A

Project Report

on

5G NR Downlink Transport channel modeling using MatLab

submitted as fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2022-23

in

Electronics and Communication

By

Mridul Nigam (1900290310082)

Under the supervision of

Dr. Narendra Kumar

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A. P. J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2023

DECLARATION

I hereby declare that this submission is our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name: Mridul Nigam

Roll No.: 1900290310082

Date:

CERTIFICATE

This is to certify that Project Report entitled "5G NR DOWNLINK TRANSPORT CHANNEL MODELLING USING MATLAB" which is submitted by Mridul Nigam in fulfilment of the requirement for the award of degree B. Tech. in Department of Electronics & Communication Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Date: Dr. Narendra Kumar

(Assistant Professor)

ACKNOWLEDGEMENT

It gives me a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. I owe special debt of gratitude to Dr. Narendra Kumar, Assistant Professor, Department of Electronics & Communication Engineering, KIET Group of Institutions, Ghaziabad, for his constant support and guidance throughout the course of my work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for me. It is only his cognizant efforts that my endeavours have seen light of the day.

I also take the opportunity to acknowledge the contribution of Dr. Vibhav Kumar Sachan, HoD, Electronics and Communication Engineering Department, KIET Group of Institutions, Ghaziabad, for his full support and assistance during the development of the project. I also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of my project. Last but not the least, I acknowledge my friends for their contribution in the completion of the project.

Date:

Sig. (s):

Name (s): Mridul Nigam

Roll No. : 1900290310082

ABSTRACT

The fifth generation (5G) of mobile communication networks offers significant improvements in terms of data rates, latency, and reliability. One critical component of the 5G system is the transport channel, which facilitates data transmission between the base station and the user equipment. Accurate modelling of the Downlink transport channel is essential to optimize network performance and ensure seamless communication. In this study, we investigate the modelling of the downlink transport channel in 5G NR using MATLAB. We employ simulation-based techniques to evaluate the transport channel's performance under various scenarios and configurations. Our findings demonstrate the efficacy of the MATLAB-based approach in accurately predicting downlink transport channel characteristics, facilitating optimization of network design and deployment. This research contributes to the field of 5G NR downlink transport channel modelling by validating the effectiveness of MATLAB-based simulation techniques, enabling further study and refinement of network performance optimization.

This report explains how to model 5G NR DOWNLINK TRANSPORT CHANNELs with multiple hybrid automatic repeat-request (HARQ) processes using the downlink shared channel (DL-SCH) encoder and decoder 5G Toolbox System objects. By using this DOWNLINK TRANSPORT CHANNEL model with any wireless system design in MATLAB and Simulink, one can optimize link performance, perform system architecture trade-offs, and provide a realistic assessment of the overall system performance. It can also be used to predict link performance (e.g., BER) in a single-user scenario, system performance (e.g., throughput, latency) in a multi-user scenario and reduce the need for costly channel measurement projects.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3GPP	3rd generation partnership project
5G	Fifth Generation
5GC	5G Core Network
AF	Application Function
BCH	Broadcast Channel
BWP	Bandwidth Part
CDSA	Control/Data Separation Architecture
CHF	Charging Function
CN	Core Network
CP-OFDM	Orthogonal Frequency Division Multiplexing with Cyclic Prefix
CQI	Channel Quality Indicator
CSI	Channel State Information
DL-SCH	Downlink Shared Channel
eMBB	enhanced Mobile Broadband
EN-DC	E-UTRAN NR-Dual Connectivity EPC Evolved Packet Co
mMTC	massive Machine Type Communications
IMT Adv	International mobile telecommunications-advanced
ITU	International telecommunication union
IEEE	Institute of electrical and electronics engineers
NG	Next Generation
ng-eNB	new generation LTE NodNG Next Generation
ng-eNB	new generation LTE NodeBeB
PL	Pathloss
PS	Propagation scenario
PCH	Paging Channel
PDSCH	Physical Downlink Shared Channel
RX	Receiver
TX	Transmitter
UE	User equipment

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

5G is the major phase of mobile telecommunications standards. The scope of 5G ultimately range from mobile broadband services to next-generation automobiles and connected devices. The initial 5G New Radio (NR) specification was completed in June 2018 and published in the 3GPP Release 15 specification. Now, a variety of industry players, including network equipment vendors, network operators, semiconductor vendors, and device manufacturers, are developing new products that implement the new standard.

Two major trends are behind the race to 5G: the explosive growth in demand for wireless broadband that can carry video and other content-rich services, and the Internet of Things (IoT), where large numbers of smart devices communicate over the Internet. To achieve these objectives, 5G will provide **extreme broadband speed**, **ultralow latency**, and **ultrareliable** web connectivity. 5G networks and devices will require different architectures, radio access technology, and physical layer algorithms. Dense networks of small cells will complement macro base stations, operating at millimetre wave technologies and employing massive MIMO antenna arrays. And the processing components within network equipment and user devices will become more integrated and adaptive. Innovations like hybrid beamforming are stretching the old ways of developing wireless systems. These highly integrated technologies require a corresponding integration of engineering domain expertise and tools.

5G New Radio (NR) is the air interface supporting the next generation of mobile communication, commonly referred to as fifth generation or 5G. The predecessors of 5G NR are GSM, UMTS, and LTE, also referred to as second generation (2G), third generation (3G), and fourth generation (4G) technologies, respectively. GSM primarily enabled voice calls. The redesigned interfaces of UMTS and LTE enabled and gradually improved mobile broadband connectivity with high data rates and high efficiency. 5G NR continues the path of LTE by enabling much higher data rates and much higher efficiency for mobile broadband. However, as a response to the demands of networked society, the scope of 5G NR goes beyond mobile broadband connectivity. The main requirement of 5G NR is to enable wireless connectivity everywhere, at any time to anyone and anything.

5G NR specification is developed by the Third Generation Partnership Project (3GPP). The first release of the standard was frozen in mid-2018 as 3GPP 5G NR Release 15. 5G Toolbox provides implementations for a subset of the 5G NR physical layer specification and channel model specifications.

1.2 PROJECT DESCRIPTION

The outcome of this project is in the form of a research paper in which I am describing the challenges and opportunities faced in modelling 5G NR DOWNLINK TRANSPORT CHANNEL using MATLAB. A channel model is also known as radio wave propagation model, it characterizes radio wave propagation as a function of frequency, distance, environment, and other factors. In other words, we can say that channel model provides us rough idea how much distance a signal can travel in a certain environment (morphology like urban, suburban, or rural etc.) with the known transmitter and receiver height.

A channel model is usually developed to predict the behaviour of propagation of radio signal, for all similar channel under and similar constraints (environment, channel fading, multi-path etc). Channel models typical predict the path loss along a wireless link or effective coverage area of a transmitter. It is a mathematical representation of the effects of a communication channel through which wireless signals are propagated. The channel model can represent the power loss incurred by the signal as it travels through the wireless medium. In a more general case, the channel model is the impulse response of the channel medium in the time domain, or its Fourier transform in the frequency domain. The channel impulse response of a wireless communication system typically varies randomly over time.

By using this DOWNLINK TRANSPORT CHANNEL model with any wireless system design in MATLAB and Simulink, one can optimize link performance, perform system architecture trade-offs, and provide a realistic assessment of the overall system performance. It can also be used to predict link performance (e.g., BER) in a single-user scenario, system performance (e.g., throughput, latency) in a multi-user scenario and reduce the need for costly channel measurement projects.

This report explains how to model 5G NR DOWNLINK TRANSPORT CHANNELs with multiple hybrid automatic repeat-request (HARQ) processes using the downlink shared channel (DL-SCH) encoder and decoder 5G Toolbox System objects. By using this DOWNLINK TRANSPORT CHANNEL model with any wireless system design in MATLAB and Simulink, one can optimize link performance, perform system architecture trade-offs, and provide a realistic assessment of the overall system performance. It can also be used to predict link performance (e.g., BER) in a single-user scenario, system performance (e.g., throughput, latency) in a multi-user scenario and reduce the need for costly channel measurement projects.

CHAPTER 2

LITERATURE REVIEW

2.1 5G TERMINOLOGY

By providing higher bandwidth capacity than current 4G–supporting broadband, 5G will enable a higher density of mobile broadband users and support ultrareliable device-to-device and massive machine-type communications.

a) **eMBB—Enhanced Mobile Broadband.**
For high-capacity and ultrafast mobile communications for phones and infrastructure, virtual and augmented reality, 3D and ultra-HD video, and haptic feedback

b) **URLLC—Ultrareliable and Low Latency**
For vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, autonomous driving.

c) **mMTC—Massive Machine-Type Communications**
For consumer and industrial IoT, Industry 4.0 mission-critical machine-to-machine (MC-M2M)

2.2 5G KEY PARAMETERS

Latency in the air link	<1 ms
Latency end-to-end (device to core)	<10 ms
Connection density	100x vs. current 4G LTE
Area capacity density	1 (Tbit/s)/km2
System spectral efficiency	10 Gbit/s
Peak throughput (downlink) per connection	10 (bit/s)/Hz/cell
Energy efficiency	>90% improvement over LTE

Table 1.1: 5G Key Parameters

2.3 THE 5G NR ARCHITECTURE

The 5G NR architecture comprises **next-generation RAN (NG-RAN)** and **5G Core Network (5GC)**, both described in the following subsections.

2.3.1 The NG-RAN

It includes new generation **LTE eNodeB (ng-eNB)** and **5G NodeB (gNB),** which are responsible for the radio functions, e.g., RRM, admission and connection control, and **Quality of Service (QoS)** flow management. The ng-eNB employs **Evolved-Universal Terrestrial Radio Access (E-UTRA)** user-/control- plane protocols to serve LTE **User Equipment (UEs)** and is connected to the 5GC via the NG interface.

Figure 2.1 depicts the NG-RAN architecture consisting of a set of gNBs. The gNB employs NR user-/control- plane protocols to serve NR UEs and is connected to the 5GC via the NG interface and to other gNBs through the Xn interface.

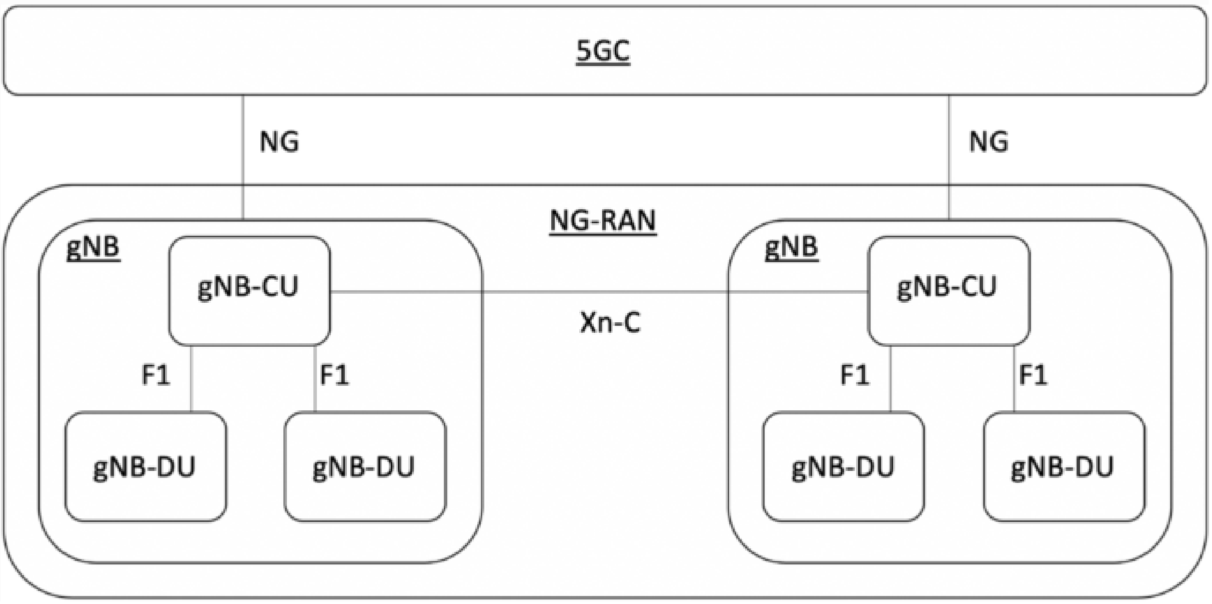


Figure 2.1. NG- RAN

The gNB consists of a central unit (i.e., gNB-CU) and one or more distributed units (i.e., gNB-DU). One gNB-DU is connected to only one gNB-CU via F1 interface. NG, Xn, and F1 are logical interfaces. The Xn-C interface interconnects gNB-CUs of different gNBs. The gNB can also consist of a gNB-CU control-plane (gNB-CU-CP), multiple gNB-CU user-plane (gNB-CU-UPs), and multiple gNB-DUs.

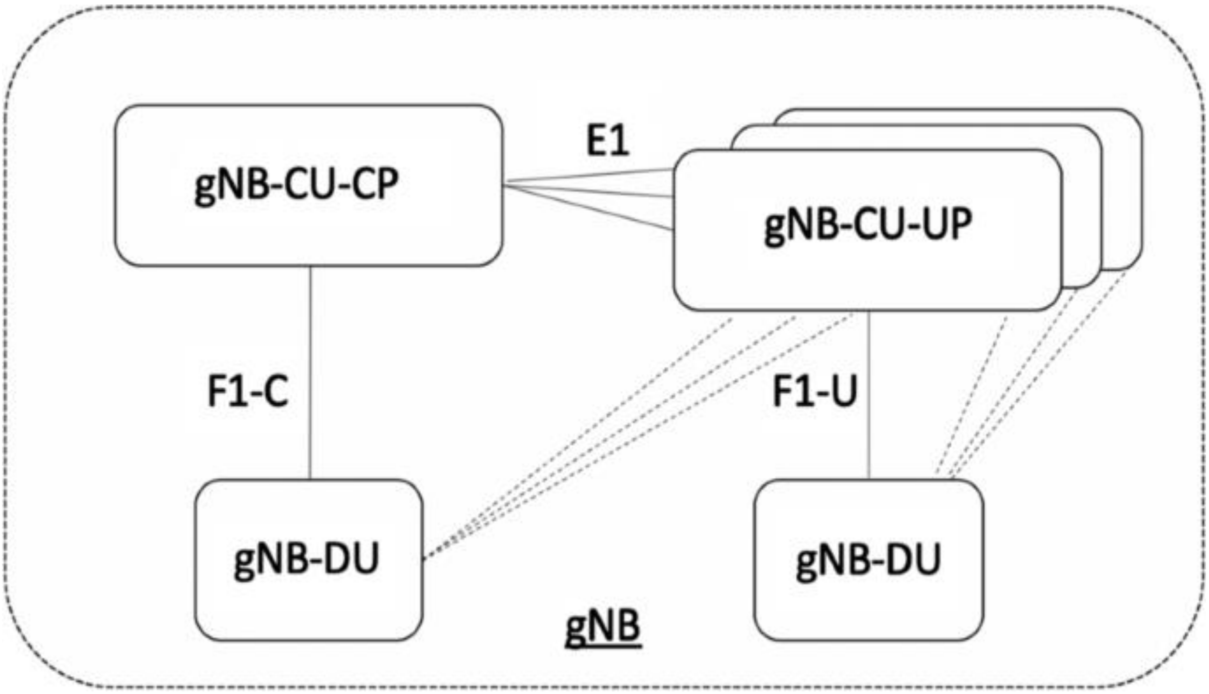


Figure 2.2. gNB architecture with separation of gNB-CU-CP and gNB-CU-UP

Figure 2.2 illustrates the overall architecture with separation of the control-plane and the user-plane for the gNB-CU (i.e., gNB-CU-CP and gNB-CU-UP). The gNB-CU-CP is connected to the gNB-DU through the F1-C interface. A gNB-CU-UP is connected to the gNB-DU through the F1-U interface and to only one gNB-CU-CP through the E1 interface. The gNB-CU-UP is connected to only one gNB-CU-CP, to multiple gNB-CU-UPs and multiple gNB-DUs under the control of the same gNB-CU-CP.

The NG-RAN offers new functions, such as: (i) network slicing, (ii) contacting UEs in inactive mode, (iii) handover between E-UTRA and NR via a direct interface between eNB and gNB, (iv) handover between E-UTRA and NR via core network (CN), (v) session management, and (vi) tight interworking between NR and E-UTRA, and dual connectivity

2.3.2 The 5G Core Network

Figure 2.3 shows the high-level representation of the 5GC. Detailed descriptions (i.e., reference architecture type, reference points, and point-to-point interactions between the functions) are provided in the specification [18].

