the RF shielded chamber. Figure 6 shows how The Walls are designed to allow the simulation of smartphone movement in different environments. This is greatly beneficial for beamforming verification which is a crucial phase before new software and hardware are deployed to the customers. They can be utilized to produce a more detailed and accurate analysis of the overall performance of the software and hardware. These testing environments provide the ability to identify issues that may only occur when multiple connections are used together, and the ability to test for edge cases that may not be caught in smaller testing environments.

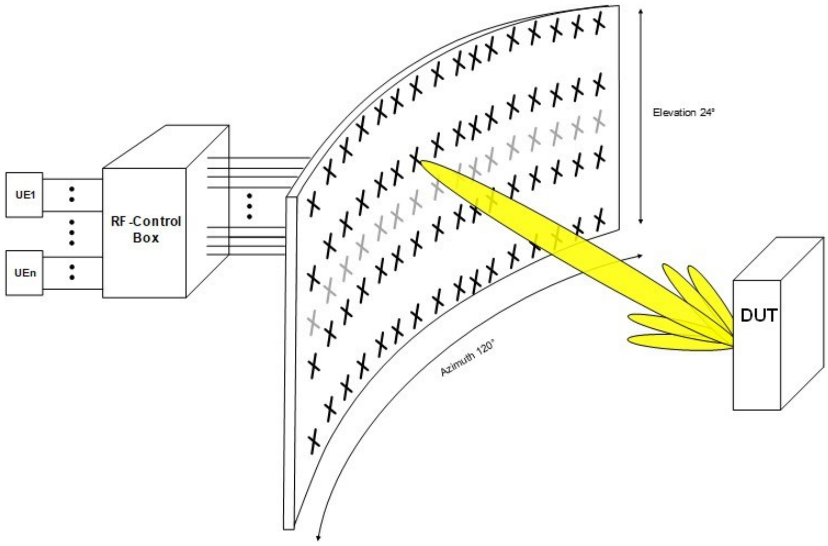

Figure 5: Illustration of a simplified Wall testing environment layout

Walls are used for MIMO testing which is a critical technology in 5G networks. Walls can also be combined with channel emulators to recreate realistic scenarios. Channel emulators are devices that simulate the real-world RF environment in which a variety of 5G devices such as smartphones and base stations will operate. They are used to test the performance of 5G devices in a variety of conditions, including different signal strengths, different levels of interference, and different types of fading. They can also be used to test different types of 5G networks, such as SA and NSA networks.

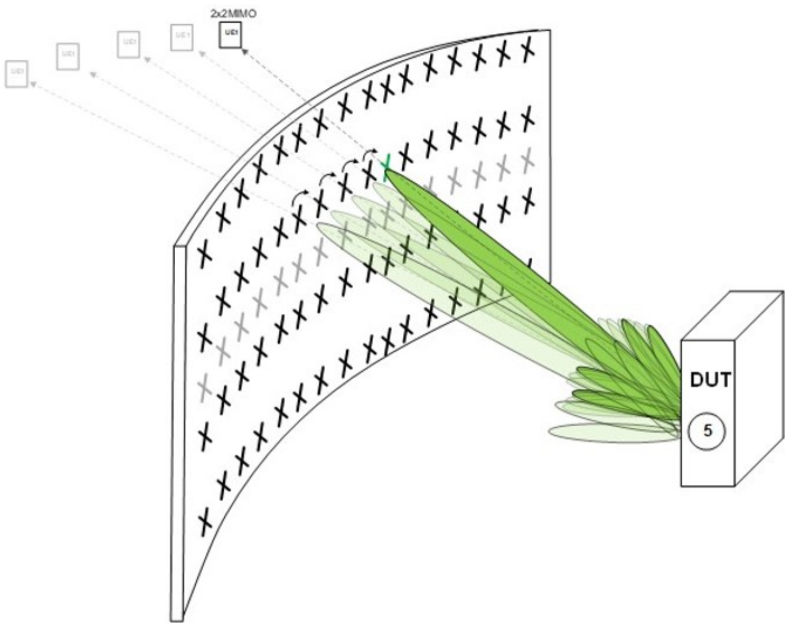

Figure 6: UE movement simulated in a Wall environment

Channel emulators can simulate attenuation, shadowing, fast fading, doppler effect and noise. Attenuation is the loss of signal strength as it travels through the air whereas shadowing is variation in signal strength caused by objects that block the signal [32]. Fast fading and doppler effect both are caused by the movement of the user or the base station. In fast fading there is variation in signal strength and in doppler effect there is shift in frequency of the signal. Noise is interference from other signals in the environment. The use of channel emulators in 5G testing is essential to ensure that the devices will perform as expected in the real world [33]. Channel emulators can be used together with the Wall capabilities to simulate UE attenuation, as shown in figure 7. This combination creates even more precise simulation of real-life scenarios. This helps researchers and engineers assess how 5G devices and networks would perform in different environments and helps to ensure a positive customer experience and to avoid costly recalls. This capability within the Wall testing environments makes them superior to the RF chambers.

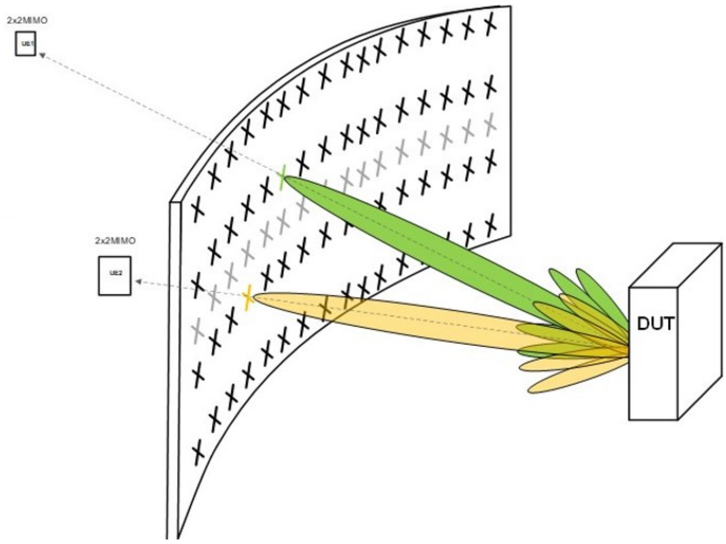

Figure 7: Illustration of the effects of antenna attenuation

Multipath propagation is a phenomenon in which a radio signal is reflected, scattered, or diffracted by objects in the environment, resulting in multiple copies of the transmitter signal arriving to the receiver. This can cause effects such as signal interference and fading. [34.] The effects of multipath propagation can be simulated with different propagation models, such as the Saleh-Valenzuela model, the Clarke's model, or the COST 231 Hata model to name a few. MIMO systems inherently experience multipath propagation without needing reflective surfaces. By using multiple antennas at both the transmitter and receiver, the signals can take different paths and experience various phases and amplitudes, leading to multipath effects. [35.]

By simulating the effects of multipath propagation, engineers can predict and study the signal reflections and scattering that occur in real-world environments and assess the performance of 5G systems to identify potential problems encountered in urban, suburban, or rural environments [36]. This method is demonstrated in figure 8. This can help to ensure that 5G systems meet the performance requirements for different applications.

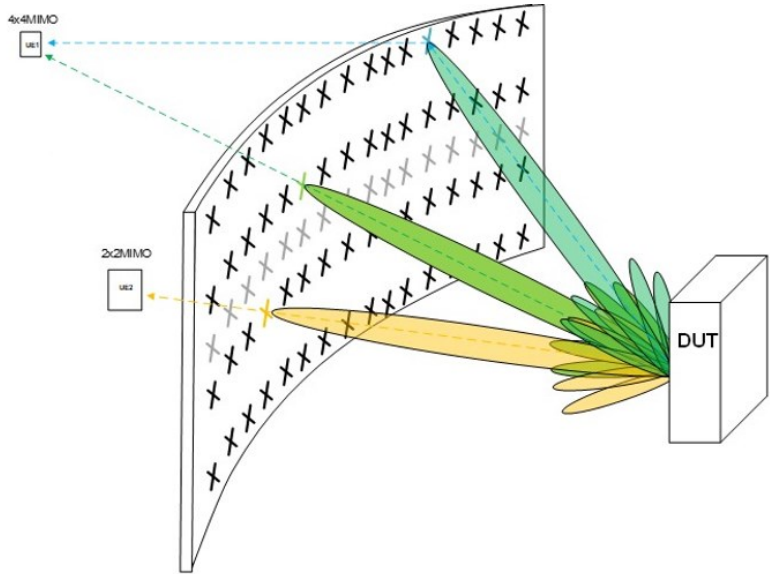

Figure 8: Illustration of the usage of multipath propagation in testing

Due to the complexity of these testing environments operating and setting up RF measurement walls may require specialized knowledge and expertise, which can be a temporary barrier for beginner testers. Building and maintaining Walls is also much more expensive due to the specialized materials and equipment required for the additional features compared to the previously mentioned RF chambers. Despite this, these RF measurement walls are immensely valuable tools for engineers as well as the whole industry to thoroughly study the rapidly evolving technology of wireless networks.

3.3 New Wall environment

Nokia has an ongoing project underway to build a new state-of-the-art testing environment setup that incorporates the latest and most advanced testing technologies available. Setup of the antennas, radio and the overall architecture of the hardware and software significantly improve the efficiency and accuracy of testing. Benefits of the new state-of-the-art testing environment include considerably faster feedback loops compared to the Walls or even the chambers, improved test coverage, and the ability to identify issues that may otherwise be missed in other testing environments. The complexity of this testing environment requires the incorporation of new tools to aid with creating extensive tests, hence the plan for the new Sequence Creator Tool was introduced. The first version of the tool is focused on the new Wall environment but there are plans to implement support for other testing environments as well in the future.

The new Wall environment enables testing that is difficult or even impossible, expensive, unreliable, not easily reproducible or time consuming in field testing. This is achieved with the ability to divide the test environment into sectors, which enables the isolation of different simultaneous test cases. Each sector can be configured to match real-world conditions or specific scenarios in urban, suburban, and rural environments. By using multiple sectors, the behaviour of a real 5G network with multiple base stations and devices can be emulated. This enhances the realism of testing and provides insights into how the technology will perform in a multi-cell environment.

This isolation prevents interference between different tests, ensuring accurate results and reducing the potential for false positives or negatives. With the ability to run different tests simultaneously in different sectors, efficiency of testing is significantly increased. This is particularly valuable in the fast-paced development of 5G technology, where multiple features or configurations need to be validated concurrently. Parallel testing allows optimal utilization of the resources of the testing environment. Instead of running one test at a time, testers can use the available sectors to execute multiple tests, reducing downtime and maximizing the return on investment for the testing infrastructure. By conducting multiple tests simultaneously, the validation process can be accelerated, and the time required to identify and rectify issues is reduced. These also speed up the overall development cycle and enables quicker deployment of stable 5G products to the market.

Different sectors can be configured to simulate various testing scenarios. This allows comprehensive testing of the performance of 5G radios across a wide range of real-world situations, ensuring robustness and reliability. A sector-divided test environment offers flexibility to customize test parameters for each sector. This customization ensures that each test case accurately reflects the conditions under which the 5G radio will operate, leading to more accurate and actionable results.

4 WALL CONTROL TOOL FOR SEQUENCE CREATION

The creation of sequences for test cases is not a new concept. A tool called Wall Control Tool was created to generate customisable test sequences which are a highly beneficial asset that can be used in radio performance testing. The previously existing tool works in a similar fashion as the new SCT, hence the old tool acted as a framework for the thesis as it was created for the same purpose.

4.1 Purpose of the Sequence Creator Tool

Instead of manually selecting and running individual test cases, testers use this tool to create and then run a series of tests as a sequence, saving time and effort while ensuring consistent and repeatable results. By creating specific sequences, they ensure that the same tests can be reproduced with updated products, making it easier to compare results and identify potential issues.

Test sequences allow testers to streamline the testing process and they are crucial in the testing and validation process. Test sequences refer to predefined sets of test cases or scenarios that are designed to evaluate the performance and reliability of 5G networks and devices. When talking about the tool used at Nokia for 5G radio testing, test sequences refer to predefined scenarios or sequences that consist of a series of specific actions and parameters that simulate the real-world to cover a range of different scenarios to test that the radios meet their requirements.

The Wall Control Tool is used for creating custom sequences with step-by-step instructions. The sequences created with this tool are highly customisable, with control over UEs used, the antennas they connect to, attenuation for each UE on each step and many more settings. These custom sequences allow the creation of tests that help testers to find shortcomings in radio software or even hardware before these kinds of issues make their way to the customer which would result in additional costs and dissatisfied customers.

4.2 Operation of the tool

The tool has a visual user interface for sequence creation which allows the user to choose from available resources e.g., mobile phones connected to the test environments. The availability of these resources depends on other services which the tool relies on. The status and details of the connected resources are always up to date as the data is fetched from a separate source dedicated to handle this subdivision. The user can select multiple UEs to a single sequence and the amount is only limited by the available UEs and physical restrictions of the test environment which is used in the sequence. After the desired UE for the sequence creation is selected, the user selects which antenna it connects to in the first state of the sequence. In each state each UE may connect to one or more antennas depending on how many antennas the UE itself has connected to the test environment.

Next the user sets variables that alter the connection between the radio and the UE. Setting parameters such as attenuation and delay can be used to simulate speed, distance or obstacles encountered in a real environment. Each UE may have a specific delay and attenuation value respectively. After the parameters are set to desirable values for each antenna used in the sequence, the created sequence step is saved and added to the sequence as the first state. After the first step has been created, the user can proceed to the next state and repeat all the steps to create as many states as necessary to create a desired sequence. The available settings for UEs can be seen in figure 9 below.

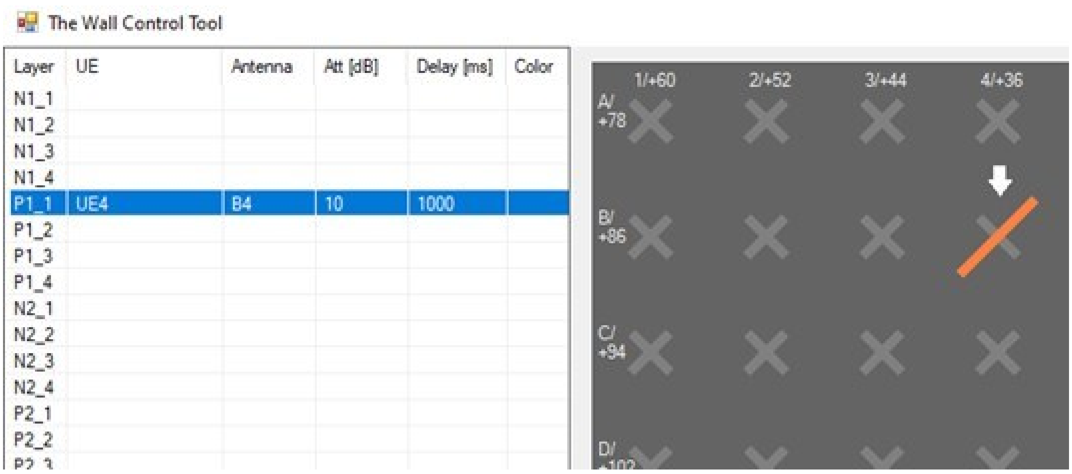

Figure 9: Sequence creation in The Wall Control Tool

When the sequence is finished, the output of the sequence is a custom file that is used to run the created sequence in the test environment. The file includes all the necessary parameters to select the correct UEs, the antennas they connect to and the instructions for the simulated movement of the UEs within the test environment. These sequence files can be saved and used indefinitely, or they can be used as a foundation for new and improved test sequences.