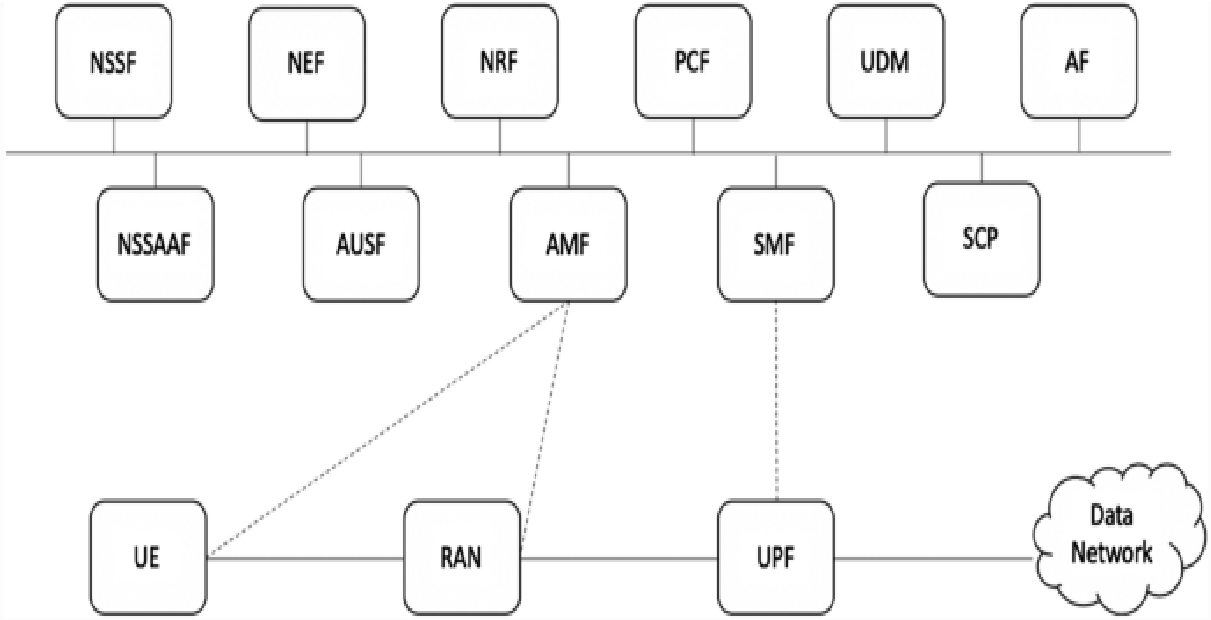


Figure 2.3. High-level representation of the 5G Core Network

The 5GC follows a service-based architecture, supports network slicing, and splits the user-plane and the control-plane. The *User Plane Function* (UPF) acts as a gateway to connect the RAN to external networks. In certain cases, it also represents the anchor point for intra-/inter-Radio Access Technology (RAT) mobility. UPF is responsible for packet routing, forwarding, and inspection, handling QoS, and managing traffic measurements.

Control Plane Functions (CPF) are as follows. The *Session Management Function* (SMF) handles session management and establishment, allocates IP addresses to the UEs, facilitates roaming, and controls the UPF. The *Access and Mobility Management Function* (AMF) manages registration, reachability, mobility, connection, and location services. AMF also handles access authentication and authorization and facilitates idle-state mobility. The *Non-Access Stratum* (NAS) operates in-between the AMF and the device, while the *Access Stratum* (AS) operates in-between the device and the RAN.

Other types of functions and entities are: *Unified Data Management* (UDM) that authenticates and authorizes access, *Policy Control Function* (PCF) providing policy rules, *Authentication Server Function* (AUSF) handling authentication, *Application Function* (AF) influencing the traffic routing, *Network Exposure Function* (NEF) providing secure information from external application to 3GPP network, *Network Repository Function* (NRF) supporting service discovery function, *Unified Data Repository* (UDR) responsible of storage and retrieval of subscription data by UDM, *Unstructured Data Storage Function* (UDSF) responsible of storage and retrieval of unstructured data by any network function, *Network Data Analytics Function* (NWDAF) managing network analytics, *Network Slice Specific Authentication and Authorization Function* (NSSAAF) and *Network Slice Selection Function (NSSF)* in charge of Network Slicing, *UE radio Capability Management Function* (UCMF) storing all UE Radio Capability ID, *5G-Equipment Identity Register* (5G-EIR) checking the status of Permanent Equipment Identifier (PEI) (e.g., whether it has been blacklisted), and Charging Function (CHF) that manages charging information

2.4 5G NR PROTOCOL LAYERS

The 5G NR radio access network is comprised of these protocol entities:

i. Service data adaptation protocol (SDAP)
ii. Packet data convergence protocol (PDCP)
iii. Radio link control (RLC)
iv. Medium access control (MAC)
v. Physical layer (PHY)

The SDAP protocol is new in 5G NR compared to the LTE protocol stack. SDAP handles the new QoS framework of the 5G System (in the 5G Core). SDAP applies also to LTE when connected to the 5G Core. The introduction of SDAP enables end-to-end QoS framework that works in both directions. To meet the desired key capabilities of 5G NR, the other layers of the stack provide various enhancements over their LTE counterparts. The PDCP, RLC, and MAC protocols handle tasks such as header compression, ciphering, segmentation, and concatenation, and multiplexing and demultiplexing. PHY handles coding and decoding, modulation and demodulation, and antenna mapping.

The figure 2.4 shows the 5G NR user plane protocol stack for user equipment (UE) and the NR radio access network node (gNB). 5G Toolbox supports the 5G NR physical layer, including physical channels and signals. The toolbox also supports interfacing with portions of the RLC and MAC layers, including DOWNLINK TRANSPORT CHANNELs and logical channels.

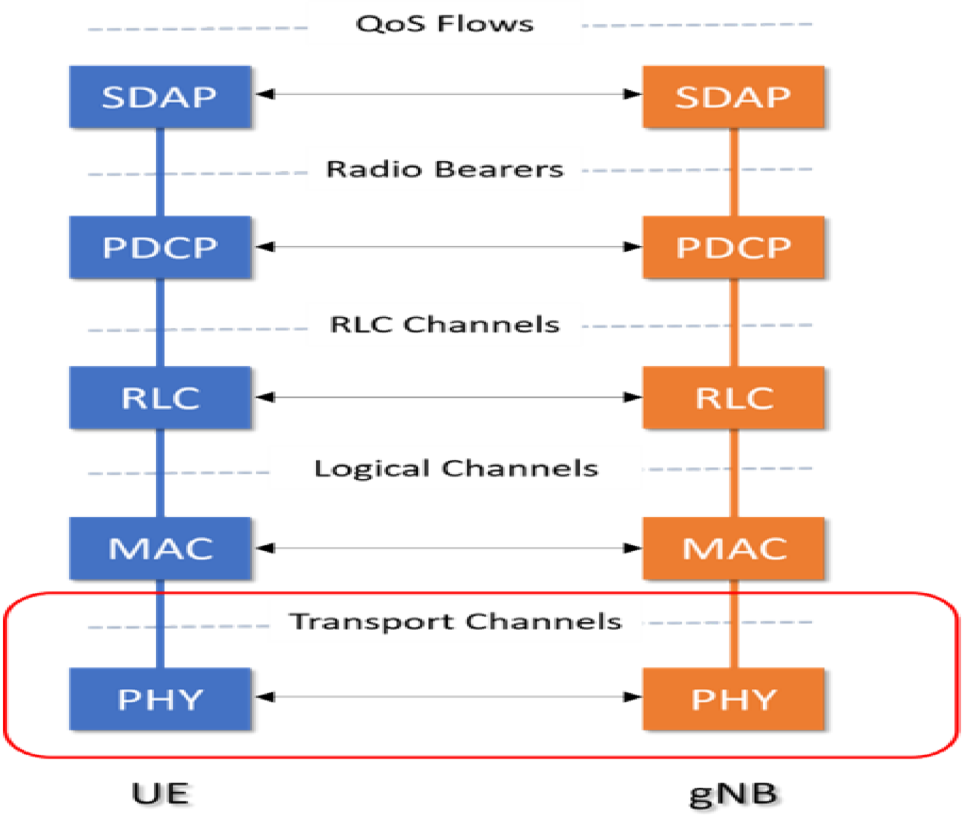


Figure 2.4. 5G NR protocol stack

2.5 CHANNEL MAPPING

2.5.1 Downlink Channel Mapping

5G NR system downlink data follows the mapping between logical channels, DOWNLINK TRANSPORT CHANNELs, and physical channels, as indicated in the diagram. 5G Toolbox provides the red-highlighted downlink functionality for physical channels, DOWNLINK TRANSPORT CHANNELs, and control information.

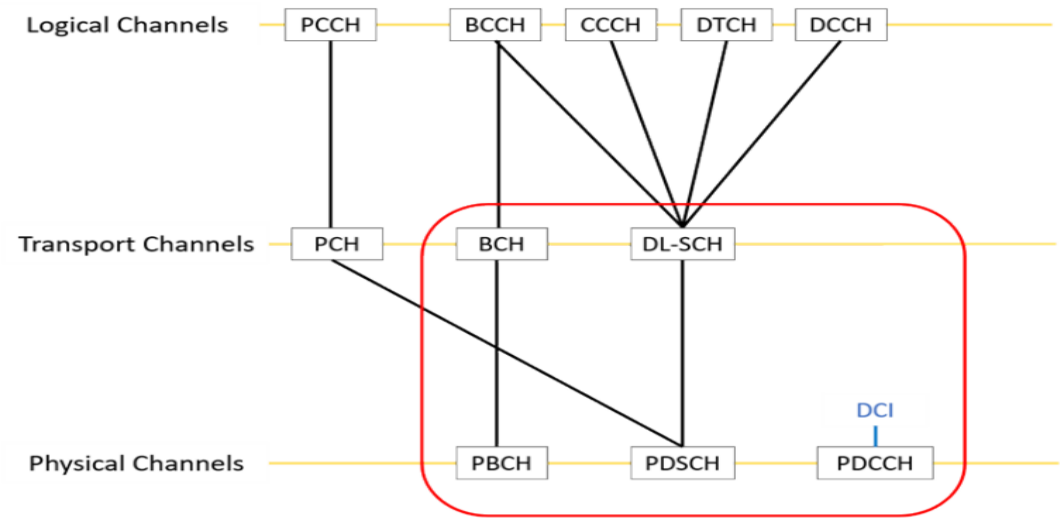


Figure 2.5. Downlink channel mapping

2.5.2 Uplink Channel Mapping

5G NR system uplink data follows the mapping between logical channels, DOWNLINK TRANSPORT CHANNELs, and physical channels, as indicated in the diagram. 5G Toolbox provides the red-highlighted uplink functionality for physical channels, DOWNLINK TRANSPORT CHANNELs, and control information.

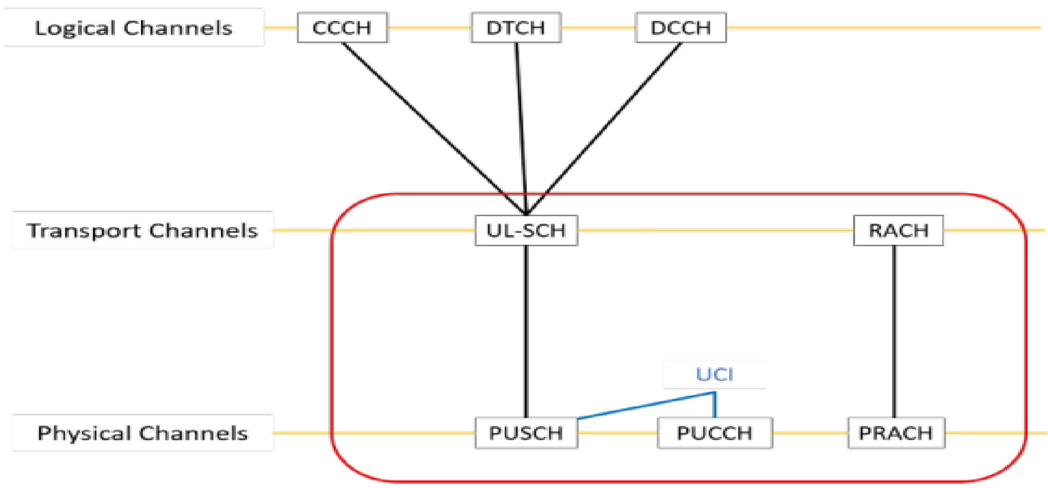


Figure 2.6. Uplink channel mapping

2.6 5G NR DOWNLINK TRANSPORT CHANNELS

There are five different DOWNLINK TRANSPORT CHANNELs. Some are used on the uplink, others on the downlink, and some can be used on both.

i. **Broadcast Channel, BCH:**

The BCH 5G channel is used in the downlink only for transmitting the BCCH system information and specifically the Master Information Block, MIB, information. In order that the data can be utilised, it has a specific format.

ii. **Paging Channel, PCH:**

The PCH is used for carrying paging information from the PCCH logical channel. The PCH supports discontinuous reception, DRX, to enable the UE to save battery power by waking up at a specific time to receive the PCH.
In order that the PCH is received by all mobiles / UEs in the cell, the PCH must be broadcast over the entire cell as a single message, or where beam forming is used, this can be done using several different PCH instances.

iii. **Downlink Shared Channel, DL-SCH**

As the name indicates, this is a downlink only channel. It is the main DOWNLINK TRANSPORT CHANNEL used for transmitting downlink data and it supports all the key 5G NR features. These include dynamic rate adaptation; HARQ, channel aware scheduling, and spatial multiplexing. The DL-SCH is also used for transmitting some parts of the BCCH system information, specifically the SIB. Each UE has a DL-SCH for each cell it is connected to.

iv. **Uplink Shared Channel, UL-SCH**

This is the uplink counterpart to the DLSCH that is, the uplink DOWNLINK TRANSPORT CHANNEL used for transmission of uplink data.

v. **Random-Access Channel, RACH**

The RACH is a DOWNLINK TRANSPORT CHANNEL, which carries the random access preamble which is used to overcome the message collisions that can occur when UEs access the system simultaneously.

2.7 5G NR DOWNLINK TRANSPORT CHANNELS KEY PARAMETERS

In the downlink, there are three distinct types of transport channels defined for NR: the Downlink Shared Channel (DL-SCH), the Paging Channel (PCH), and the Broadcast Channel (BCH), although the last two mentioned are not used in the non-standalone operation.
The overall transport channel processing for NR follows a similar structure

as for LTE. The processing is mostly similar in uplink and downlink and the structure in Fig. 2.7 is applicable for the DL-SCH, BCH, and PCH in the downlink, and the UL-SCH in the uplink. The part of the BCH mapped to the PBCH follows a different structure, as does the RACH.

Within each transmission time interval (TTI), up to two transport blocks of dynamic size are delivered to the physical layer and transmitted over the radio interface for each component carrier. Two transport blocks are only used in the case of spatial multiplexing with more than four layers, which is only supported in the downlink direction and useful in scenarios with very high signal-to-noise ratios. Hence, at most a single transport block per component carrier and TTI is a typical case in practice.

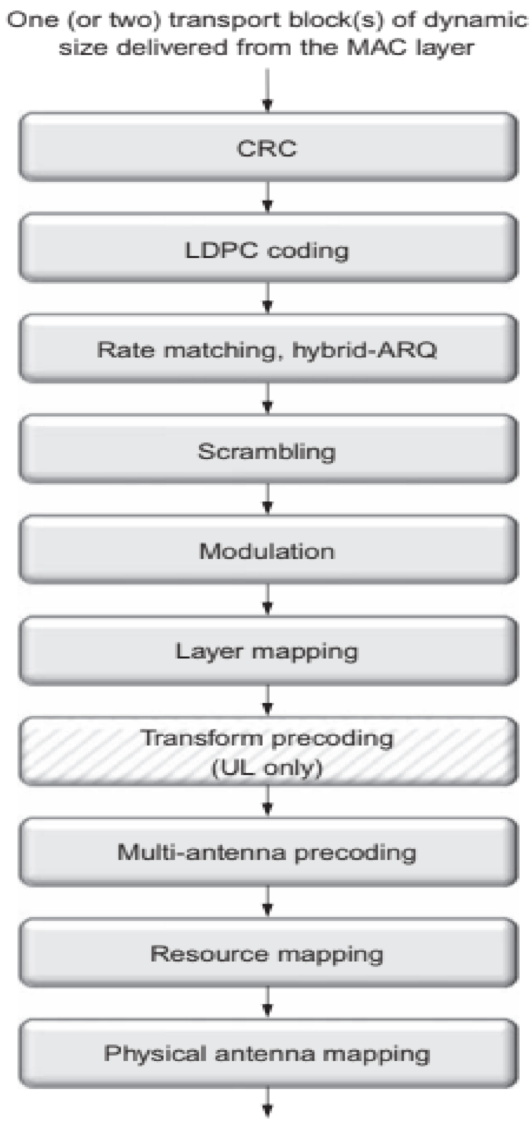


Figure 2.7 Transport channel processing

2.7.1 CRC

A CRC for error-detecting purposes is added to each transport block, followed by error-correcting coding using LDPC codes. Rate matching, including physical-layer hybrid-ARQ functionality, adapts the number of coded bits to the scheduled resources. The code bits are scrambled and fed to a modulator, and finally the modulation symbols are mapped to the physical resources, including the spatial domain.

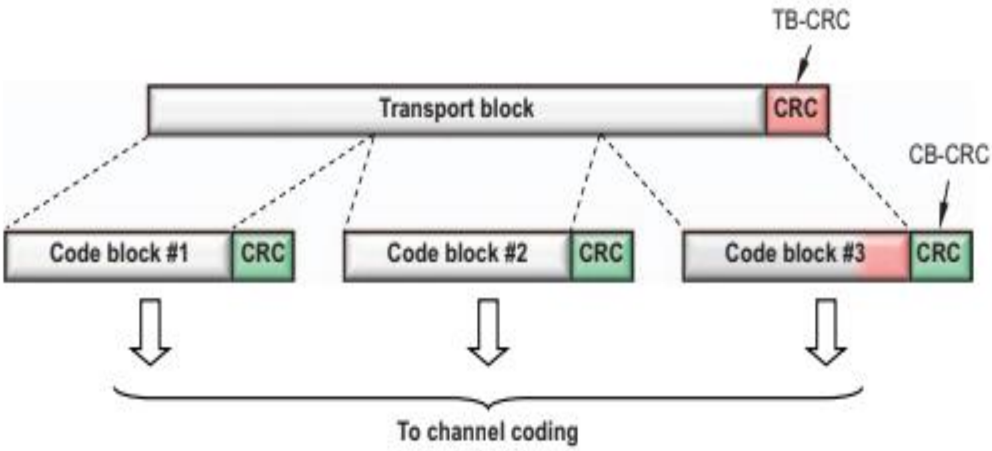


Figure 2.8 CRC Transport Block structure

2.7.2 Channel coding

Channel coding is based on LDPC codes, a code design which was originally proposed in the 1960s but forgotten for many years. They were "rediscovered" in the 1990s and found to be an attractive choice from an implementation perspective. From an error-correcting capability point of view, turbo codes, as used in LTE, can achieve similar performance, but LDPC codes can offer lower complexity, especially at higher code rates, and were therefore chosen for NR. The basis for LDPC codes is a sparse (low-density) parity check matrix H where for each valid code word c the relation $Hc^T = 0$ holds. Designing a good LDPC code to a substantial extent boils down to finding a good parity check matrix H which is sparse (the sparseness implies simple decoding). It is common to represent the parity-check matrix by a graph connecting n variable nodes at the top with (nk) constraint nodes at the bottom of the graph, a notation that allows a wide range of properties of an (n, k) LDPC code to be analysed. This explains why the term base graph is used in the NR specifications. A detailed description of the theory behind LDPC codes is beyond the scope of this report, but there is a rich literature in the field.