4.3 Topics of improvement for the existing tool

This tool should not be replaced but rather there should be an additional option for a simplified and streamlined way to create sequences. The other option offered should be intuitive for the user and be scalable for multiple UEs. The Wall Control Tool can be used to create sequences by fine granular control over antenna lines and steps. This, however, is a very time-consuming way to create sequences especially for multiple UEs. With customisable antenna patterns, UE coordinates, speed and attenuation all set individually for each UE in each step of a sequence, these sequences quickly become quite tedious to create. Improving the existing tool for sequence creation involves refining its capabilities, efficiency, and user experience. There was room for improvement, hence a plan was set to build a tool that can be used to generate complicated test sequences faster and with more ease.

The areas of improvement include introducing simple user control to support complex custom user-defined sequence patterns and optimizing the algorithm to generate patterns more quickly, especially for longer sequences or complex patterns. The new tool should automatically handle the limitations of Wall control hardware and attempt to resolve conflicts when creating sequences. All this should be implemented into the tool while keeping the focus in keeping the tool simple. There was also a decision made to initially make the tool available as a desktop application as a command-line interface (CLI) and later extend it to be available on web browsers as a React based tool. Both would serve an important role, and both should be designed to be intuitive and user-friendly interfaces that allow users to easily input their requirements and understand the generated antenna patterns and test sequences. This should be achieved by including tooltips, explanations, and examples to guide users in using the tool effectively.

5 SEQUENCE CREATOR TOOL

The problem to be resolved in all its simplicity was making the creation of test sequences easier and faster. Plan for the tool was set in motion alongside the new and improved test environment being built. This environment has a larger set of antennas in use compared to the older wall test environments, which makes creating sequences with the older tool even more laborious. The initial concept for the new SCT was solely focused on the new Wall environment, which was to be expanded later to the older environments. The SCT enables effective use of Wall environments.

5.1 The benefits of the new tool

Costs of the tool can be measured through required workforce as it takes time for engineers to develop the tool and create additional functionalities. There would possibly be a need for a dedicated engineer to work on the tool for a long period of time. The benefits and resulting savings outweigh the costs by a long mile as the SCT enables effective use of the new Wall test environment. The threshold to create custom sequences is lowered for testers which results in more tests and thus faster detection of issues in radio software and hardware. The new tool also acts as a quality-of-life improvement in creating test sequences.

The new tool allows time saving with considerably faster creation of test sequences. When the first version of the new tool was compared to the existing tool when creating sequences, creating sequences with the new tool saved a considerable amount of time. Creating a simple single UE sequence using the new tool took less than 30 seconds, whereas with the old tool it took around 5 minutes. When the tool is further refined and expanded to support multi-UE sequence creation, the time saved would further improve. Creating single UE sequences would possibly be 5-10 times faster and more than 20 times faster when creating multi-UE sequences. Less time spent creating test sequences allows for more time to be spent on other tasks and productive work.

Earlier test cases have been less complex which has resulted in not being able to take advantage of expensive Wall investments in full effect. Creating complex multi-UE test case scenarios has generally been possible through field testing, instead of easily generating test sequences for these purposes. Test cases in a controlled environment are easily reproducible from UE scenarios point of view as the speed, attenuation and other variables are controllable whereas in field testing tests, this can be challenging. These tests are also significantly faster to repeat.

5.2 Planning the concept of Sequence Creator Tool

Planning out the concept of the SCT was an important part of the thesis, as there was no clear plan for it, other than a few requirements set for its functionalities. Proof of concept was decided to be made as a simple React application to visualize what the product would possibly look like, and how it would be used. After the proof of concept received a green light from principal technical leader Keijo Hyttinen, further discussion resulted in redirecting the first version of the tool to be a CLI application.

The type of data handled and the data output as well as the capabilities of the tool were determined as the minimum viable product requirements were set. It was decided that the focus for the tool would be the ability to create UE sequences for the new Wall environment fast and easily. It was decided that the tool must include functionalities such as antenna pattern creation, attenuation, and speed settings as well as simple conflict resolution capabilities. Other requirements set for the tool were sector selection to support improved capabilities of the new Wall environment and intuitive syntax to control pattern movement in the sequence.

All the functionalities for the tool would be first and foremost applied in a simplified way. The data model for the sequences should also be scalable for multiple UEs, when this capability would be added later in development of the tool. With these requirements predefined the development of the SCT began by first building a proof of concept for the tool.

5.3 Proof of concept for the new tool

The proof of concept for the new tool was created with a web browser graphical user interface (GUI) implementation. The user interface was created with React and its purpose was to show the idea of creating an antenna pattern easily by user clicking and choosing antennas from a grid and then giving input with simple syntax to give instructions how they want the pattern to move on the new Wall. This would result in a sequence that would be displayed on the grid by moving the pattern. The appearance of the proof of concept for the tool can be seen in figure 10.

The aim of the proof of concept was to show that optimizing the creation of sequences to generate complex test quickly would result in saving a vast amount of time in testing. The concept was also used in a presentation where the benefits and costs of the tool were discussed, and it was demonstrated to be a valuable asset. The proof of concept turned out to be valid, but it was decided that the tool would be first developed to be run as a CLI tool that would be later extended with a web UI to improve further usability. The concept was used as a framework to aid with the development of the tool, as a visual representation of the desired product for a CLI tool is greatly beneficial.

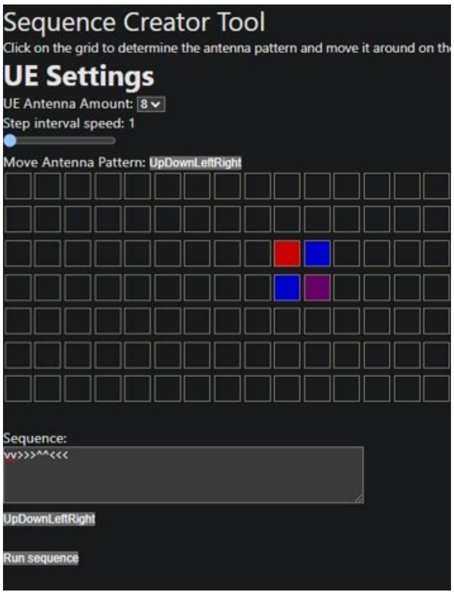

Figure 10: Sequence Creator Tool proof of concept

5.4 Sequence creation

The sequence creation with the tool works with text-based syntax to generate complex UE antenna movements in the new Wall test environment to allow testers to quickly make sequences with one or more UEs in testing. The tool consists of a few principal settings that the user must define to generate a sequence for UE movement. The sequence will be generated based on these settings. To create a sequence the user must select which sectors are used and which sector is active when creating an antenna pattern. To create an antenna pattern, the user selects the starting position for the pattern and then uses the predefined syntax to create a desired antenna pattern used in the test sequence. The user needs to give instructions for the movement of the UE and define the speed and attenuation settings. When these settings are defined, a sequence can be created.