Quasi-cyclic LDPC codes with a dual-diagonal structure of the kernel part of the parity check matrix are used in NR, which gives a decoding complexity which is linear in the number of coded bits and enables a simple encoding operation. Two base graphs are defined, BG1 and BG2, representing the two base matrices. The reason for two base graphs instead of one is to handle the wide range of payload sizes and code rates in an efficient way. Supporting an exceptionally large payload size at a medium to high code rate, which is the case for extremely high data rates, using a code designed to support a very low code rate is not efficient. At the same time, the lowest code rates are necessary to provide superior performance in challenging situations. In NR, BG1 is designed for code rates from 1/3 to 22/24 (approximately 0.33~0.92) and BG 2 from 1/5 to 5/6 (approximately 0.2~0.83). Through puncturing, the highest code rate can be increased, up to 0.95, beyond which the device is not required to decode. The choice between BG1 and BG2 is based on the transport block size and code rate targeted for the first transmission (Fig. 2.9).

The base graphs, and the corresponding base matrices, define the general structure of the LDPC code. To support a range of payload sizes, 51 different lifting sizes and sets of shift coefficients are defined and applied to the base matrices.

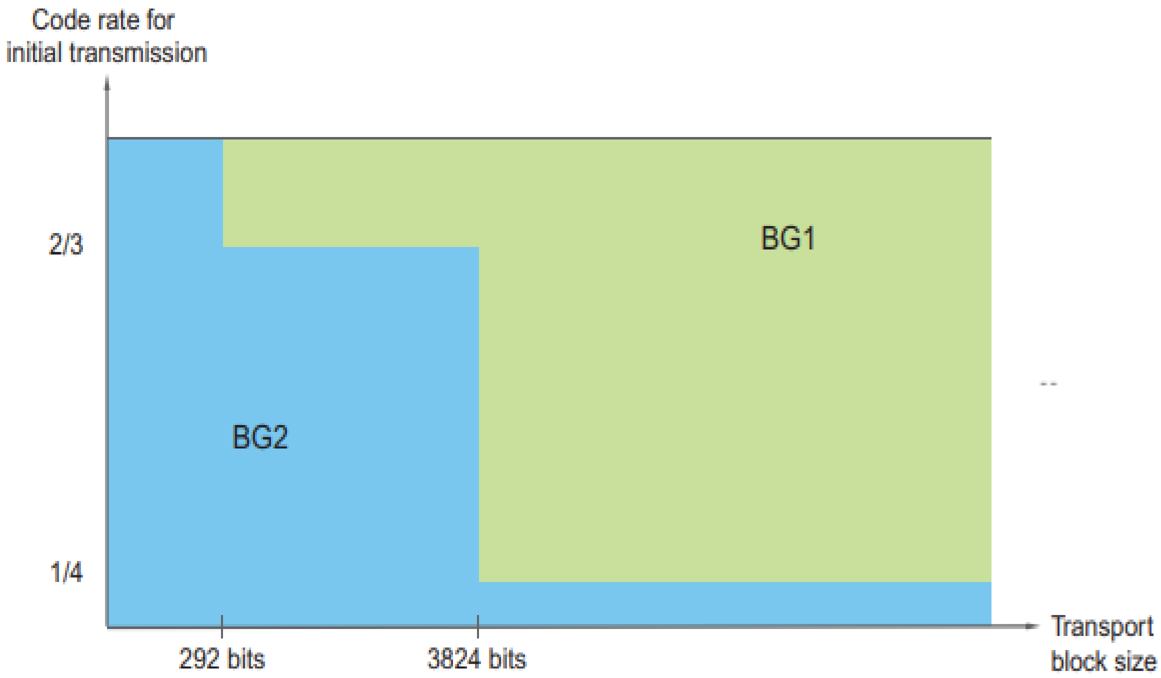


Figure 2.9 Selection of base graph for the LDPC code

In short, for a given lifting size Z, each "1" in the base matrix is replaced by the Z x Z identify matrix circularly shifted by the corresponding shift coefficient and each "0" in the base matrix is replaced by the Z x Z all-zero matrix. Hence, a relatively large number of parity-check matrices can be generated to support multiple payload sizes while maintaining the general structure of the LDPC code. To support payload sizes that are not a native payload size of one of the 51 defined parity check matrices, known filler bits can be appended to the code block before encoding. Since the NR LDPC codes are systematic codes, the filler bits can be removed before transmission.

2.7.3 Rate Matching and Hybrid-arq Functionality

The rate-matching and physical-layer hybrid-ARQ functionality serves two purposes,

1. to extract a suitable number of coded bits to match the resources assigned for transmission.
2. to generate different redundancy versions needed for the hybrid-ARQ protocol. The number of bits to transmit on the PDSCH or PUSCH depends on a wide range of factors,

Not only the number of resource blocks and the number of OFDM symbols scheduled, but also on the amount of overlapping resource elements used for other purposes and such as reference signals, control channels, or system information. There is also a possibility to, in the downlink, define reserved resources as a tool to provide future compatibility, which affects the number of resource elements usable for the PDSCH.

Rate matching is performed separately for each code block. First, a fixed number of the systematic bits are punctured. The fraction of systematic bits punctured can be high, up to 1/3 of the systematic bits, depending on the code block size. The remaining coded bits are written into a circular buffer, starting with the non-punctured systematic bits, and continuing with parity bits as illustrated in Fig. 2.10. The selection of the bits to transmit is based on reading the required number of bits from the circular buffer where the exact set of bits to transmit depends on the redundancy version (RV) corresponding to different starting positions in the circular buffer. Hence, by selecting different redundancy versions, different sets of coded bits representing the same set of information bits can be generated, which is used when implementing hybrid-ARQ with incremental redundancy. The starting points in the circular buffer are defined such that both RV0 and RV3 are self-decodable, that is, includes the systematic bits under typical scenarios. This is also the reason RV3 is located after "nine o'clock" in Fig. 2.10 as this allows more of the systematic bits to be included in the transmission.

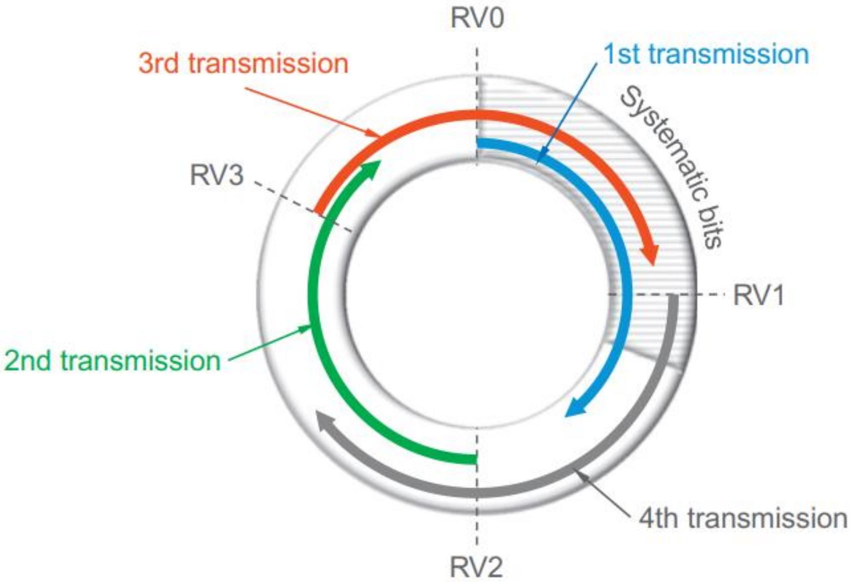


Figure 2.10 Circular buffer for incremental redundancy

In the receiver, soft combining is an important part of the hybrid-ARQ functionality. The soft values representing the received coded bits are buffered and, if a retransmission occurs, decoding is performed using the buffered bits combined with the retransmitted coded bits. In addition to a gain in accumulated received Eb/N0, with different coded bits in different transmission attempts, additional parity bits are obtained and the resulting code rate after soft combining is lower with a corresponding coding gain obtained. Soft combining requires a buffer in the receiver. Typically, a high probability of successful transmission on the first attempt is targeted and hence the soft buffer remains unused most of the time. Since the soft buffer size is large for the largest transport block sizes, requiring the receiver to buffer all soft bits even for the largest transport block sizes is suboptimal from a cost

performance trade-off perspective. Hence, limited-buffer rate-matching is supported as illustrated in Fig. 2.11.
In principle, only bits the device can buffer are kept in the circular buffer, that is, the size of the circular buffer is determined based on the receiver's soft buffering capability. For the downlink, the device is not required to buffer more soft bits than corresponding to the largest transport block size coded at rate 2/3. Note that this only limits the soft buffer capacity for the highest transport block sizes, that is, the highest data rates. For smaller transport block sizes, the device can buffer all soft bits down to the mother code rate. For the uplink, full-buffer rate matching, where all soft bits are buffered irrespective of the transport block size, is supported given sufficient gNB memory. Limited-buffer rate matching using the same principles as for the downlink can be configured using RRC signaling. The final step of the rate-matching functionality is to interleave the bits using a block interleaver and to collect the bits from each code block. The bits from the circular buffer are written row-by-row into a block interleaver and read out column-by-column. The number of rows in the interleaver is given by the modulation order and hence the bits in one column correspond to one modulation symbol3 (Fig. 2.12). This results in the systematic bits spread across the modulation symbols, which improves performance. Bit collection concatenates the bits for each code block. 2.9

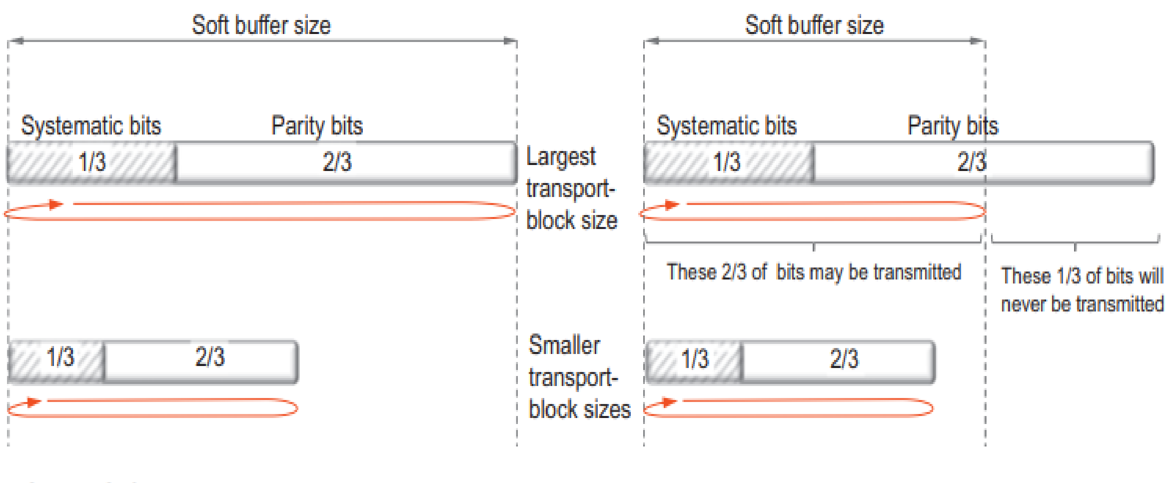


Figure 2.11 Limited-buffer rate matching.

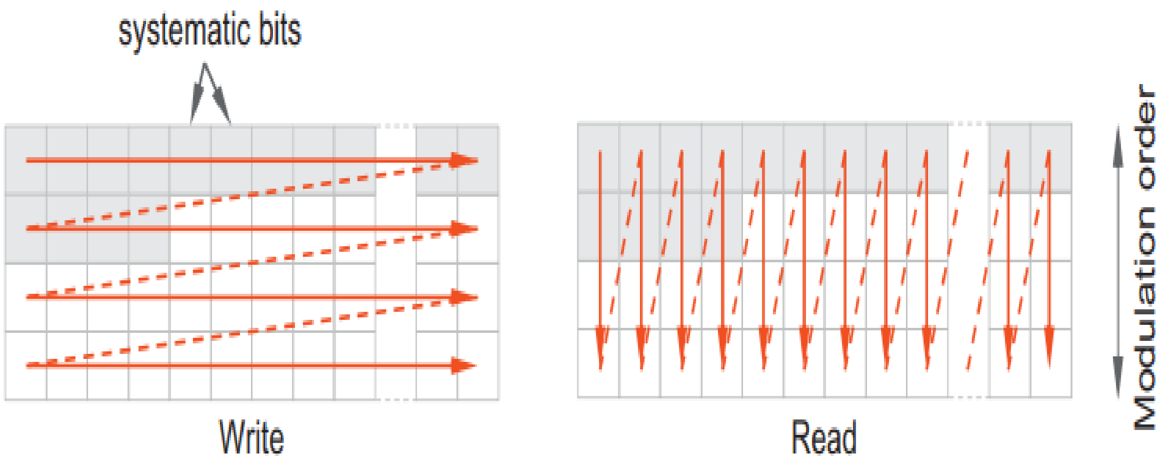


Figure 2.12 Bit interleaver (16QAM assumed in this example).

2.7.4 SCRAMBLING

Scrambling is applied to the block of coded bits delivered by the hybrid-ARQ functionality by multiplying the sequence of coded bits with a bit-level scrambling sequence. Without scrambling, the channel decoder at the receiver could, at least in principle, be equally matched to an interfering signal as to the target signal, thus being unable to properly suppress the interference. By applying different scrambling sequences for neighbouring cells in the downlink or for different devices in the uplink, the interfering signal(s) after descrambling is (are) randomized, ensuring full utilization of the processing gain provided by the channel code.

The scrambling sequence in both downlink (PDSCH) and uplink (PUSCH) depend on the identity of the device, that is, the C-RNTI, and a data scrambling identity configured in each device. If no data scrambling identity is configured, the physical layer cell identity is used as a default value to ensure that neighbouring devices, both in the same cell and between cells, use different scrambling sequences. Furthermore, in the case of two transport blocks being transmitted in the downlink to support more than four layers, different scrambling sequences are used for the two transport blocks.

2.7.5 MODULATION

The modulation step transforms the block of scrambled bits to a corresponding block of complex modulation symbols. The modulation schemes supported include QPSK, 16QAM, 64QAM, and 256QAM in both uplink and downlink. In addition, for the uplink $\pi/2$-BPSK is supported in the case the DFT-precoding is used, motivated by a reduced cubic metric [60] and hence improved power amplifier efficiency, for coverage limited scenarios. Note that $\pi/2$- BPSK is neither supported nor useful in the absence of DFT-precoding as the cubic metric in this case is dominated by the OFDM waveform.

2.7.6 LAYER MAPPING

The purpose of the layer-mapping step is to distribute the modulation symbols across the different transmission layers. This is done in an analogous way as for LTE; every nth symbol is mapped to the nth layer. One coded transport block can be mapped on up to four layers. In the case of five to eight layers, supported in the downlink only, a second transport block is mapped to layers five to eight following the same principle as for the first transport block.

Multi-layer transmission is only supported in combination with OFDM, the baseline waveform in NR. With DFT-precoding in the uplink, only a single transmission layer is supported. This is motivated both by the receiver complexity, which in the case of multi-layer transmission would be significantly higher with a DFT-precoder than without, and the use case originally motivating the additional support of DFT-precoding, namely handling of coverage-limited scenarios.

In such a scenario, the received signal-to-noise ratio is too low for efficient usage of spatial multiplexing and there is no need to support spatial multiplexing to a single device.

2.7.7 MULTI-ANTENNA PRECODING

The purpose of multi-antenna precoding is to map the different transmission layers to a set of antenna ports using a precoder matrix. In NR, the precoding and multi-antenna operation differs between downlink and uplink and the codebook-based precoding step is, except for CSI reporting, only visible in the uplink direction.

2.7.8 DOWNLINK PRECODING

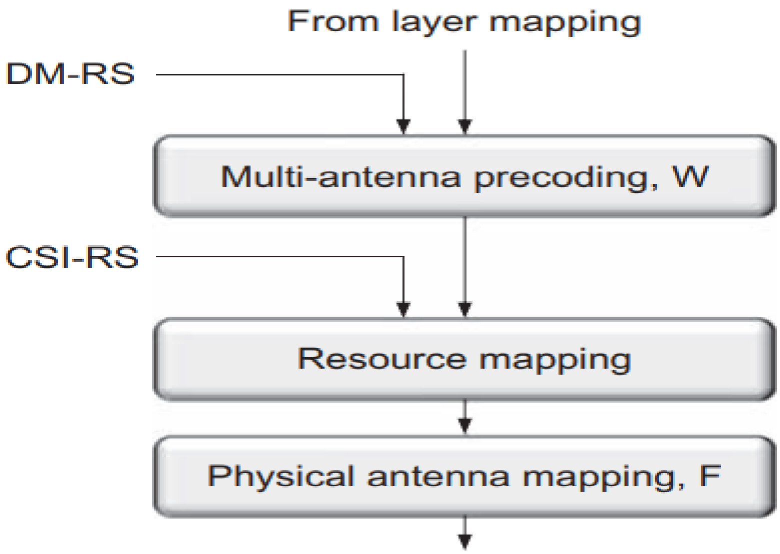


Figure 2.13 Downlink precoding.