5.4.1 Configuration selection

The users can choose a wall configuration file from the tool. These configuration files allow the creation of sequences in different test environments available, which can significantly improve the user experience as well as streamline creating sequences for all test environments. These files contain essential information about the test environments, such as the antenna grid parameters i.e., amount of antenna rows and columns, the distance between antennas as well as additional antenna grid extensions in use. This feature makes the tool more versatile and adaptable for future additions to the configurations.

Configuration files promote standardization of the creation of these models across the team. As this feature is already implemented to the tool, users can simply create a new configuration file to add a new test environment to be available in the tool, without changing any of the existing code. By avoiding the need for code changes, the existing configurations remain unaltered, eliminating the potential errors introduced to the code during updates, making it a highly effective and scalable solution.

5.4.2 Sector selection

The new Wall environment consists of multiple sectors. This is a new feature that enables multiple test engineers to simultaneously run tests without interfering an ongoing test started by other engineers. When creating test sequences, the sectors can be used either individually or as groups, further expanding testing capabilities. If only one sector for the sequence is chosen, the pattern can be moved within the borders of the chosen sector. The pattern retains its form throughout the sequence as it moves along the antenna grid, even when "colliding" with the borders i.e., any of the connected antennas is at the last antenna position on the active sector. When the sequence is created for multiple sectors, the pattern movement is enabled for all the chosen sectors, similarly to how it works with a single sector. When reaching the last antenna of a sector, the antennas in the pattern simply switch to the next available sector. The sectors used in the sequence are selected before creating the antenna pattern.

5.4.3 Antenna patterns

A core part of developing the SCT was to plan how to create antenna patterns. Each UE has multiple antennas which are connected to the test environments, and they can switch to any antenna in the environment. To simulate movement of one or multiple UE, antenna patterns can be created, which mean there is a predefined pattern on the antenna grid that can be moved around on the antenna grid on the wall. Antenna pattern files are created before sequence files and then used as a foundation for the sequence file created to run the finished test. They require a couple of parameters that make up the necessary details to create the pattern. All antennas in the test environments have two polarities, positive and negative polarity. Antennas that include both polarities are called cross-polarity antennas. With the tool, it can be decided which polarities are used in the antenna pattern. Forward slashes represent positive polarity antennas, back slashes negative polarity antennas and the letter X represents cross-polarity antennas.

To create an antenna pattern that is made up of two positive polarity antennas, two negative polarity antennas and two cross-polarity antennas all next to each other on the same row, this can be achieved by using the corresponding syntax previously explained: //\\XX
To create an antenna pattern that is made up of four cross-polarity antennas in a square pattern, it can be done with any of the following syntax: XXDXLX, XDXXUX or XXDLXX. This simple syntax can be used to create various antenna patterns. The available syntax and respective functions of each character are listed below in table 2.

Table 2: Syntax and their functions in antenna pattern creation

Syntax	Function
/	Positive polarity antenna
\	Negative polarity antenna
X	Cross-polarity antenna

U	Up one antenna row
D	Down one antenna row
L	Left one antenna column
R	Right one antenna column

Combining these characters results in a set of instructions for the tool to create an antenna pattern file that contains the right number of antennas in their desired positions on the wall grid respectively. This pattern is then used in combination with the movement instructions as well as speed and attenuation settings given by the user to create the sequence file. Before creating the sequence file there is an automatic conflict detection and resolution run on each created antenna pattern to prevent unwanted behavior that would be physically impossible to run in the actual test environment.

5.4.4 Antenna pattern visualization

The integration of antenna pattern visualization within the CLI version of the tool provides users with a clear and intuitive way to analyse the patterns created. By having a visual aid at disposal, the users can quickly see any deviations in the pattern, enabling them to edit their designs with precision. Instead of relying exclusively on the syntax the user prompts the tool with, which can be challenging for non-experts to understand, a visual representation offers a universal language that is easily understood by everyone.

The visual representation shows the created pattern at its starting coordinate on the Wall environment set by the user. It offers a clear view of how the pattern turned out, allowing users to do adjustments as needed. The visual representation of the created antenna is shown on a corresponding grid that aligns with the selected configuration of the test environment that was used for the sequence created. Figure 11 includes an excerpt from code to implement this feature.

```python
def show_antenna_pattern(self, x_value, y_value, pattern):
    cross_polarity_antenna = "╳"
    positive_polarity_antenna = "╱"
    negative_polarity_antenna = "╲"

    min_x = self.get_min_x_value()
    min_y = self.get_min_y_value()
    max_x = self.get_max_x_value()
    max_y = self.get_max_y_value()

    mutable_grid, offset_value = self.get_grid(self.config_file_name)

    x_position = x_value
    y_position = y_value

    for ind, char in enumerate(pattern):
        if char == 'U':
            y_position -= 1
        if char == 'D':
            y_position += 1
        if char == 'L':
            x_position -= 1
        if char == 'R':
            x_position += 1


        if y_position < min_y:
            y_position = min_y
        elif y_position > max_y:
            y_position = max_y

        if x_position < min_x:
            x_position = min_x
        elif x_position > max_x:
            x_position = max_x

        antenna_coordinate = offset_value + y_position * offset_value + x_position * 2 + 2


        if char == 'X':
            mutable_grid[antenna_coordinate] = cross_polarity_antenna
        elif char == '/':
            mutable_grid[antenna_coordinate] = positive_polarity_antenna
        elif char == '\\':
            mutable_grid[antenna_coordinate] = negative_polarity_antenna

        if char in {'X', '/', '\\'} and (ind + 1 < len(pattern)) and pattern[ind+1] in {'X', '/', '\\'}:
            x_position += 1

    grid = ''.join(mutable_grid)
```

Figure 11: Excerpt from the code for antenna pattern visualization

5.4.5 Conflict detection and resolution

An important feature that lies under the hood of the tool is the conflict detection and resolution functionality. It was determined in the backlog for the tool that the tool should be able to handle common conflicts when creating sequences, so that even less technical testers could use the tool and easily create sequences without having to resolve conflicts arising due to the physical limitations of the new Wall environment that they may not be familiar with. The antennas in the testing environment have specific switches that allow the coupling of the antennas in the test environment with the UEs. These switches have restrictions that limit the way the patterns can be created for the test sequences.