In the downlink, the demodulation reference signal (DMRS) used for channel estimation is subject to the same precoding as the PDSCH (Fig. 2.13). Thus, the precoding is not explicitly visible to the receiver but is seen as part of the overall channel. This is like the receiver-transparent spatial filtering in the context of CSI-RS and SRS in. In essence, in terms of actual downlink transmission, any multi-antenna precoding can be seen as part of such, to the device, transparent spatial filtering.

However, for the purpose of CSI reporting, the device may assume that a specific precoding matrix W is applied at the network side. The device is then assuming that the precoder maps the signal to the antenna ports of the CSI-RS used for the measurements on which the reporting was done. The network is still free to use whatever precoder it finds advantageous for data transmission. To handle receiver-side beamforming, or in general multiple reception antennas with different spatial characteristics, QCL relations between a DM-RS port group, which is the antenna ports used for PDSCH transmission, and the antenna ports used for CSI-RS or SS block transmission can be configured. The Transmission Configuration Index (TCI) provided as part of the scheduling assignment indicates the QCL relations to use, or in other words, which reception beam to use.

Demodulation reference signals are, transmitted in the scheduled resource blocks and it is from those reference signals that the device can estimate the channel, including any precoding W and spatial filtering F applied for PDSCH. In principle, knowledge about the correlation between reference signal transmissions, both in terms of correlation introduced by the radio channel itself and correlation in the use of precoder, is useful to know and can be exploited by the device to improve the channel estimation accuracy.

In the time domain, the device is not allowed to make any assumptions on the reference signals being correlated between PDSCH scheduling occasions. This is necessary to allow full flexibility in terms of beamforming and spatial processing as part of the scheduling process.

In the frequency domain, the device can be given some guidance on the correlation. This is expressed in the form of physical resource-block groups (PRGs). Over the frequency span of one PRG, the device may assume the downlink precoder remains the same and may exploit this in the channel-estimation process, while the device may not make any assumptions in this respect between PRGs. From this it can be concluded that there is a trade-off between the precoding flexibility and the channel-estimation performance—a large PRG size can improve the channel-estimation accuracy at the cost of precoding flexibility and vice versa.

Hence, the gNB may indicate the PRG size to the device where the possible PRG sizes are two resource blocks, four resource blocks, or the scheduled bandwidth as shown in the bottom of Fig. 9.10. A single value may be configured, in which case this value is used for the PDSCH transmissions. It is also possible to dynamically, through the DCI, indicate the PRG size used. In addition, the device can be configured to assume that the PRG size equals the scheduled bandwidth.

2.7.9 RESOURCE MAPPING

The resource-block mapping takes the modulation symbols to be transmitted on each antenna port and maps them to the set of available resource elements in the set of resource blocks assigned by the MAC scheduler for the transmission. As, a resource block is 12 subcarriers wide and typically multiple OFDM symbols, and resource blocks, are used for the transmission. The set of timefrequency resources used for transmission is determined by the scheduler. However, some or all of the resource elements within the scheduled resource blocks may not be available for the transport-channel transmission as they are used for:

i. Demodulation reference signals (potentially including reference signals for other coscheduled devices in the case of multi-user MIMO)
ii. Other types of reference signals such as CSI-RS and SRS
iii. Downlink L1/L2 control signaling.
iv. Synchronization signals and system information.
v. Downlink reserved resources to provide forward.

The time frequency resources to be used for transmission are signaled by the scheduler as a set of virtual resource blocks and a set of OFDM symbols. To these scheduled resources, the modulation symbols are mapped to resource elements in a frequency-first, time-second manner. The frequency-first, time-second mapping is chosen to achieve low latency and allows both the transmitter and receiver to process the data “on the fly”. For high data rates, there are multiple code blocks in each OFDM symbol, and the device can decode those received in one symbol while receiving the next OFDM symbol. Similarly, assembling an OFDM symbol can take place while transmitting the previous symbols, thereby enabling a pipelined implementation. This would not be possible in the case of a time-first mapping as the complete slot needs to be prepared before the transmission can start.

The virtual resource blocks containing the modulation symbols are mapped to physical resource blocks in the bandwidth part used for transmission. Depending on the bandwidth part used for transmission, the carrier resource blocks can be determined and the exact frequency location on the carrier determined (Fig. 2.14 is an illustration). The reason for this, at first sight complicated mapping process with both virtual and physical resource blocks is to be able to handle a wide range of scenarios.

There are two methods for mapping virtual resource blocks to physical resource blocks, non-interleaved mapping (Fig. 2.14: top) and interleaved mapping (Fig. 2.14: bottom). The mapping scheme to use can be controlled on a dynamic basis using a bit in the DCI scheduling the transmission.

Non-interleaved mapping means that a virtual resource block in a bandwidth part maps directly to the physical resource block in the same bandwidth part. This is useful in cases when the network tries to allocate transmissions to physical resource with instantaneously favourable channel conditions. For example, the scheduler might have determined that physical resource blocks six to nine in Fig. 2.14 have favourable radio channel properties and are therefore preferred for transmission and a non-interleaved mapping is used.

Interleaved mapping maps virtual resource blocks to physical resource blocks using an interleaver spanning the whole bandwidth part and operating on pairs or quadruplets of resource blocks. A block interleaver with two rows is used, with pairs/quadruplets of resource blocks written column-by-column and read out rowby-row. Whether to use pairs or quadruplets of resource blocks in the interleaving operation is configurable by higher-layer signalling.

The reason for interleaved resource-block mapping is to achieve frequency diversity, the benefits of which can be motivated separately for small and large resource allocations.

For small allocations, for example voice services, channel-dependent scheduling may not be motivated from an overhead perspective due to the amount of feedback signaling required or may not be possible due to channel variations not being possible to track for a rapidly moving device. Frequency diversity by distributing the transmission in the frequency domain is in such cases an alternative way to exploit channel variations. Although frequency diversity could be obtained by using resource allocation type 0, this resource allocation scheme implies a relatively large control signaling overhead compared to the data payload transmitted as well as limited possibilities to signal very small allocations. Instead, by using the more compact resource allocation type 1, which is only capable of signaling contiguous resource allocations, combined with an interleaved virtual to physical resource block mapping, frequency diversity can be achieved with a small relative overhead. This is very similar to the distributed resource mapping in LTE. Since resource allocation type 0 can provide a high degree of flexibility in the resource allocation, interleaved mapping is supported for resource allocation type 1 only.

For larger allocations, possibly spanning the whole bandwidth part, frequency diversity can still be advantageous. In the case of a large transport block, that is, at very high data rates, the coded data are split into multiple code blocks as discussed in

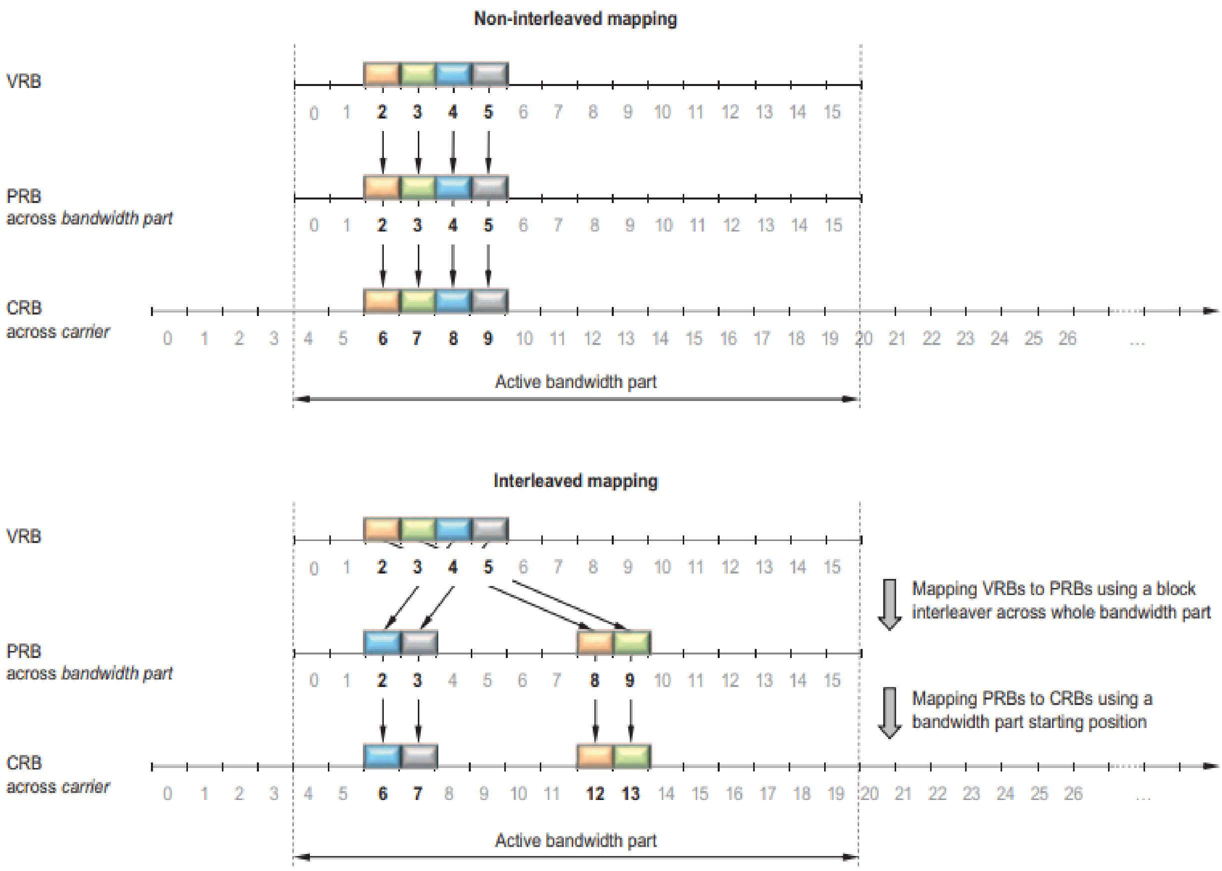


Figure 2.14 Mapping from virtual to physical to carrier resource blocks.

Section 2.7.2. Mapping the coded data directly to physical resource blocks in a frequency-first manner (remember, frequency-first mapping is beneficial from an overall latency perspective) would result in each code block occupying only a fairly small number of contiguous physical resource blocks. Hence, if the channel quality varies across the frequency range used for transmission, some code blocks may suffer worse quality than other code blocks, possibly resulting in the overall transport block failing to decode despite almost all code blocks being correctly decoded. The quality variations across the frequency range may occur even if the radio channel is flat due to imperfections in RF components. If an interleaved resource-block mapping is used, one code block occupying a contiguous set of virtual resource blocks would be distributed in the frequency domain across multiple, widely separated physical resource blocks, similarly to what is the case for the small allocations discussed in the previous paragraph. The result of the interleaved VRB-to-PRB mapping is a quality-averaging effect across the code blocks, resulting in a higher likelihood of correctly decoding very large transport blocks. This aspect of resource block mapping was not present in LTE, partially because the data rates were not as high as in NR, partly because the code blocks in LTE are interleaved.

The discussion above holds in general and for the downlink. In the uplink, release 15 only specifies RF requirements for contiguous allocations and therefore interleaved mapping is only supported for downlink transmissions. To obtain frequency diversity also in the uplink, frequency hopping can be used where the data in the first set of OFDM symbols in the slot are transmitted on the resource block as indicated by the scheduling grant. In the remaining OFDM symbols, data are transmitted on a different set of resource blocks given by a configurable offset from the first set. Uplink frequency hopping can be dynamically controlled using a bit in the DCI scheduling the transmission.

2.7.10 REFERENCE SIGNALS

Reference signals are predefined signals occupying specific resource elements within the downlink time frequency grid. The NR specification includes several types of reference signals transmitted in different ways and intended to be used for different purposes by a receiving device.

Unlike LTE, which relies heavily on always-on, cell-specific reference signals in the downlink for coherent demodulation, channel quality estimation for CSI reporting, and general time frequency tracking, NR uses different downlink reference signals for different purposes. This allows for optimizing each of the reference signals for their specific purpose. It is also in line with the overall principle of ultra-lean transmission as the different reference signals can be transmitted only when needed. Later release of LTE took some steps in this direction, but NR can exploit this to a much larger degree as there are no legacy NR devices to cater for. The NR reference signals include:-
:

i. Demodulation reference signals (DM-RS) for PDSCH are intended for channel estimation at the device as part of coherent demodulation. They are present only in the resource blocks used for PDSCH transmission. Similarly, the DM-RS for PUSCH allows the gNB to coherently demodulate the PUSCH.
ii. Phase-tracking reference signals (PT-RS) can be seen as an extension to DMRS for PDSCH/PUSCH and are intended for phase-noise compensation. The PT-RS is denser in time but sparser in frequency than the DM-RS, and, if configured, occurs only in combination with DM-RS.
iii. CSI reference signals (CSI-RS) are downlink reference signals intended to be used by devices to acquire downlink channel-state information (CSI). Specific instances of CSI reference signals can be configured for time/frequency tracking and mobility measurements.
iv. Tracking reference signals (TRS) are sparse reference signals intended to assist the device in time and frequency tracking. A specific CSI-RS configuration serves the purpose of a TRS.

v. Sounding reference signals (SRS) are uplink reference signals transmitted by the devices and used for uplink channel-state estimation at the base stations.

In the following, the demodulation reference signals intended for coherent demodulation of PDSCH and PUSCH are described in more detail, starting with the reference signal structure used for OFDM. The same DM-RS structure is used for both downlink and uplink in the case of OFDM. For DFT-spread OFDM in the uplink, a reference signal based on Zadoff Chu sequences as in LTE is used to improve the power-amplifier efficiency but supporting contiguous allocations and single-layer transmission only as discussed in a later section. Finally, a discussion on the phase-tracking reference signal is provided.

2.8 DEMODULATION REFERENCE SIGNALS FOR OFDM-BASED DOWNLINK

The DM-RS in NR provides quite some flexibility to cater for different deployment scenarios and use cases: a front-loaded design to enable low latency, support for up to 12 orthogonal antenna ports for MIMO, transmissions durations from 2 to 14 symbols, and up to four reference-signal instances per slot to support very high-speed scenarios.

To achieve low latency, it is beneficial to locate the demodulation reference signals early in the transmission, sometimes known as front-loaded reference signals. This allows the receiver to obtain a channel estimate early and, once the channel estimate is obtained, process the received symbols on the fly without having to buffer a complete slot prior to data processing. This is essentially the same motivation as for the frequency-first mapping of data to the resource elements.

Two main time-domain structures are supported, differencing in the location of the first DM-RS symbol:

i. Mapping type A, where the first DM-RS is located in symbol 2 or 3 of the slot and the DM-RS is mapped relative to the start of the slot boundary, regardless of where in the slot the actual data transmission starts. This mapping type is primarily intended for the case where the data occupy (most of) a slot. The reason for symbol 2 or 3 in the downlink is to locate the first DM-RS occasion after a CORESET located at the beginning of a slot.
ii. Mapping type B, where the first DM-RS is located in the first symbol of the data allocation, that is, the DM-RS location is not given relative to the slot boundary but rather relative to where the data are located. This mapping is originally motivated by transmissions over a small fraction of the slot to support very low latency and other transmissions that benefit from not waiting until a slot boundary starts but can be used regardless of the transmission duration. The mapping type for PDSCH transmission can be dynamically signaled as part of the DCI, while for the PUSCH the mapping type is semi statically configured.

Although front-loaded reference signals are beneficial from a latency perspective, they may not be sufficiently dense in the time domain in the case of rapid channel variations. To support high-speed scenarios, it is possible to configure up to three additional DM-RS occasions in a slot. The channel estimator in the receiver can use these additional occasions for more accurate channel estimation, for example, to use interpolation between the occasions within a slot. It is not possible to interpolate between slots, or in general different transmission occasions, as different slots may be transmitted to different devices and/or in different beam directions. This is a difference compared to LTE, where interslot interpolation of the channel estimates is possible but also restricts the multi-antenna and beamforming flexibility in LTE compared to NR.

Multiple orthogonal reference signals can be created in each DM-RS occasion. The different reference signals are separated in the frequency and code domains, and, in the case of a double-symbol DM-RS, additionally in the time domain. Two different types of demodulation reference signals can be configured, type 1 and type 2, differing in the mapping in the frequency domain and the maximum number of orthogonal reference signals. Type 1 can provide up to four orthogonal signals using a single-symbol DM-RS and up to eight orthogonal reference signals using a double-symbol DM-RS. The corresponding numbers for type 2 are six and twelve. The reference signal types (1 or 2) should not be confused with the mapping types (A or B); different mapping types can be combined with different reference signal types.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 INTRODUCTION

The methodology for "5G NR Downlink transport channel modelling using MATLAB" involves several steps. The first step is to study the 5G NR standard and understand the different types of downlink transport channels used in the downlink directions.

The next step is to identify the parameters that need to be considered in the modelling process. These parameters include channel bandwidth, modulation scheme, channel coding rate, and channel characteristics.