Conflicts arise when depending on the chosen antenna polarities and positions of the antennas in the pattern. The tool assigns each antenna their respective groups and sub-groups, so that conflicts are resolved. An example of a situation where a conflict would arise in the pattern is when there are multiple antennas of the same polarity and group on the same row in different antenna coordinates. Multiple UEs are not able to connect to different antenna positions on the same row of antennas simultaneously if the same group of switches is used for different UEs. These limitations are considered in the tool and the common conflicts are detected, which can be seen from the excerpt from the code in figure 12. There are a set of ways the most common conflicts are resolved automatically, and the user is notified of the changes made to the pattern file to prevent from conflicts.

```
def resolve_pattern_conflicts(self, file_name):
    dir_path = get_seq_files()
    file_path = os.path.join(dir_path, file_name)
    try:
        with open(file_path, "r+") as f:
            data = f.readlines()
            f.seek(0)
            for line in data:
                json_line = json.loads(line)
                layers = json_line["layers"]

                for i, layer1 in enumerate(layers):
                    for j, layer2 in enumerate(layers):
                        if (
                            i != j
                            # layer1["name"][:2] equals to "P1" or "N1" for example
                            and layer1["name"][:2] == layer2["name"][:2]
                            and layer1["y"] == layer2["y"]
                            and layer1["x"] != layer2["x"]
                        ):
                            # Change group when same polarity and row
                            group = int(layer2["name"][1:2])
                            group += 1
                            subgroup = 1
                            if group <= 5:
                                layer2["name"] = layer2["name"][:1] + str(group) + "_" + str(subgroup)
                            else:
                                del layers[j]  # Delete layer from pattern

                        elif (
                            (i != j
                            and layer1["name"] == layer2["name"]
                            and layer1["y"] != layer2["y"]) or
                            (i != j
                            and layer1["name"] == layer2["name"]
                            and layer1["x"] == layer2["x"]
                            and layer1["y"] == layer2["y"])
                        ):
                            # Change subgroup when same polarity but different row
                            subgroup = int(layer2["name"][3:])
                            subgroup += 1
                            if subgroup <= self.get_max_y_value():
                                layer2["name"] = layer2["name"][:3] + str(subgroup)
                            else:
                                del layers[j]  # Delete layer from pattern
```

Figure 12: Excerpt from the code for antenna pattern conflict detection and resolution

5.4.6 Sequence syntax

The sequence creation was developed to use simple syntax to control the UE movement. The syntax allows the movement of the antenna pattern in single steps in all the four directions on the wall i.e., right, left, up and down. The user may also decide to move the antenna pattern to any of the directions continuously until the last antenna position available is reached. The options for antenna pattern movement are listed below in table 3.

Table 3: Sequence control syntax

Syntax for antenna pattern movement	**Movement achieved with syntax**
>	Right one step
<	Left one step
^	Up one step
v	Down one step
->	Right until at the rightmost antenna column
-<	Left until at the rightmost antenna column
-^	Up until at the highest antenna row
-v	Down until at the lowest antenna row

These instructions can be chained to create a sequence that moves step by step on the antenna grid. An example of a pattern made of two cross-polarity antennas placed next to each other in the pattern having a starting position on sector 1, antenna row 3 and antenna column 10 and the movement syntax used being “>>>vvv-^-<” would set the end position for the first cross-polarity antenna at sector 1, antenna row 0, antenna column 0 and end position for the second cross-polarity antenna at sector 1, antenna row 0, antenna column 1. Figure 13 shows the iteration over the movement instructions given by the user which then modifies the position of each antenna in the test sequence file.

```
def modify_state(self, char, next_char, skip_char, sequence_file_name, previous_state):
    permit_append = True
    direction = None

    if char == ">":
        direction = 'right'
    elif char == "<":
        direction = 'left'
    elif char == "v":
        direction = 'down'
    elif char == "^":
        direction = 'up'

    if direction:
        original_values = copy.deepcopy(previous_state['layers'])  # Make a copy of the original values
        any_false = False  # Flag variable to track if any iteration returned False
        for item in previous_state['layers']:
            if direction in ['right', 'left']:
                item['x'], item['wallSector'], permit_append = self.move(item, permit_append, direction)
            elif direction in ['up', 'down']:
                item['y'], permit_append = self.move(item, permit_append, direction)
            if not permit_append:
                any_false = True
                break

        if any_false:
            # Revert values to original state
            previous_state['layers'] = original_values
        else:
            self.append_new_state_to_sequence(permit_append, sequence_file_name, previous_state)
```

Figure 13: Excerpt from the code for antenna pattern movement

5.4.7 Speed and attenuation settings

With the SCT, the user can define the speed used for the UE during the test sequence. There are three different modes the user can use for the UE speed: fixed speed, time-based speed and zig-zag speed mode. In the fixed speed mode, the user can set a speed value which remains unchanged for the UE throughout the sequence. In the time-based speed mode, the user can set start and end speed values and the value will either increase or decrease depending on the set values. The user must also set step values so that each state during the sequence increases or decreases the speed value as much as desired. In the zig-zag speed mode, user sets start, end and step speed values like the time-based speed mode. In addition, the user sets the desired number of loops that are performed for the speed values during the test sequence. These speed settings allow vast amounts of different test scenarios to be created, that simulate the movement of UE in the test environments.

Attenuation used for the UE in the test sequence is a parameter which the user can define. Like the speed settings, there are three different modes available for the user to choose from: fixed attenuation, time-based attenuation and zig-zag attenuation mode. Fixed attenuation mode allows the user to set a static value for the attenuation of the selected antenna. Time-based attenuation requires the user to set the start, end, and step attenuation values to create an increasing or decreasing pattern for the attenuation value for the selected antenna. Zig-zag attenuation mode requires the additional parameter for the loops performed during the test sequence.

5.5 API

The development of an application programming interface (API) was also started to later extend the Sequence Creator Tool to be available as a web browser tool. There is support for all the necessary functions to have the same capabilities as the CLI version of the tool currently has. The backend was built with support for both CLI and API in mind, hence both routes work by calling mostly the same functions. The API was built using FastAPI so that it aligned with the goals of the test automation team to slowly unify the technology that is developed down the line. FastAPI is a modern web framework tool used in building APIs with Python. This framework allowed very fast development cycles and made testing and maintain the API simpler and easier. An excerpt from the code of the tools API can be seen in figure 14.

Creating a backend for the tool that can be used with both a CLI version and an API offers multiple advantages. Firstly, the coexistence of these alternatives offers flexibility in usage of the tool as different users have different preferences. Users can choose their preferred version of the tool as some may prefer the convenience of a CLI to quickly create simple tasks, while others might want to benefit from the API through the graphical user interface that the web browser tool would later offer. Secondly, by providing an API, users can automate tasks programmatically, making it easier to integrate the use of the tool into workers existing automation workflows. Thirdly, building an API for the tool is beneficial as it can then be more easily integrated to the existing software ecosystem.

```python
# POST new pattern
@app.post("/pattern", tags=["Post"])
async def create_new_pattern_file(pattern: SequenceState):
    result = seq_service.create_new_pattern_file(pattern)
    return result

# POST set directions
@app.post("/sequence/directions", tags=["Post"])
async def set_directions(directions: str):
    result = seq_service.validate_directions(directions)
    return result

# POST set starting coordinates
@app.post("/sequence/starting_coordinates", tags=["Post"])
async def set_starting_coordinates_API(x_value: int, y_value: int):
    result = seq_service.set_starting_coordinates_API(x_value, y_value)
    return result

# POST new speed settings
@app.post("/sequence/speed", tags=["Post"])
async def set_delay_settings(start_delay: int, end_delay: Optional[int] = None, step_delay: Optional[int] = No
    result = seq_service.set_delay_API(start_delay, end_delay, step_delay, zig_zag_step)
    return result
```

Figure 14: Basic API operations of the tool

This integration can lead to developing the tool to be even more powerful as it can use other services available as extensions to provide more functionalities. Testing the tool also becomes more straightforward as automated tests can be written to verify the functionality of the API. A well-designed API can serve as a stable foundation for the backend as it allows for scalable and complex interactions with the tool.

5.6 Unit tests

Unit tests have an essential part in software development as they help in detecting bugs early, improve code quality and maintainability as well as support collaboration with people working on the same code [37]. Unit tests are extremely beneficial for further development of the tool when other people start working on the tool in the future. They serve as documentation as they provide examples of how functions of the tool are intended to work, making it easier for new developers to understand the code. When new developers follow the framework laid with the unit tests, they can confidently make changes without worrying about breaking existing functionality. [38.]

5.6.1 Implementation

While implementing unit tests required an initial investment of time and effort to build during the development of the Sequence Creator Tool, the long-term benefits in quality and maintainability of the code saved time many times over. Development of the tool incorporated lots of unit tests. Early in the development process unit tests were implemented to discover bugs and inconsistencies in the performance of the tool, making refining and refactoring the code easier, thus accelerating the development cycle. Writing unit tests required to think critically about the design and functionality of the code. The tests also forced to break the code down into smaller units with clear purposes and functionalities, leading to cleaner, better-quality code.

By writing tests for specific components of the code during the development of the tool, they helped to catch and fix issues as soon as they occurred, reducing the likelihood of bugs going unnoticed in the later stages of development. These bugs would have otherwise created bottlenecks later in the development process which naturally saved a lot of time. If a function or a class was modified in a way that accidentally introduced a new bug, existing unit tests caught most of them, helping in maintaining code stability over time. This way unit tests made it safer to refactor the code as they provided confidence that new bugs have not emerged while refactoring the code. They also made it easier to understand and use the code correctly when returning to a functionality created very early in the development process. Writing unit tests also encouraged to consider different error scenarios and handle them properly.

5.6.2 Input validation and algorithmic tests

Different types of unit tests were implemented to the tool. This was done to ensure that changes made to the code during development of the tool would not result in unexpected behaviour. Input validation tests were added to verify that the tool can handle different data types correctly. The unit tests verified that the tool can handle data types, such as text, numbers and special characters without producing errors or unexpected behavior. This ensured that these kinds of unexpected instances would be handled gracefully.

Algorithmic unit tests were responsible for generating test sequences that simulated various scenarios while ensuring that all the specified criteria were followed to create valid sequences. These tests also assessed the accuracy and efficiency of the core algorithms for test sequence generation as well as the scalability of the algorithms, anticipating future requirements as the software would be developed further for increased data and complexity. These tests also utilized deliberately faulty or unexpected inputs to ensure that the tool can generate valid output.

5.6.3 Integration and edge case tests

Integration tests validated the interactions between different components within the tool. They ensured that data is passed correctly between components and reduced the risk of errors. Not only did the integration tests verify that the data flow works seamlessly, but they also scrutinized the entire software, revealing poor design in the code architecture. These kinds of tests enhanced the performance of the tool and refined the overall structure of the codebase.

Edge case tests assessed how the tool behaves when subjected to boundary conditions. These unit tests were instrumental in ensuring the overall robustness of the tool in real-world usage as the tool should efficiently handle scenarios such as minimal or maximal input values. Edge case tests were used to identify and address any bottlenecks or performance issues that may arise during sequence generation. An example of this can be seen in figure 15 which shows an excerpt from a conflict resolution unit test. These tests ensured that necessary edge cases were handled accordingly instead of causing the tool to crash or handle the input incorrectly.

```python
    @mock.patch("backend.src.sequence_service.sequence_service.get_seq_files")
    def test_resolve_pattern_conflicts(self, mock_get_files):
        with TemporaryDirectory() as temp_dir:
            mock_get_files.return_value = temp_dir
            response_post_pattern = client.post(f"/pattern", json=TEST_PATTERN2)
            self.assertEqual(response_post_pattern.status_code, 200)
            assert "new_file" in response_post_pattern.json()
            # check that a new file was created with the expected name
            file_name = response_post_pattern.json()["new_file"].split("/")[-1]
            assert file_name.startswith("pattern_") and file_name.endswith(".json")
            assert os.path.join(temp_dir, file_name)

            response = client.put(f"/pattern/conflicts/{file_name}")
            self.assertEqual(response.status_code, 200)

            with open(os.path.join(temp_dir, file_name), "r") as f:
                file_contents = f.read().splitlines()
                data = [json.loads(row) for row in file_contents]
                self.assertEqual(data[0], RESOLVED_TEST_PATTERN2)
```

Figure 15: Excerpt from the code for pattern conflict resolution unit test

By implementing a comprehensive array of unit tests, the codebase could be decisively refactored during the development of the tool while maintaining confidence in the reliability of each unit of code. This enabled easier maintenance, early bug detection, and it provides a solid foundation for future enhancements. These tests functioned as documentation, keeping track of expected behavior of individual units. This documentation proved invaluable in catching potential problems before they escalated into more complex issues.

6 RESULTS

As a result of the thesis work on building the Sequence Creator Tool for 5G radio testing, great results were achieved. The tool successfully exhibited its efficiency in generating test sequences tailored for the existing 5G radio environments and for the new Wall environment currently under construction. Through testing and analysis, it was shown that the tool not only streamlined the process of sequence creation but also made a notable improvement in the speed of sequence creation. The results indicate that with further development and added functionalities the tool has potential to significantly contribute to the advancement of 5G testing methodologies.

Through experimentation, it was observed that the tool could be used to create diverse test sequences, covering a wide range of scenarios. Functionalities for diverse antenna pattern creation and conflict detection were successfully developed. The automation of certain parts of sequence creation streamlined the process workflow, which minimizes manual labour and accelerates the overall testing cycle. The tool has the possibility to contribute to significant cost savings in terms of reduction in time saving. By automating the sequence creation process, the tool also reduced the risk of human error and inconsistencies in test sequences, leading to more robust and reproducible results.

The theoretical cost savings that the first version of the tool could provide were calculated to be around 40,000 euros annually. This does not include the benefits that further development of the tool would bring nor does it consider the primary benefits of improved testing capabilities that the tool provides. In conclusion, the tool proved to have the potential to bring monetary value as well as provide substantial practical value by streamlining the complex sequence creation process thus aiding the engineers with practical advantages in their daily tasks.

7 FUTURE WORK

The Sequence Creator Tool is used in a very complex test environment which made developing this tool fascinating. As a result of this thesis the tool that was created was seen as a great benefit to sequence creation. Throughout the development of the tool new possible implementations were planned. The new Wall environment at the Nokia test laboratory facilities builds a new foundation for 5G radio testing offering a plethora of opportunities to further improve the tool in the future. Some functionalities have already been discussed and defined as features for phase two deployment of the tool while others are potential ideas proposed in this thesis to push 5G radio testing even further with the SCT.

7.1 Improved movement syntax

To support the creation of more diverse test sequences, the syntax for creating antenna pattern movement can be improved in the future. Instead of setting starting coordinates for the antenna pattern and then giving step by step instructions for the movement, there could be an alternative option to set coordinates in between of the steps, allowing the pattern to switch to a different coordinate instantly. These coordinate changes could be performed by writing the desired coordinates into the syntax of pattern movement. There could also be an option to change the step size i.e., the number of antennas skipped when moving one step towards the desired direction. This added functionality could also be used to change the attenuation and delay values on the fly, though this would most likely work only as a general value change for all the antennas used, as this would otherwise result in over complicating the syntax invalidating the objective of the tool to make creating sequences simple and fast. Table 4 below contains options to achieve coordinate and sequence movement changes by using horizontal and elevation angles as well as step sizes or alternatively x, y coordinates.