The third step is to use MATLAB to model the downlink transport channel. This involves using functions provided by the MATLAB 5G Toolbox, such as nrDLSCH, nrPDSCH, and nrOFDMModulate, to simulate the transmission and reception of data over the downlink transport channel. The model should also include the effects of channel resource mapping (nrResourceGrid), channel estimation (nrChannelEstimate) , and channel equalization (nrEqualizeMMSE)

The fourth step is to validate the model by comparing the simulation results with the expected results based on the 5G NR standard. This involves analysing metrics such as error rate, throughput, and spectral efficiency.

The final step is to use the validated model to study the performance of the 5G NR downlink transport channel under different scenarios. This includes varying the transport block size and modulation schemes and studying the impact on the performance of the downlink transport channel and compare the changes in the presence of channel improvement techniques like channel estimation, precoding and equalization.

Overall, the methodology involves a combination of theoretical analysis and practical simulation using MATLAB to develop a comprehensive understanding of the behaviour and performance of the 5G NR downlink transport channel.

3.2 MODEL-1

This is the model of 5G NR downlink transport channel with multiple hybrid automatic repeat-request (HARQ) processes using the downlink shared channel (DL-SCH) encoder and decoder 5G Toolbox System objects.

3.2.1 Introduction

Elements that are modelled in the context of a 5G downlink link are:

i. DL-SCH encoding and decoding
ii. Physical downlink shared channel (PDSCH) encoding and decoding

iii. HARQ management

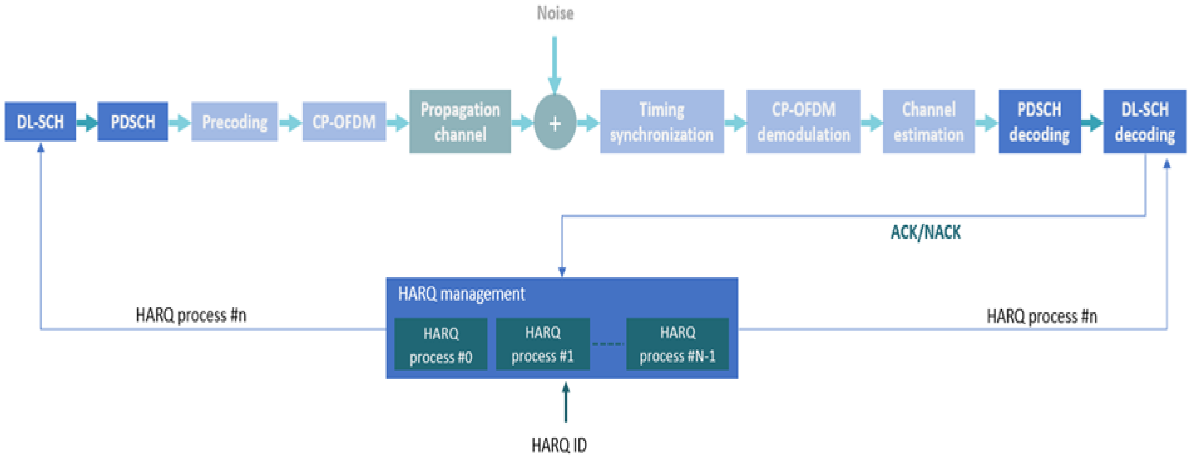


Figure 3.1. Link elements

I have also measured the block error rate (BLER) using an AWGN channel. Figure 3.2 shows all the link elements modelled followed by the BLER calculation

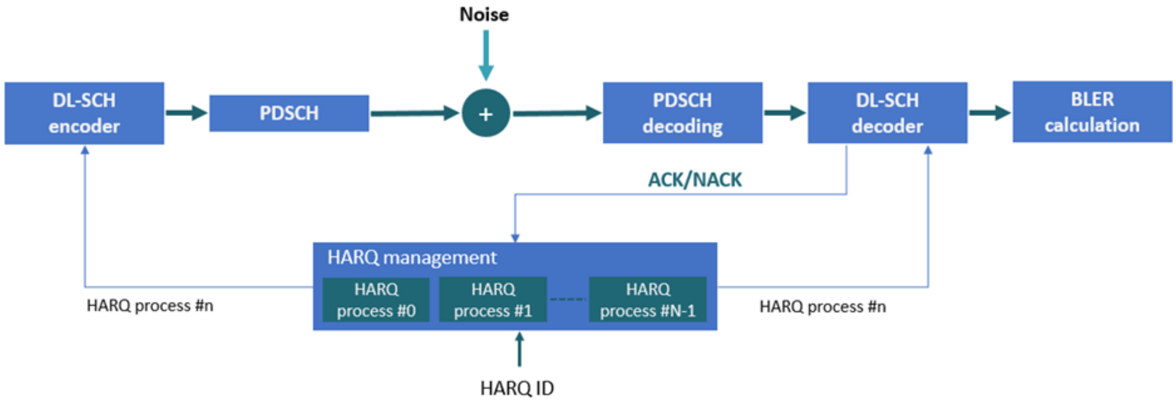


Figure 3.2. Link elements modelled

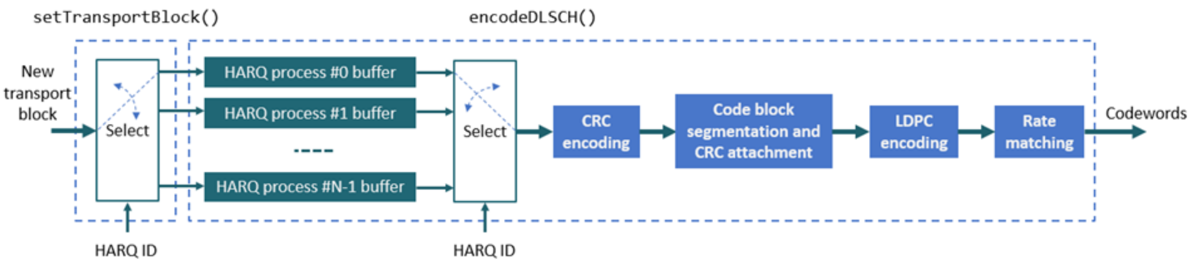


Figure 3.3. HARQ process buffer

This figure shows that the DL-SCH encoder uses internal buffers to store the transport blocks for each HARQ process and then selects the active HARQ process buffer content for the encoding. The DL-SCH decoder uses a similar buffering mechanism to store and select HARQ processes.

The DL-SCH encoder and decoder do not manage the HARQ processes internally. I have used the HARQ entity object, HARQEntity.m, for HARQ process management. This figure shows the structure of the HARQ entity object.

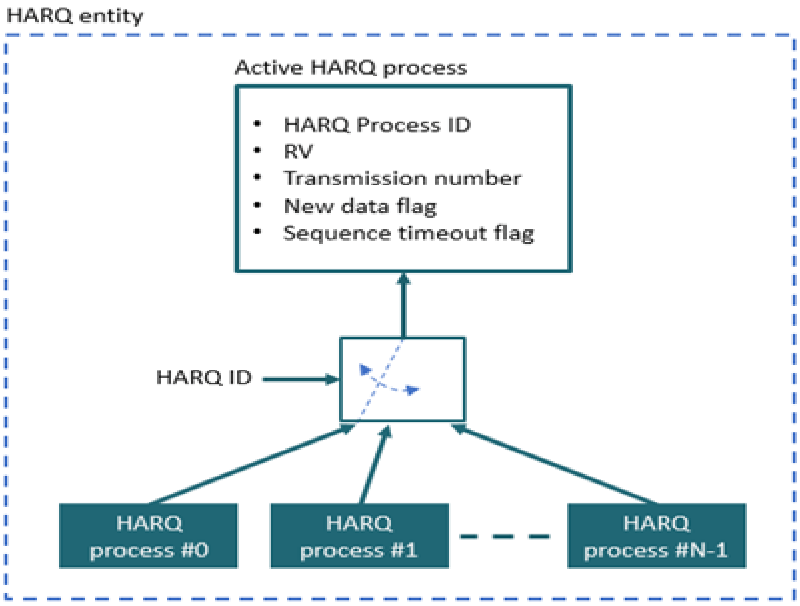


Figure 3.4. Harq Entity

3.2.2 Simulation Parameters

Specify the number of transport blocks to simulate and the signal to noise ratio (SNR).

```
noTransportBlocks = 100; % No. of Transport blocks
SNRdB = 7; % SNR in dB
```

Reset random number generator for reproducibility.

```
rng("default");
```

3.2.3 Dl-Sch Configurations

Specify the code rate, the number of HARQ processes, and the redundancy values (RVs) sequence. This sequence controls the redundancy version retransmissions in case of error.

```
% DL-SCH parameters
codeRate = 490/1024;
NHARQProcesses = 16; % Number of parallel HARQ processes to use
rvSeq = [0 2 3 0];
```

Create the DL-SCH encoder and decoder objects. To use multiple processes, set the *MultipleHARQProcesses* property to true for both objects. To enable retransmissions for multiple HARQ processes, the encoder buffers the input bits. The decoder needs a similar mechanism to enable soft combining of retransmissions for each HARQ process.

```
% Create DL-SCH encoder object
encodeDLSCH = nrDLSCH;
encodeDLSCH.MultipleHARQProcesses = true;
encodeDLSCH.TargetCodeRate = codeRate;


% Create DL-SCH decoder object
decodeDLSCH = nrDLSCHDecoder;
decodeDLSCH.MultipleHARQProcesses = true;
decodeDLSCH.TargetCodeRate = codeRate;
decodeDLSCH.LDPCDecodingAlgorithm = "Normalized min-sum";
decodeDLSCH.MaximumLDPCIterationCount = 6;
```

The DL-SCH encoder and decoder objects can model up to 16 HARQ processes. The encoder and decoder objects use the *HARQprocessID* property of the HARQ entity object to identify the active HARQ process when performing any of these operations.

i. Setting new transport block to transmit
ii. Encoding data
iii. Resetting soft buffers
iv. Decoding data

3.2.4 Carrier and PDSCH Configuration

Specify the carrier and PDSCH parameters. These parameters are used for PDSCH encoding and decoding and for calculating the transport block size.

Create a carrier object, specifying the subcarrier spacing (SCS) and the bandwidth (BW).

```
% Numerology
SCS = 15;           % SCS: 15, 30, 60, 120 or 240 (kHz)
NRB = 52;           % BW in number of RBs (52 RBs at 15 kHz SCS for 10 MHz BW)
carrier = nrCarrierConfig;
carrier.NSizeGrid = NRB;
carrier.SubcarrierSpacing = SCS;
carrier.CyclicPrefix = "Normal";  % "Normal" or "Extended"
```

Create a PDSCH configuration object. The PDSCH parameters determine the available bit capacity and the transport block size.

```
modulation = "16QAM";            % Modulation scheme
pdsch = nrPDSCHConfig;
pdsch.Modulation = modulation;
```

```
pdsch.PRBSet = 0:NRB-1;                 % Assume full band allocation
pdsch.NumLayers = 1;                    % Assume only one layer and one codeword
```

3.2.5 HARQ management

Create a HARQ entity object to manage the HARQ processes. For each HARQ processes, the object stores these elements:

i. HARQ ID number.
ii. RV.
iii. Transmission number, which indicates how many times a certain transport block has been transmitted.
iv. Flag to indicate whether new data is required. New data is required when a transport block is received successfully or if a sequence timeout has occurred (all RV transmissions have failed).
v. Flag to indicate whether a sequence timeout has occurred (all RV transmissions have failed).

```
harqEntity = HARQEntity(0:NHARQProcesses-1,rvSeq,pdsch.NumCodewords);
```

The HARQ entity is used to manage the buffers in the DL-SCH encoder and decoder.

3.2.6 BER simulation

Loop over a number of transport blocks. For each transport block:

i. Calculate the transport block size in number of bits.
ii. Generate new data block or reset buffers in the decoder.
iii. Apply DL-SCH encoding.
iv. Modulate bits to symbols.
v. Apply AWGN.
vi. Demodulate soft bits (symbols to soft bits).
vii. Decode the DL-SCH.
viii. Update the HARQ processes.

```
% Initialize loop variables
noiseVar = 1./(10.^(SNRdB/10)); % Noise variance
numBlkErr = 0;                  % Number of block errors
numRxBits = [];         % Number of successfully received bits per transmission
txedTrBlkSizes = [];    % Number of transmitted info bits per transmission
for nTrBlk = 1:noTransportBlocks      % A transport block or transmission time
                                      % interval (TTI) corresponds to one slot
    carrier.NSlot = carrier.NSlot+1;
```

3.2.6.1 Transport Block Size

Calculate the transport block size.

```
    % Generate PDSCH indices info, which is used to calculate the transport

    % block size

    [~,pdschInfo] = nrPDSCHIndices(carrier,pdsch);

    % Calculate transport block sizes
    Xoh_PDSCH = 0;

    trBlkSizes =

nrTBS(pdsch.Modulation,pdsch.NumLayers,numel(pdsch.PRBSet),pdschInfo.NREPerPRB
,codeRate,Xoh_PDSCH);
```

Because the PDSCH capacity in bits, *pdsch.G*, is dynamically determined, the actual code rate might not be exactly equal to the target code rate specified by the *TargetCodeRate* property of the *encodeDLSCH* object.

3.2.6.2 HARQ Processing (Buffer Management)

This section explains the buffer management in the encoder and decoder.

i. DL-SCH encoder buffers: Generate a new transport block if new data is required for the active HARQ process. Store the transport block in the corresponding buffer. If no new data is required, the buffered bits in the DL-SCH encoder are used for retransmission.
ii. DL-SCH decoder buffers: The soft buffers in the receiver store previously received versions of the same transport block. These buffers are cleared automatically upon successful reception (no CRC error). However, if the RV sequence ends without successful decoding, the buffers must be flushed manually by calling the *resetSoftBuffer* object function.

```
    % Get new transport blocks and flush decoder soft buffer, as required

    for cwIdx = 1:pdsch.NumCodewords

        if harqEntity.NewData(cwIdx)

            % Create and store a new transport block for transmission

            trBlk = randi([0 1],trBlkSizes(cwIdx),1);

       setTransportBlock(encodeDLSCH,trBlk,cwIdx-1,harqEntity.HARQProcessID);

            % If the previous RV sequence ends without successful decoding,
            % flush the soft buffer explicitly

            if harqEntity.SequenceTimeout(cwIdx)
                resetSoftBuffer(decodeDLSCH,cwIdx-1,harqEntity.HARQProcessID);

            end
        end
    end
```

3.2.6.3 DL-SCH Encoding

Encode the DL-SCH transport blocks.

```
   codedTrBlock   =   encodeDLSCH(pdsch.Modulation,pdsch.NumLayers,pdschInfo.G,
...harqEntity.RedundancyVersion,harqEntity.HARQProcessID);
```

3.2.6.4 PDSCH Encoding

Generate the PDSCH symbols.

```
    modOut = nrPDSCH(carrier,pdsch,codedTrBlock);
```

3.2.6.5 AWGN Channel

Add white Gaussian noise.

```
    rxSig = awgn(modOut,SNRdB);
```

3.2.6.6 PDSCH Demodulation

Soft demodulate the received symbols.

```
    rxLLR = nrPDSCHDecode(carrier,pdsch,rxSig,noiseVar);
```

3.2.6.7 DL-SCH Decoding

Apply DL-SCH decoding.

```
    decodeDLSCH.TransportBlockLength = trBlkSizes;

    [decbits,blkerr] = decodeDLSCH(rxLLR,pdsch.Modulation,pdsch.NumLayers, ...

        harqEntity.RedundancyVersion,harqEntity.HARQProcessID);
```

3.3 MODEL -2

3.3.1 Introduction

This appendix explains the downlink transport channel MATLAB code.

3.3.2 Simulation Parameters

Specify the signal-to-noise ratio (SNR), number of slots to simulate, perfect channel estimation flag, no. of slots and transport block size

```
SNRdB = 7;                     % SNR in dB

totalNoSlots = 20;              % Number of slots to simulate

perfectEstimation = false; % Perfect synchronization and channel estimation

rng("default");             % Set default random number generator for repeatability

noTransportBlocks = 100;     % No. of transport block
```

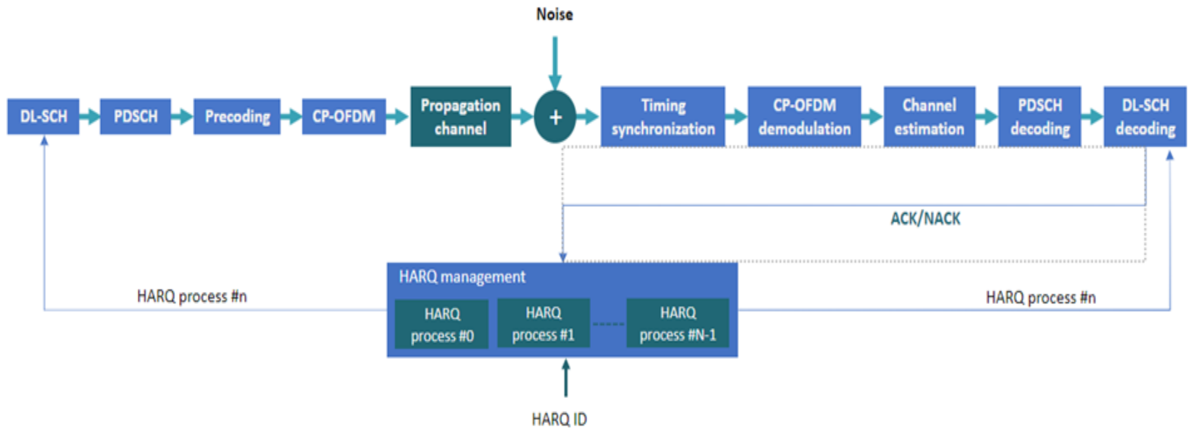


Figure 3.5 5G NR Downlink transport channel full processing chain

3.3.3 Carrier Configuration

This object controls the numerology, such as, the subcarrier spacing, bandwidth, and cyclic prefix (CP) length. This example uses the default set of properties.

```
carrier = nrCarrierConfig
```

3.3.4 PDSCH and DM-RS Configuration

Define the modulation scheme (16-QAM) and the number of layers. Allocate all resource blocks (RBs) to the PDSCH (full band allocation). You can also specify other time-allocation parameters and demodulation reference signal (DM-RS) settings in this object.

```
pdsch = nrPDSCHConfig;
pdsch.Modulation = "16QAM";
pdsch.NumLayers = 1;
pdsch.PRBSet = 0:carrier.NSizeGrid-1;     % Full band allocation
```

Set the DM-RS parameters. To improve channel estimation, add an additional DM-RS position.

```
pdsch.DMRS.DMRSAdditionalPosition = 1;
```

Set the DM-RS configuration type and the DM-RS length, which determines the number of orthogonal DM-RS sequences or DM-RS ports.