Table 4: Improved sequence syntax alternatives

Desired effect	**Syntax option using degree values**	**Syntax option using x, y coordinate values**
Move right towards coordinates e.g., on one sector or multiple sectors.	h0/e45->h50/e45 or h0/e75->h100/e75	5/2->10/2 or 5/3->20/2
Chain movements to move back and forth.	h0/e45->h50/e45;h50/e45-<h0/e45	5/2->10/2;10/2-<5/2
Start from a certain position, move down 5 steps and right 3 steps with antenna step size of 1, then change antenna step size to 2 and move left 6 steps.	h0/e45/s1vvvvv>>>s2<<<<<<	5/2/s1vvvvv>>>s2<<<<<<
Use previous values but only change elevation.	h0/e45/s1vvvvvh/e60/s>>>	5/2/s1vvvvvh/e3/s>>>

The tool can be used to create sequences moving from a coordinate to another one, but it cannot be used in conjunction with the step-by-step directions. The syntax to create movement for the test sequence can be enhanced to create a wider range of more complex test scenarios. The sequence syntax should be developed to cater to different user needs while remaining simple and easy to understand.

7.2 Multi-UE support

An essential addition for the tool defined as a functionality for phase two delivery is multi-UE support. This would allow the user to define sequence inputs for multiple UEs separately and the tool would be responsible for combining them and generating one sequence. Multi-UE sequence creation is a time-consuming process and automating this would save a considerable amount of time. Multi-UE sequence creation is a necessary feature for the tool as the radios must be tested to support multiple users simultaneously in real user scenarios. Multi-UE sequences could also be created by combining previously generated single UE sequences by implementing a comprehensive conflict handling system within the tool also supports when creating these sequences. Multi-UE sequences are susceptible to conflicts especially when creating them manually.

7.3 Conflict handling

The small conflicts are resolved with a simple algorithm e.g., allocating a conflicting antenna to the next possible coordinate found. Bigger conflicts could be handled by notifying the user and sequence generation would be prevented until user has fixed the found conflict. The tool would handle all the most common conflicts and offer solutions to the user to handle the more complex conflicts. Implementing a robust error handling system for the tool to address unexpected inputs or scenarios while providing helpful recommendations to the user would be highly beneficial as the new added functionalities might otherwise confuse a new user. Correction suggestions would help when the parameters set by the user might not lead to viable patterns or their movement.

Conflict handling would have an even more important role when the multi-UE support is added to the tool, as these sequences are vastly more complex and require knowledge of the physical limitations of the test environment. If these conflicts are handled automatically, the user does not need to worry about debugging the sequence to make corrections. Instead, the user is notified of changes made or instructed to make required corrections.

7.4 React frontend as a graphical user interface

Another functionality defined for the tool is to have a GUI available as a web application which would have a frontend created with React. The GUI would work by having an antenna grid which the user can interact with, clicking and choosing from the grid which antennas and their polarities are used for the pattern, to create and modify antenna patterns. The GUI would create an even lower threshold for users to create more complex patterns and sequences. Designing an intuitive and user-friendly interface that allow the users to input the required parameters to create an antenna pattern and the sequence and to also have a visual representation of the test environment and the created pattern available would aid when creating complex and comprehensive test sequences.

Having a GUI as a web application would also allow not only the visualization of antenna patterns, which is also available with the CLI version, but also the created sequences.

Visual representations of the generated sequences would be shown to the user to better understand and evaluate the created test and make required modifications. This would be achieved by displaying the antenna pattern movement on the antenna grid of the test environment. The sequence could be previewed with the GUI of the tool.

7.5 Collaboration and sharing

The ability to share the generated patterns and sequences with other testers should be made simple. Instead of sharing the created files through other channels, the tool itself could have a functionality to save patterns and sequences as personal files as well as share and collaborate with others. Providing a simple way to export and import these files within the tool would prevent from creating duplicate test sequences that someone else has already created and saved in the database of patterns and sequences, thus saving valuable time in testing.

Collaboration between testers and developers could also be strengthened by incorporating an option to leave feedback in the tool. The feedback mechanism would continuously gather insight from active users that want to participate in the continuous development of the tool to implement necessary additions. This would help in prioritizing tasks when planning the development of the tool and diminish the time wasted waiting for feedback that would be requested on the spot and ensuring that the feedback given is carefully thought out.

8 SUMMARY

Throughout the course of the project, several valuable lessons were learned. The complexity of testing 5G radios and the importance having sophisticated automation in testing became clear during the development of the Sequence Creator Tool. The integration of various customisable parameters within the tool proved to be crucial in generating accurate and comprehensive test sequences. Additionally, the project highlighted the fast evolution of 5G technology and the testing methodologies, which requires tools that can adapt and evolve alongside the rapidly advancing telecommunication technologies.

The achievements of the thesis project were plentiful. The first version of a functional tool specifically designed for 5G radio testing was successfully developed. This achievement not only addressed the need for such a tool within the field but also positioned the project as a potential factor for further innovations in 5G testing methodologies. The project also contributed to building personal academic understanding of the challenges and opportunities associated with 5G radio testing.

Undoubtedly, certain aspects turned out to be challenging during the development and implementation of the Sequence Creator Tool. The dynamic nature of 5G technology presented challenges in anticipating all potential scenarios and parameters that should be considered in sequence generation. Ensuring the adaptability of the tool to diverse testing conditions and configurations required careful consideration and continuous refinement. Balancing the ability to create complex sequences with creating a user-friendly tool was a challenge, as the tool needed to generate complex test sequences while being intuitive to a broad range of users within radio testing teams. Overall, these challenges added depth to the learning experience and contributed to the functionality of the developed tool.

9 REFERENCES

[1] Broadband Compared, "What is 4G? Everything about 4G Explained," Broadband Compared, [Online]. Available: https://www.broadbandcompared.co.uk/guides/what-is-4g-everything-about-4g-explained. [Accessed 1 September 2023].

[2] Cloudflare, "What is the Internet Protocol?," [Online]. Available: https://www.cloudflare.com/learning/network-layer/internet-protocol/. [Accessed 1 September 2023].

[3] Ecotone, "RF Chambers," Ecotone, [Online]. Available: https://www.rfchambers.com/rf-shielded-chamber.html. [Accessed 12 May 2023].

[4] P. Salmon, "LTE vs. 4G: What's the Difference?," History-Computer, [Online]. Available: https://history-computer.com/lte-vs-4g-whats-the-difference/. [Accessed 1 November 2023].

[5] I. Nicholls, "4G vs LTE: Understanding the Difference Between 4G and LTE," UCtel, [Online]. Available: https://www.uctel.co.uk/blog/4g-vs-lte-understanding-the-difference-between-4g-and-lte. [Accessed 1 November 2023].

[6] S. M. Blust, "Development of IMT-Advanced: The SMaRT approach," ITU, [Online]. Available: https://www.itu.int/itunews/manager/display.asp?lang=en&year=2008&issue=10&ipage=39&ext=html. [Accessed 10 November 2023].

[7] H. Remmert, "What is LTE: How It Works and Why It Matters," Digi International Inc., [Online]. Available: https://www.digi.com/blog/post/what-is-lte. [Accessed 1 November 2023].

[8] 3GPP, "Introducing 3GPP," 3GPP, [Online]. Available: https://www.3gpp.org/about-us/introducing-3gpp . [Accessed 2 November 2023].

[9] 3GPP, "5G System Overview," 3GPP, [Online]. Available: https://www.3gpp.org/technologies/5g-system-overview . [Accessed 2 November 2023].