1. DMRSConfigurationType = 1 supports up to 4 DM-RS ports when DMRSLength = 1.
2. DMRSConfigurationType = 1 supports up to 8 DM-RS ports when DMRSLength = 2.
3. DMRSConfigurationType = 2 supports up to 6 DM-RS ports when DMRSLength = 1. This is designed for multi-user MIMO (MU-MIMO).
4. DMRSConfigurationType = 2 supports up to 12 DM-RS ports when DMRSLength = 2. This is designed for MU-MIMO.

The maximum number of layers must be less than or equal to the number of DM-RS ports.

```
pdsch.DMRS.DMRSConfigurationType = 1;
```

```
pdsch.DMRS.DMRSLength = 2;
pdsch.DMRS                            % Display DM-RS properties
```

3.3.5 DL-SCH Configuration

Specify the code rate, the number of HARQ processes, and the redundancy version (RV) sequence values. This sequence controls the RV retransmissions in case of error. To disable HARQ retransmissions, you can set rvSeq to a constant value.

```
NHARQProcesses = 16;     % Number of parallel HARQ processes
rvSeq = [0 2 3 1];
codeRate= 490/1024;
```

Create the DL-SCH encoder and decoder objects. To use multiple processes, set the MultipleHARQProcesses property to true for both objects. You do not need to specify the number of HARQ processes. The DL-SCH encoder and decoder objects can model up to 16 HARQ processes. To identify the active HARQ process when performing operations with the DL-SCH encoder and decoder objects, use the HARQprocessID property of the HARQ entity object.

```
% Create DL-SCH encoder object
encodeDLSCH = nrDLSCH;
encodeDLSCH.MultipleHARQProcesses = true;
encodeDLSCH.TargetCodeRate = codeRate;
% Create DLSCH decoder object
decodeDLSCH = nrDLSCHDecoder;
decodeDLSCH.MultipleHARQProcesses = true;
decodeDLSCH.TargetCodeRate = codeRate;
decodeDLSCH.LDPCDecodingAlgorithm = “Normalized min-sum”;
decodeDLSCH.MaximumLDPCIterationCount = 6;
```

3.3.6 HARQ Management

Create a HARQ entity object to manage the HARQ processes and the DL-SCH encoder and decoder buffers. For each HARQ process, a HARQ entity stores these elements:

i. HARQ ID number.
ii. RV.
iii. Transmission number, which indicates how many times a certain transport block has been transmitted.
iv. Flag to indicate whether new data is required. New data is required when a transport block is received successfully or if a sequence timeout has occurred (all RV transmissions have failed).

v. Flag to indicate whether a sequence timeout has occurred (all RV transmissions have failed).

```
harqEntity = HARQEntity(0:NHARQProcesses-1,rvSeq,pdsch.NumCodewords);
```

3.3.7 Channel Configuration

Specify the number of transmit and receive antennas.

```
nTxAnts = 8;

nRxAnts = 8;

% Check that the number of layers is valid for the number of antennas
if pdsch.NumLayers > min(nTxAnts,nRxAnts)

    error("The number of layers ("+string(pdsch.NumLayers)+") must be smaller than min(nTxAnts,nRxAnts) ("+string(min(nTxAnts,nRxAnts))+")")

end
```

Create a channel object.

```
Channel = nrTDLChannel;

channel.DelayProfile = "TDL-C";

channel.NumTransmitAntennas = nTxAnts;

channel.NumReceiveAntennas = nRxAnts;
```

Set the channel sample rate to that of the OFDM signal. To obtain the sampling rate of the OFDM signal, use the **nrOFDMInfo** function.

```
ofdmInfo = nrOFDMInfo(carrier);

channel.SampleRate = ofdmInfo.SampleRate;
```

3.3.8 Transmission and Reception

Set up a loop to simulate the transmission and reception of slots. Create a **comm.ConstellationDiagram** to display the constellation of the equalized signal.

```
constPlot = comm.ConstellationDiagram;                                    % Constellation diagram object

constPlot.ReferenceConstellation = getConstellationRefPoints(pdsch.Modulation); % Reference constellation values

constPlot.EnableMeasurements = 1;                                         % Enable EVM measurements

% Initial timing offset
offset = 0;

estChannelGrid = getInitialChannelEstimate(channel,carrier);
newPrecodingWeight = getPrecodingMatrix(pdsch.PRBSet,pdsch.NumLayers,estChannelGrid);
```

3.3.9 BER Simulation

```
% Initialize loop variables
noiseVar = 1./(10.^(SNRdB/10)); % Noise variance
numBlkErr = 0;                  % Number of block errors
numRxBits = [];       % Number of successfully received bits per transmission
txedTrBlkSizes = [];       % Number of transmitted info bits per transmission
for nTrBlk = 1:noTransportBlocks
% A transport block or transmission time interval (TTI) corresponds to 1 slot
    carrier.NSlot = carrier.NSlot+1;
```

3.3.10 Calculate Transport Block Size

The transport block size is the number of bits to send to the channel coding stages. This value depends on the capacity of the PDSCH. To calculate the transport block size, use the nrTBS function.

```
    % Generate PDSCH indices info, which is needed to calculate the transport
    % block size
    [pdschIndices,pdschInfo] = nrPDSCHIndices(carrier,pdsch);
    % Calculate transport block sizes
    Xoh_PDSCH = 0;
    trBlkSizes                                                              =
nrTBS(pdsch.Modulation,pdsch.NumLayers,numel(pdsch.PRBSet),pdschInfo.NREPerPRB
,codeRate,Xoh_PDSCH);
```

3.3.11 HARQ Processing (Buffer Management)

1. DL-SCH encoder buffers: Generate a new transport block if new data is required for the active HARQ process. Store the transport block in the corresponding buffer. If new data is not required, the DL-SCH encoder uses its buffered bits for retransmission.
2. DL-SCH decoder buffers: The soft buffers in the receiver store previously received versions of the same codeword. These buffers are cleared automatically upon successful reception (no CRC error). However, if the RV sequence ends without successful decoding, flush the buffers manually by using the `resetSoftBuffer` object function.

```
    % Get new transport blocks and flush decoder soft buffer, as required
    for cwIdx = 1:pdsch.NumCodewords
        if harqEntity.NewData(cwIdx)
            % Create and store a new transport block for transmission
            trBlk = randi([0 1],trBlkSizes(cwIdx),1);
```

```
            setTransportBlock(encodeDLSCH,trBlk,cwIdx-
1,harqEntity.HARQProcessID);

            % If the previous RV sequence ends without successful
            % decoding, flush the soft buffer

            if harqEntity.SequenceTimeout(cwIdx)

                resetSoftBuffer(decodeDLSCH,cwIdx-1,harqEntity.HARQProcessID);

            end

        end

    end
```

3.3.11.1 DL-SCH Encoding

Encode the transport blocks.

```
    codedTrBlock                                                            =
encodeDLSCH(pdsch.Modulation,pdsch.NumLayers,pdschInfo.G,harqEntity.Redundancy
Version,harqEntity.HARQProcessID);
```

3.3.11.2 PDSCH Modulation and MIMO Precoding

Generate PDSCH symbols from the coded transport blocks.

```
    pdschSymbols = nrPDSCH(carrier,pdsch,codedTrBlock);
```

Get the precoding weights. This example assumes channel knowledge for precoding

```
    precodingWeights = newPrecodingWeight;
```

Precode the PDSCH symbols.

```
    pdschSymbolsPrecoded = pdschSymbols*precodingWeights;
```

3.3.12 PDSCH DM-RS Generation

Generate DM-RS symbols and indices.

```
    dmrsSymbols = nrPDSCHDMRS(carrier,pdsch);

    dmrsIndices = nrPDSCHDMRSIndices(carrier,pdsch);
```

3.3.13 Mapping to Resource Grid

Generate an empty resource grid. This grid represents a slot.

```
    pdschGrid = nrResourceGrid(carrier,nTxAnts);

    [~,pdschAntIndices] = nrExtractResources(pdschIndices,pdschGrid);

    pdschGrid(pdschAntIndices) = pdschSymbolsPrecoded;
```

MIMO-precode and map the DM-RS symbols to the resource grid. Similar to the PDSCH indices, the DM-RS indices refer to layers. To convert these layer indices to antenna indices, we use the **nrExtractResources** function again.

```
% PDSCH DM-RS precoding and mapping
for p = 1:size(dmrsSymbols,2)
    [~,dmrsAntIndices] = nrExtractResources(dmrsIndices(:,p),pdschGrid);
    pdschGrid(dmrsAntIndices) = pdschGrid(dmrsAntIndices) + dmrsSymbols(:,p)*precodingWeights(p,:);
end
```

3.3.14 OFDM Modulation

OFDM-modulate the resource grid.

```
[txWaveform,waveformInfo] = nrOFDMModulate(carrier,pdschGrid);
```

3.3.15 Propagation Channel

The propagation channel generates *N* output samples for an input with *N* samples. However, the block of *N* output samples includes the channel filter transient (*K* samples).. *N*-*K* samples are not enough to decode a slot-worth of data. Part of the slot samples are in the channel filter delay line and are not flushed yet. To flush all relevant samples out of the channel filter, we padded the input signal with zeros.

```
chInfo = info(channel);
maxChDelay = ceil(max(chInfo.PathDelays*channel.SampleRate)) + chInfo.ChannelFilterDelay;
txWaveform = [txWaveform; zeros(maxChDelay,size(txWaveform,2))];
```

Send the signal through the channel and add noise.

```
[rxWaveform,pathGains,sampleTimes] = channel(txWaveform);
noise = generateAWGN(SNRdB,nRxAnts,waveformInfo.Nfft,size(rxWaveform));
rxWaveform = rxWaveform + noise;
```

3.3.16 Timing Synchronization

One can perform perfect or practical synchronization.

Perform perfect or practical timing estimation and synchronization.

```
if perfectEstimation
    % Get path filters for perfect timing estimation
    pathFilters = getPathFilters(channel);
    [offset,mag] = nrPerfectTimingEstimate(pathGains,pathFilters);
else
```

```
        [t,mag] =
nrTimingEstimate(carrier,rxWaveform,dmrsIndices,dmrsSymbols);

        offset = hSkipWeakTimingOffset(offset,t,mag);

    end

    rxWaveform = rxWaveform(1+offset:end,:);
```

3.3.17 OFDM Demodulation

OFDM-demodulate the synchronized signal.

```
    rxGrid = nrOFDMDemodulate(carrier,rxWaveform);
```

3.3.18 Channel Estimation

Channel estimation provides a representation of the channel effects per resource element (RE). The equalizer uses this information to compensate for the distortion introduced by the channel.

One can perform perfect or practical channel estimation.

This figure shows the reference points of the channel estimates in the downlink processing chain.

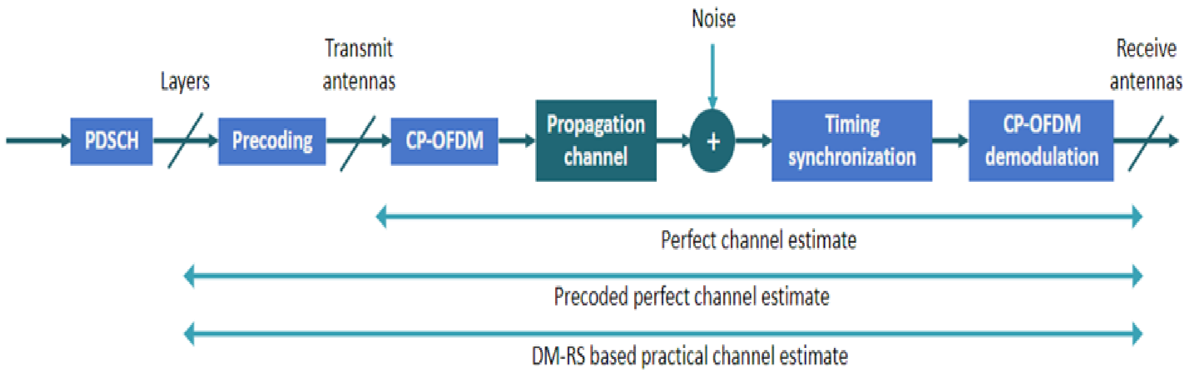


Figure 3.6 reference points of the channel estimates in the downlink processing chain.

Perform perfect or practical channel estimation.

```
    if perfectEstimation

 % Perform perfect channel estimation between transmit and receive antennas.

        estChGridAnts =
nrPerfectChannelEstimate(carrier,pathGains,pathFilters,offset,sampleTimes);

        % Get perfect noise estimate (from noise realization)
        noiseGrid = nrOFDMDemodulate(carrier,noise(1+offset:end ,:));

        noiseEst = var(noiseGrid(:));

        % Get precoding matrix for next slot
        newPrecodingWeight =
getPrecodingMatrix(pdsch.PRBSet,pdsch.NumLayers,estChGridAnts);
```

```
        % Apply precoding to estChGridAnts. The resulting estimate is for
        % the channel estimate between layers and receive antennas.

        estChGridLayers                                                                    =
precodeChannelEstimate(estChGridAnts,precodingWeights.');

    else

% Perform practical channel estimation between layers and receive antennas.

        [estChGridLayers,noiseEst]                                                         =
nrChannelEstimate(carrier,rxGrid,dmrsIndices,dmrsSymbols,'CDMLengths',pdsch.DM
RS.CDMLengths);

    % Remove precoding from estChannelGrid before precoding matrix calculation
        estChGridAnts                                                                      =
precodeChannelEstimate(estChGridLayers,conj(precodingWeights));

        % Get precoding matrix for next slot
        newPrecodingWeight                                                                 =
getPrecodingMatrix(pdsch.PRBSet,pdsch.NumLayers,estChGridAnts);

    end
```

Plot the channel estimate between the first layer and the first receive antenna.

```
    mesh(abs(estChGridLayers(:,:,1,1)));

    title('Channel Estimate');

    xlabel('OFDM Symbol');

    ylabel("Subcarrier");

    zlabel("Magnitude");
```

At this point, one can use the channel estimate to obtain the precoding matrix for transmission in the next slot

3.3.19 Equalization

The equalizer uses the channel estimate to compensate for the distortion introduced by the channel. Extract the PDSCH symbols from the received grid and associated channel estimates. The csi output has channel state information (CSI) for each of the equalized PDSCH symbols. The CSI is a measure of the channel conditions for each PDSCH symbol. Use the CSI to weight the decoded soft bits after PDSCH decoding, effectively increasing the importance of symbols experiencing better channel conditions.

```
     [pdschRx,pdschHest]                                                                   =
nrExtractResources(pdschIndices,rxGrid,estChGridLayers);

    [pdschEq,csi] = nrEqualizeMMSE(pdschRx,pdschHest,noiseEst);
```

Plot the constellation of the equalized symbols. The plot includes the constellation diagrams for all layers.

```
    constPlot.ChannelNames = "Layer "+(pdsch.NumLayers:-1:1);
    constPlot.ShowLegend = true;
  % Constellation for the first layer has a higher SNR than that for the
  % last layer. Flip the layers so that the constellations do not mask
  % each other.
    constPlot(fliplr(pdschEq));
```

3.3.20 PDSCH Decoding

Decode the equalized PDSCH symbols and obtain the soft bit codewords.

```
    [dlschLLRs,rxSymbols] = nrPDSCHDecode(carrier,pdsch,pdschEq,noiseEst);
```

Scale the soft bits or log-likelihood ratios (LLRs) by the CSI. This scaling applies a larger weight to the symbols in the REs with better channel conditions.

```
    % Scale LLRs by CSI
     csi = nrLayerDemap(csi);                          % CSI layer demapping
    for cwIdx = 1:pdsch.NumCodewords
    Qm = length(dlschLLRs{cwIdx})/length(rxSymbols{cwIdx}); % Bits per symbol
       csi{cwIdx} = repmat(csi{cwIdx}.',Qm,1); % Expand by each bit per symbol
        dlschLLRs{cwIdx} = dlschLLRs{cwIdx} .* csi{cwIdx}(:);   % Scale
    end
```

3.3.21 DL-SCH Decoding

Decode the LLRs and check for errors.

```
    decodeDLSCH.TransportBlockLength = trBlkSizes;
    [decbits,blkerr] = decodeDLSCH(dlschLLRs,pdsch.Modulation,pdsch.NumLayers,
...harqEntity.RedundancyVersion,harqEntity.HARQProcessID);
```

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Storing Result

Store the results to calculate the BLER.

```
% Store values to calculate throughput (only for active transport blocks)
    if(any(trBlkSizes ~= 0))
        numRxBits = [numRxBits trBlkSizes.*(1-blkerr)];
        txedTrBlkSizes = [txedTrBlkSizes trBlkSizes];
    end
    if blkerr
        numBlkErr = numBlkErr + 1;
    end
```

4.2 HARQ Process Update

Update the current HARQ process with the CRC error, and then advance to the next process. This step updates the information related to the active HARQ process in the HARQ entity.

```
    statusReport = updateAndAdvance(harqEntity,blkerr,trBlkSizes,pdschInfo.G);
```

Display information about the current decoding attempt.

```
    disp("Slot "+(nTrBlk)+". "+statusReport);
        end % for nTrBlk = 1:noTransportBlocks
```

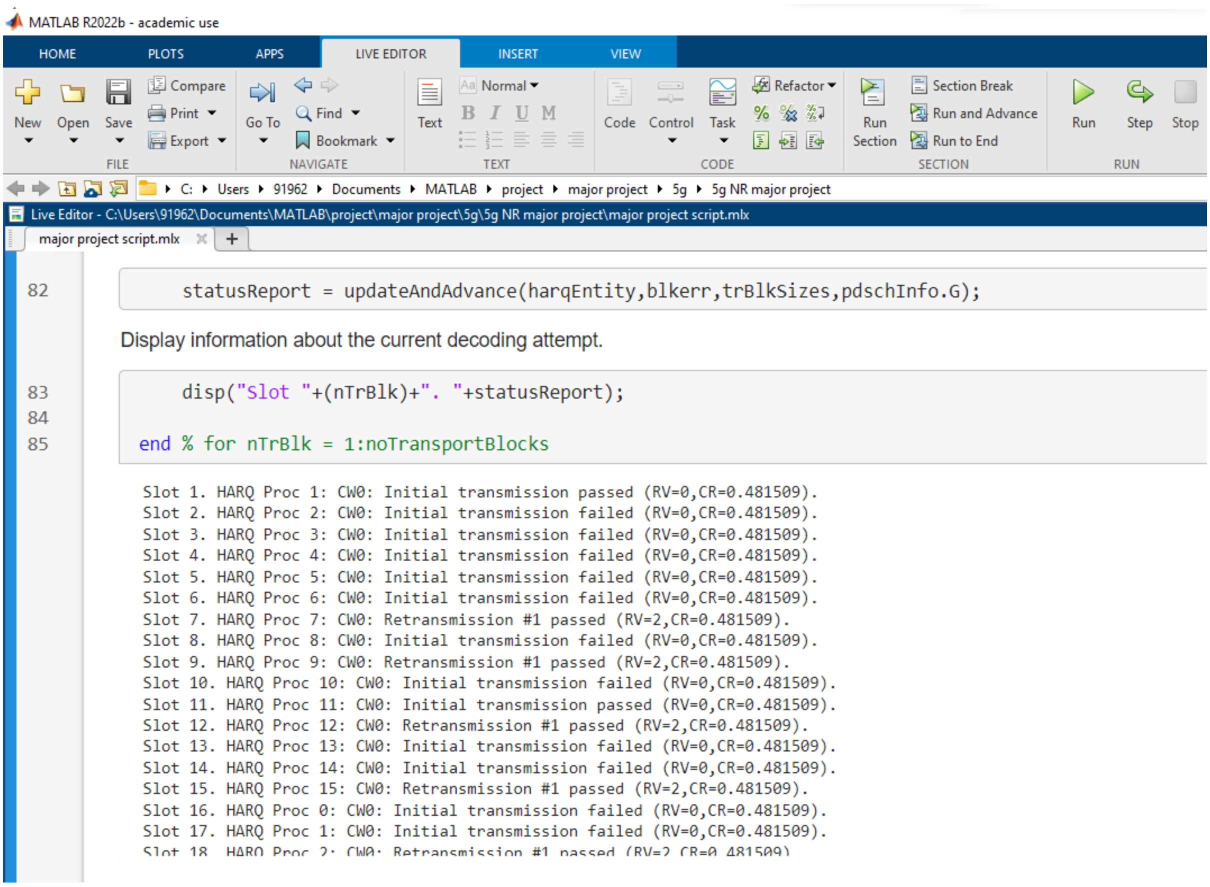


Figure 4.1 HARQ Process updates

4.3 PLOTS

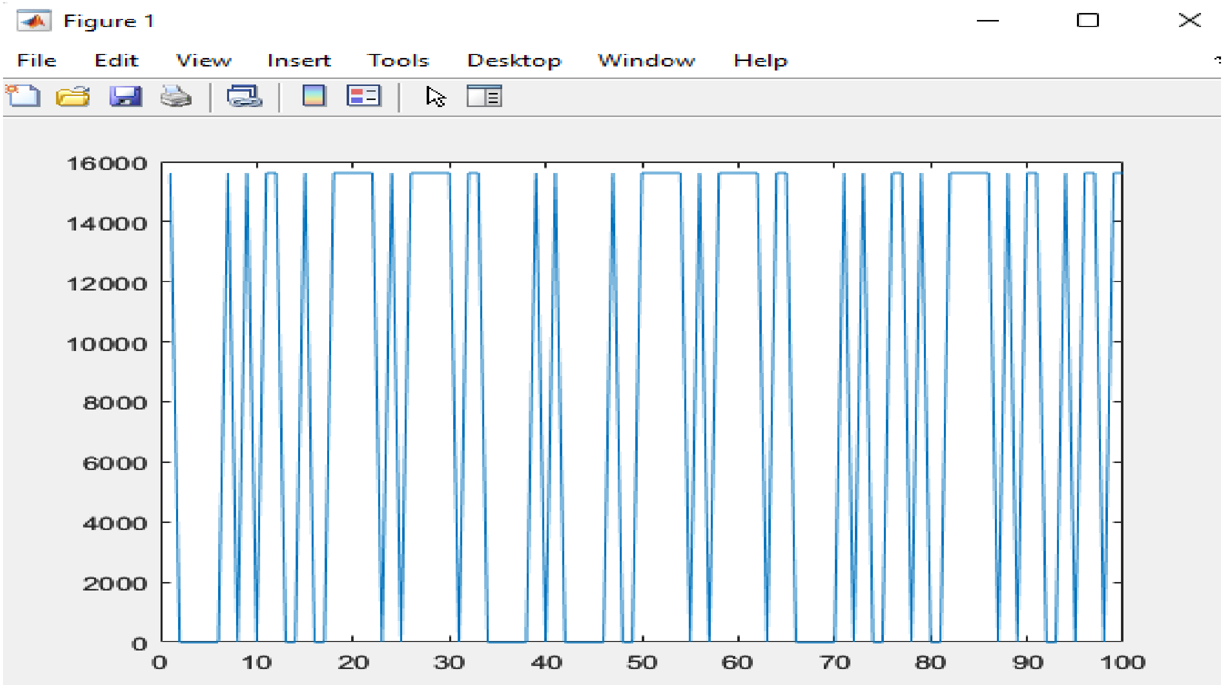


Figure 4.2 Plot of Number of successfully received bits per transmission

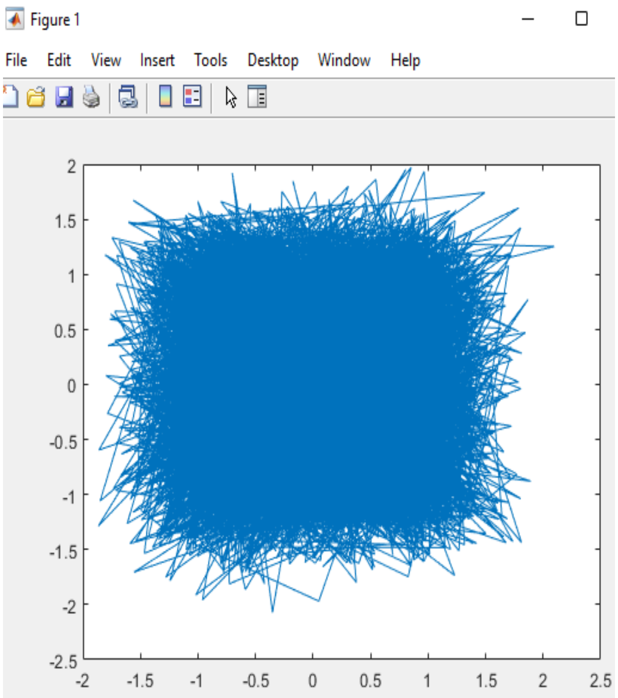


Figure 4.3. Plot of AWGN

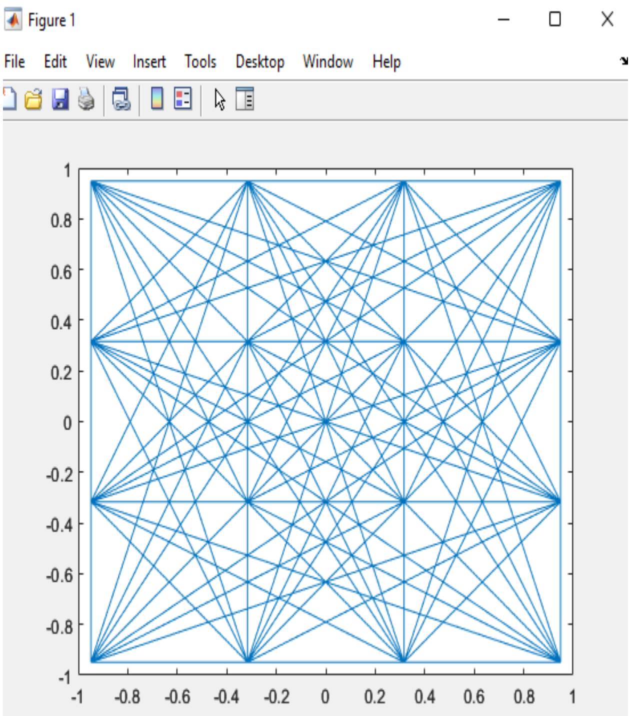


Figure 4.4. Plot of PDSCH symbols encoded

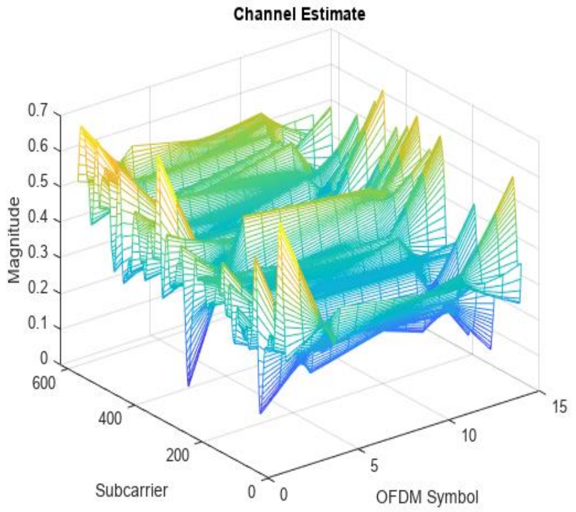


Figure 4.5 Plot of Channel Estimate

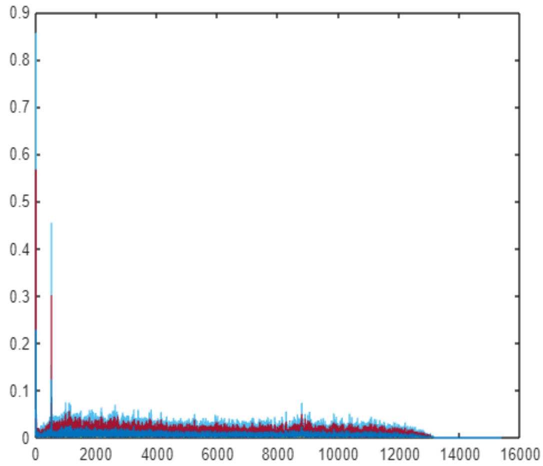


Figure 4.6 Plot of Magnitude

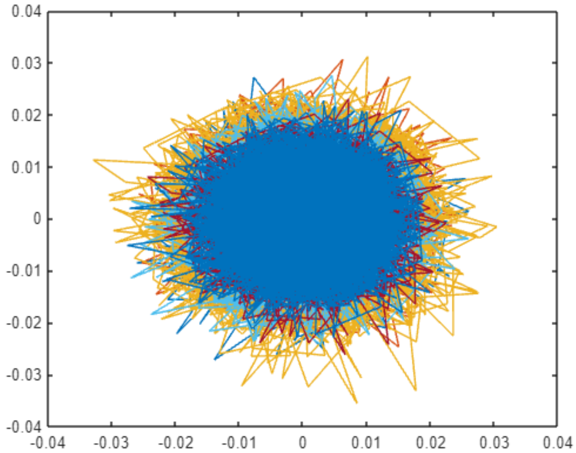


Figure 4.7 Plot of transmitted wave

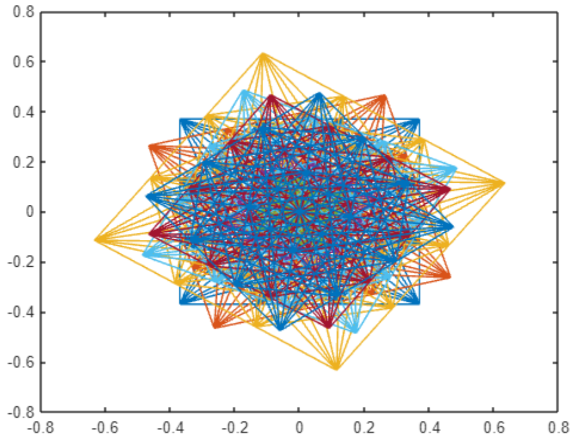


Figure 4.8 Plot of precoded symbols

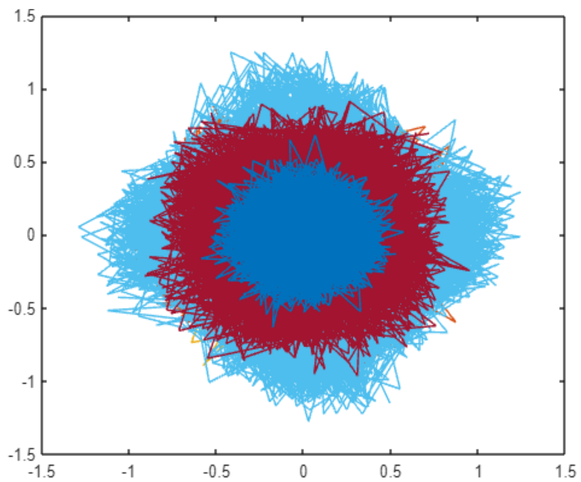


Figure 4.9 Plot of received signal

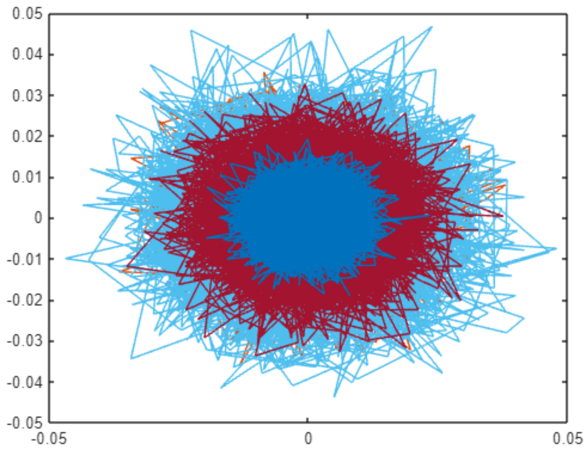


Figure 4.10 Plot of received waveform

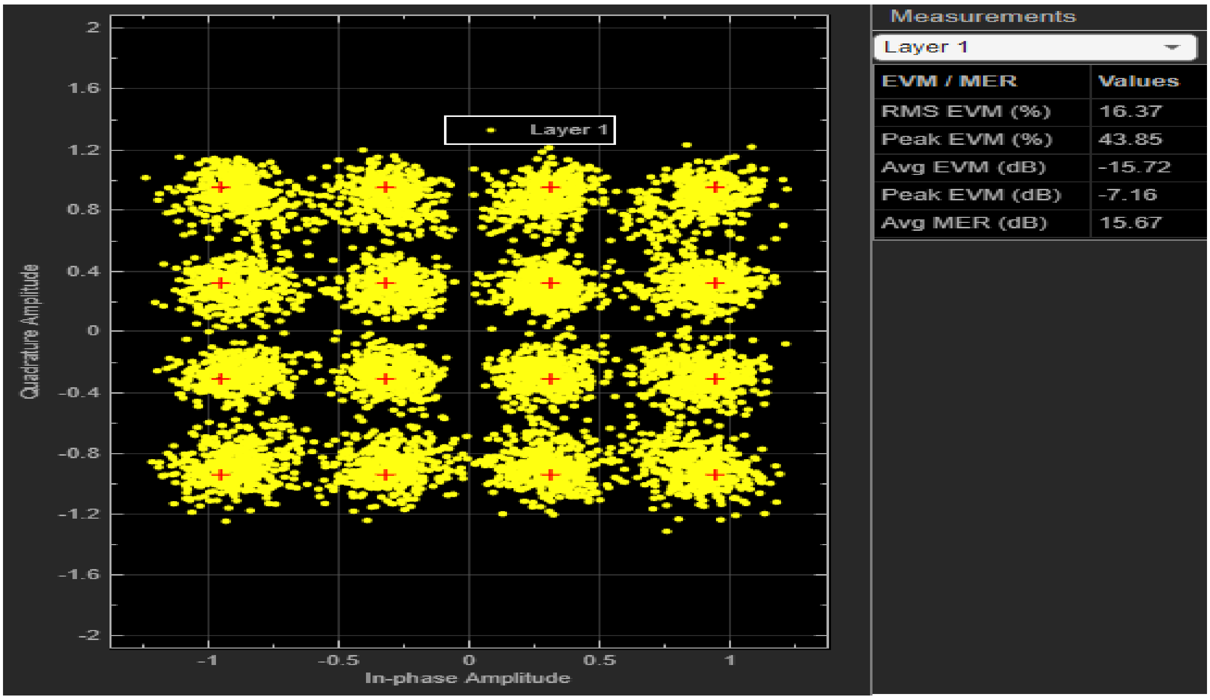


Figure 4.11 Equalization constellation diagram

4.4 BLER results

Calculate the BLER and the throughput (percentage of successfully received transport blocks). To provide statistically meaningful results, run this simulation for many transport blocks.