[10] D. Darah, "5G NSA vs. SA: How do the deployment modes differ?," TechTarget, [Online]. Available: https://www.techtarget.com/searchnetworking/feature/5G-NSA-vs-SA-How-does-each-deployment-mode-differ. [Accessed 2 November 2023].

[11] S. Tombaz, “Non-standalone and Standalone: two standards-based paths to 5G,” Ericsson, [Online]. Available: https://www.ericsson.com/en/blog/2023/4/standalone-and-non-standalone-5g-nr-two-5g-tracks. [Accessed 2 November 2023].

[12] 5GStore, “5G Networks: Non-Standalone vs. Standalone,” 5GStore, [Online]. Available: https://5gstore.com/blog/2023/12/21/5g-networks-non-standalone-vs-standalone/. [Accessed 16 January 2024].

[13] J. Sikes, “Taking 5G to the Next Level with Standalone 5G,” AT&T, [Online]. Available: https://about.att.com/blogs/2023/standalone-5g-innovations.html. [Accessed 16 January 2024].

[14] J. G. Gregory, “What is 5G core,” Verizon, [Online]. Available: https://www.verizon.com/business/resources/articles/s/what-is-5g-core-and-how-it-can-help-benefit-business/. [Accessed 23 January 2024].

[15] U. Shahab, “Why Standalone is vital for reaching 5G maturity,” Ericsson, [Online]. Available: https://www.ericsson.com/en/blog/2021/12/why-standalone-is-vital-for-reaching-5g-maturity. [Accessed 23 January 2024].

[16] J. Hollington, “What frequency is 5G? All the different 5G ranges, explained,” Digital Trends Media Group, [Online]. Available: https://www.digitaltrends.com/mobile/what-frequency-is-5g-all-the-different-5g-ranges-explained/#dt-heading-the-speed-and-range-trade-off. [Accessed 23 January 2024].

[17] A. Ghayas, “5G SA and 5G NSA: Standalone vs Non-Standalone 5G NR,” Commsbrief, [Online]. Available: https://commsbrief.com/5g-sa-and-5g-nsa-standalone-vs-non-standalone-5g-nr/. [Accessed 26 January 2024].

[18] A. Ghayas, “What is 5G Massive MIMO technology and how does it work?,” Commsbrief, [Online]. Available: https://commsbrief.com/what-is-5g-massive-mimo-technology-and-how-does-it-work/. [Accessed 10 November 2023].

[19] P. v. B. S. F. E. L. David Astely, “Meeting 5G network requirements with Massive MIMO,” Ericsson, [Online]. Available: https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/using-massive-mimo-to-meet-5g-network-requirements. [Accessed 15 December 2023].

[20] R. W. H. J. Ryan M. Dreifuerst, “Massive MIMO in 5G: How Beamforming, Codebooks, and Feedback Enable Larger Arrays,” [Online]. Available: https://arxiv.org/pdf/2301.13390.pdf. [Accessed 16 January 2024].

[21] M. Shukair, “How 5G massive MIMO transforms your mobile experiences,” Qualcomm Technologies, [Online]. Available: https://www.qualcomm.com/news/onq/2019/06/how-5g-massive-mimo-transforms-your-mobile-experiences. [Accessed 15 December 2023].

[22] HOLOPLOT, “A revolution in sound control,” HOLOPLOT, [Online]. Available: https://holoplot.com/technology/?utm_term=beamforming&utm_campaign=SearchAds-EN_NoneGerman-Generic-Features&utm_source=adwords&utm_medium=ppc&hsa_acc=6908937639&hsa_cam=6447938308&hsa_grp=78542224066&hsa_ad=386148250047&hsa_src=g&hsa_tgt=kwd-111495139&hsa. [Accessed 10 November 2023].

[23] Lumenci, “5G Beamforming,” Lumenci, [Online]. Available: https://www.lumenci.com/research-articles/5g-beamforming. [Accessed 16 January 2024].

[24] Electricity - Magnetism, “Parabolic Antennas,” Electricity - Magnetism, [Online]. Available: https://www.electricity-magnetism.org/parabolic-antennas/. [Accessed 15 December 2023].

[25] K. Benson, “Phased Array Beamforming ICs Simplify Antenna Design,” Analog Devices, [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/phased-array-beamforming-ics-simplify-antenna-design.html. [Accessed 15 December 2023].

[26] Cadence Design Systems, “Phased Array Antennas: Principles, Advantages, and Types,” Cadence Design Systems, [Online]. Available: https://resources.system-analysis.cadence.com/blog/msa2021phased-array-antennas-principles-advantages-and-types. [Accessed 15 December 2023].

[27] Wikipedia, “Phased array,” [Online]. Available: https://en.wikipedia.org/wiki/Phased_array#/media/File:Phased_array_animation_with_arrow_10frames_371x400px_100ms.gif. [Accessed 16 January 2024].

[28] D. S. Arar, “Understanding Anechoic Chambers for Electromagnetic and RF Testing,” EETech Media, [Online]. Available: https://www.allaboutcircuits.com/technical-articles/understanding-rf-anechoic-chambers-for-electromagnetic-and-radio-frequency-testing/. [Accessed 24 January 2024].

[29] G. Dash, “How RF Anechoic Chambers Work,” MIT Alumni Association, Cambridge, 2005.

[30] T. H. a. M. T. K. Shimada, “"Fully compact anechoic chamber using the pyramidal ferrite absorber for immunity test," IEEE International Symposium on Electromagnetic Compatibility. Symposium Record (Cat. No.00CH37016),” Washington, DC, USA, 2000.

[31] RF Electronics, “Products: RF Chambers,” RF Electronics, [Online]. Available: https://www.rfelectronics.net/products/rf-chambers. [Accessed 11 May 2023].

[32] L. Rosencrance, “Definition: attenuation,” TechTarget, [Online]. Available: https://www.techtarget.com/searchnetworking/definition/attenuation. [Accessed 26 January 2024].

[33] Microwaves & RF, “New Technology Redefines Channel Emulation for 5G Millimeter-Wave Systems,” Microwaves & RF, [Online]. Available: https://www.mwrf.com/technologies/test-measurement/article/21848731/new-technology-redefines-channel-emulation-for-5g-millimeter-wave-systems. [Accessed 26 January 2024].

[34] A. A. Y. A. M. A. B. A. &. D.-y. C. Farzana Arshad, “MIMO antenna array with the capability of dual polarization reconfiguration for 5G mm-wave communication,” Springer Nature Limited, [Online]. Available: https://www.nature.com/articles/s41598-022-23163-3. [Accessed 25 January 2024].

[35] C. Masterson, “Massive MIMO and Beamforming: The Signal Processing Behind the 5G Buzzwords,” Analog Devices, [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/massive-mimo-and-beamforming-the-signal-processing-behind-the-5g-buzzwords.html. [Accessed 25 January 2024].

[36] Keysight Technologies, “Channel Emulation Solutions,” Keysight Technologies, [Online]. Available: https://www.keysight.com/us/en/products/channel-emulators.html. [Accessed 25 January 2024].

[37] G. Andrades, “What is Unit Testing? 6 Best Practices to Do it Right,” ACCELQ, [Online]. Available: https://www.accelq.com/blog/unit-testing/. [Accessed 16 January 2024].

[38] B. Kaberia, “What Is Unit Testing: Detailed Guide With Best Practices,” [Online]. Available: https://www.lambdatest.com/learning-hub/unit-testing. [Accessed 16 January 2024].