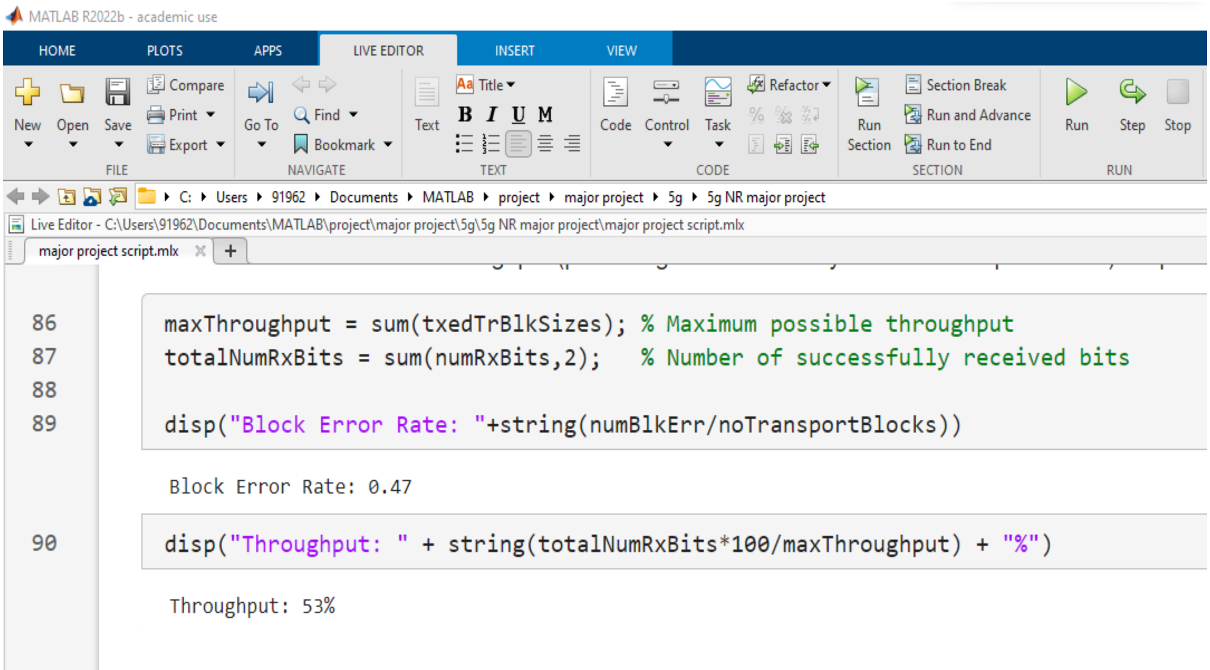


Figure 4.12 BLER Results

4.5 Discussion

After meeting our first objective i.e. "To develop a MATLAB-based model for simulating the 5G NR downlink transport channel.", we studied the channel for different parameters values including the transport block size and modulation schemes to find out "How does the performance of the proposed model vary with different transport block size and modulation schemes". It was observed that varying the code rate and Sub-carrier spacing had no change on the performance of the model. In fact, as expected the change in Transport block size and Modulation Scheme did result in variation of the block error rate and throughput of the model.

The following table list the input parameters and their values for this model.

S. NO.	PARAMETER	VALUE
1	SNR (in db)	7
2	No. of Slots	20
3	Transport block size	100
4	No. of Layers	20
5	DMRS length	2
6	No. of HARQ process	16
7	Code rate	490/1024
8	Maximum LDPC count	6
9	LDPC decoding algorithm	Normalized min-sum
10	No. of antennas	8
11	Delay profile	"TDL-C"
12	Modulation scheme	QPSK, 16QAM, 64QAM, 256QAM

Table 5.1 Input Parameters & values

The block error rate was as low as 0.377 for transport block size of 1000 blocks for '16QAM' modulation scheme and was as high as 1.0 for Transport block size of 10 blocks and modulation scheme of '64QAM' and '256QAM'.

Also, the throughput became as low as 0% for modulation scheme of '64QAM' and '256QAM' at Transport block size of 10 blocks and it became as High as 62% for transport block size of 1000 blocks at '16QAM' modulation scheme. The recorded data is shown in the table 2.1 and 2.2 below:-

Transport Block size	16QAM	64QAM	256QAM
10	0.7	1.0	1.0
50	0.44	0.64	0.68
100	0.42	0.52	0.68
200	0.43	0.52	0.68
300	0.41	0.53	0.68
500	0.388	0.512	0.68
1000	0.377	0.504	0.672

Table 5.2 Block error ratio for variable Transport Block size

Transport Block size	16QAM	64QAM	256QAM
10	30	0	0
50	56	36	32
100	58	48	32
200	56.5	48	32
300	58.6	48	32
500	61.2	48.8	32
1000	62.3	49.6	32.8

Table 5.3 Throughput in percentage for variable Transport Block size

It can be inferred that as modulation scheme become complex from '16QAM' to '64QAM' to finally '256QAM', the block error rate increases. The block error rate also increases with decrease in size of Transport blocks. Similarly, the throughput decreases as the modulation scheme become complex from '16QAM' to '64QAM' to finally '256QAM'. The throughput also increases with increase in the Transport block size.

4.5.1 Channel Improvements

After meeting our second objective i.e. "To evaluate the performance of the model under various transport block size and modulation schemes," we tried to study the effects of Channel improvements techniques on channel performance. The recorded data is as shown in Table 3.1 and Table 3.2 below:

Transport Block size	16QAM	64QAM	256QAM
10	0	0	0.5
50	0	0	0.36
100	0	0	0.48
200	0	0	0.475
300	0	0.056	0.476
500	0	0.086	0.464
1000	0	0.126	0.472

Table 5.4 Block error ratio for variable Transport Block size for improved channel model

Transport Block size	16QAM	64QAM	256QAM
10	100	100	50
50	100	100	64
100	100	100	52
200	100	100	52.5
300	100	94.33	52.3
500	100	91.4	53.6
1000	100	87.4	52.8

Table 5.5 Throughput in percentage for variable Transport Block size

We inferred that Channel improvement techniques did work and reduce the block error rate to as low as 0 and Throughput as high as 100%. However, the effect of complexity of modulation scheme and Transport block size was still observed after these channel improvements techniques. The variation behaviour did not change at all with the previously obtained result.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

This project's goal is to model DOWNLINK TRANSPORT CHANNEL for 5G NR in MATLAB. In model-1 we were able to successfully model some link elements of 5G NR DOWNLINK TRANSPORT CHANNELs with multiple hybrid automatic repeat-request (HARQ) that include the following elements

i. DL-SCH encoding and decoding.
ii. Physical downlink shared channel (PDSCH) encoding and decoding.
iii. HARQ management

Modelling the above elements at certain simulation parameters resulted in a successful model that gives a BER (Block error ratio) of 0.47 and throughput of 53% at 16 QAM.

However, model-2 was a complete downlink transport channel model along with channel estimate and other channel improving techniques that recorded a BER (Block error ratio) of 0 and throughput of 100% at 16 QAM because of channel improvements.

Finally, after thorough simulations we observed that as modulation scheme become complex from '16QAM' to '64QAM' to finally '256QAM', the block error rate increases which also increases with decrease in size of Transport blocks. Similarly, the throughput decreases as the modulation scheme become complex from '16QAM' to '64QAM' to finally '256QAM' whereas the throughput also increases with increase in the Transport block size.

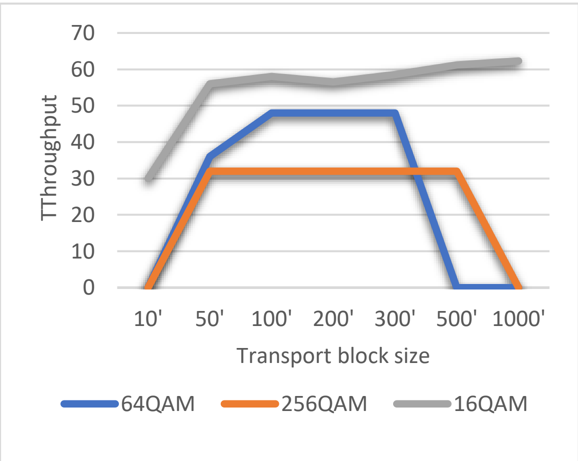


Figure 5.1. Throughput without channel improvement techniques

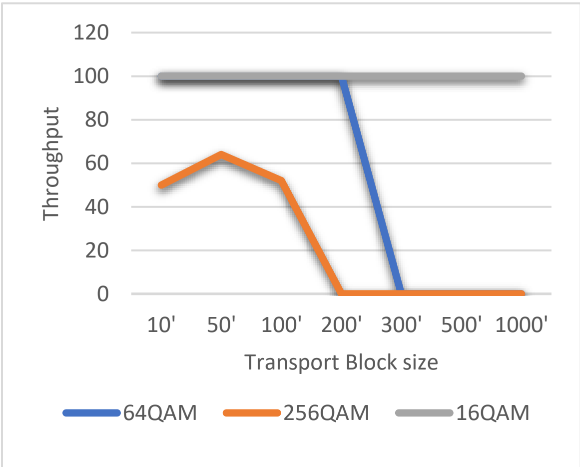


Figure 5.2 Throughput with channel improvement techniques

Also, after improving the channel using channel improvements techniques like Channel Estimation and Equalization, MIMO Precoding and Timing Synchronisation we observed that channel performance did increase by a considerable factor.

Above charts in figure 9 and figure 10 summarises table 2.2 and table 3.2. They show how the overall throughput increased after using channel improvement techniques irrespective of the modulation scheme we used.

It is also worth noting that varying the code rate and Sub-carrier spacing had no change on the overall performance of the channel model.

5.2 Future Scope

This model creates a MATLAB live script for the complete 5G NR DOWNLINK TRANSPORT CHANNEL model in MATLAB. By using this DOWNLINK TRANSPORT CHANNEL model with any wireless system design in MATLAB and Simulink, one can optimize link performance, perform system architecture trade-offs, and provide a realistic assessment of the overall system performance. It can also be used to predict link performance (e.g., BER) in a single-user scenario, system performance (e.g., throughput, latency) in a multi-user scenario and reduce the need for costly channel measurement projects.

Further, our observation and recorded data can be used to study and model better communication channel for 5G New Radio.

REFERENCES

1. G. Sanfilippo, O. Galinina, S. Andreev, S. Pizzi and G. Araniti: A concise review of 5g new radio capabilities for directional access at mmwave frequencies, Internet of Things Smart Spaces and Next Generation Networks and Systems, vol. 11118, pp. 340-354, (2018).
2. Zaidi, A., Baldemair, R., Tullberg, H., Bjo¨rkegren, H., Sundstro¨m, L., Medbo, J., et al.: Waveform and numerology to support 5G services and requirements. IEEE Communications Magazine, vol. 54, no. 11, pp. 90–98, (2016).
3. Bhushan, N., Ji, T., Koymen, O., Smee, J., Soriaga, J., Subramanian, S., & Wei, Y: Industry perspectives: 5G air interface system design principles. IEEE Wireless Communications, vol 24, no. 5, pp. 6–8, (2017).
4. Parkvall, S., Dahlman, E., Furuska¨r, A., & Frenne, M.: 5G NR: the next generation wireless access technology, IEEE Communications Standards Magazine, vol. 1, no. 4, pp. 24–30, (2017)
5. M. R. Avendi, M. R. Javan, and H. Jafarkhani: Uplink channel modelling and capacity analysis for 5G cellular networks, IEEE Trans. Wireless Commun., vol. 17, no. 8, pp. 5107-5119, (2018).
6. Y. Wang, F. Zhu, X. You, and L. Liu: Channel models for 5G communication systems, IEEE Wireless Commun., vol. 24, no. 4, pp. 136-143, (2017).
7. Erik Dahlman, Stefan Parkvall, Johan Sköld: Chapter 9- Transport channel processing; 5G NR: the Next Generation Wireless Access Technology, Academic Press, ISBN 9780128143230, pp. 153-182, (2018).
8. Youngrok Jang: Study on channel model for frequencies from 0.5 to 100 GHz, Technical Report, 3GPP (3rd Generation Partnership Project), TR 38.901, Version 14.0.0, (2017)
9. S. Sun and T. S. Rappaport: Path loss models for 5G millimeter wave propagation channels in urban microcells, IEEE Trans. Wireless Commun., vol. 16, no. 9, pp. 6171-6183, (2017).
10. P. Fan, Y. Liu, F. Gao, Y. Cai, and X. Yin: Uplink channel modelling for 5G heterogeneous networks, IEEE Commun. Mag., vol. 54, no. 6, pp. 62-68, (2016).
11. Y. Wang, F. Gao, H. Zhang, W. Zhang, and L. Su: Uplink channel modelling and capacity analysis for 5G cellular networks, IEEE Access, vol. 6, pp. 11759-11769, (2018).
12. Rinaldi, F., Raschellà, A. & Pizzi, S.: 5G NR system design: a concise survey of key features and capabilities., Wireless Network, vol. **27**, pp. 5173–5188, (2021).
13. Yan Cheng: 5G; NR; Multiplexing and channel coding, Technical Specification Group Radio Access Network, 3GPP (3rd Generation Partnership Project), TS 38.212 version 16.2.0 Release 16, (2020)
14. J. W. Wallace and M. A. Jensen: Statistical characteristics of measured MIMO wireless channel data and comparison to conventional models, IEEE 54th Vehicular Technology Conference. VTC Fall 2001, vol.2, pp. 1078-1082, (2001).

15. M. Lauridsen, L. C. Gimenez, I. Rodriguez, T. B. Sorensen and P. Mogensen: From lte to 5g for connected mobility, IEEE Communications Magazine, vol. 55, no. 3, pp. 156-162, (2017).
16. Z. Ding, Y. Liu, J. Choi, Q. Sun, M. Elkashlan, I. Chih-Lin, et al.: Application of non-orthogonal multiple access in lte and 5g networks, IEEE Communications Magazine, vol. 55, no. 2, pp. 185-191, (2017).
17. A. Rahman, S. Arabi and R. Rab: Feasibility and challenges of 5g network deployment in least developed countries (ldc), Wireless Sensor Network, vol. 13, no. 1, pp. 1-16, (2021).

Appendix-1 MATLAB code explaination

Introduction

This appendix explains the downlink transport channel MATLAB code.

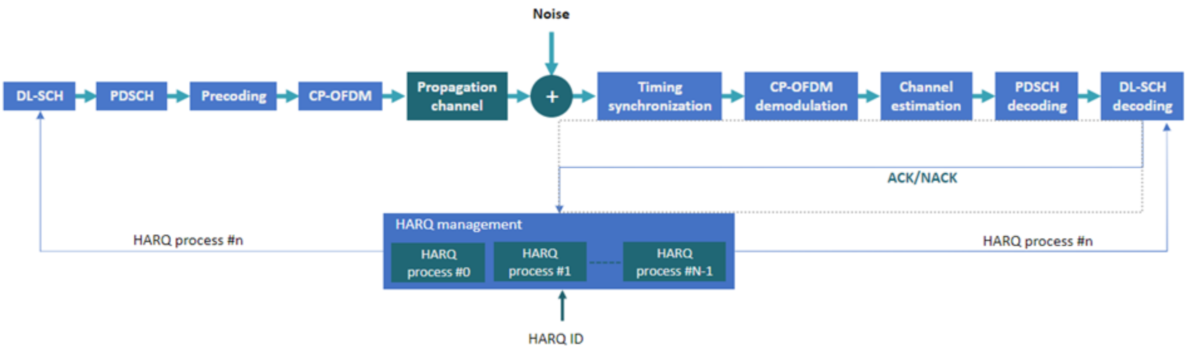


Figure 1. 5G NR Downlink transport channel full processing chain

Simulation Parameters

Specify the signal-to-noise ratio (SNR), number of slots to simulate, perfect channel estimation flag, no. of slots and transport block size

```
SNRdB = 7;                      % SNR in dB

totalNoSlots = 20;               % Number of slots to simulate

perfectEstimation = false; % Perfect synchronization and channel estimation

rng("default");                  % Set default random number generator for
repeatability

noTransportBlocks = 100;     % No. of transport block
```

Carrier Configuration

.This object controls the numerology, such as, the subcarrier spacing, bandwidth, and cyclic prefix (CP) length. This example uses the default set of properties.

```
carrier = nrCarrierConfig
```

PDSCH and DM-RS Configuration

Define the modulation scheme (16-QAM) and the number of layers. Allocate all resource blocks (RBs) to the PDSCH (full band allocation). You can also specify other time-allocation parameters and demodulation reference signal (DM-RS) settings in this object.

```
pdsch = nrPDSCHConfig;

pdsch.Modulation = "16QAM";

pdsch.NumLayers = 1;
```

```
pdsch.PRBSet = 0:carrier.NSizeGrid-1;     % Full band allocation
```

Set the DM-RS parameters. To improve channel estimation, add an additional DM-RS position.

```
pdsch.DMRS.DMRSAdditionalPosition = 1;
```

Set the DM-RS configuration type and the DM-RS length, which determines the number of orthogonal DM-RS sequences or DM-RS ports.

i. `DMRSConfigurationType = 1` supports up to 4 DM-RS ports when `DMRSLength = 1`..
ii. `DMRSConfigurationType = 1` supports up to 8 DM-RS ports when `DMRSLength = 2`.
iii. `DMRSConfigurationType = 2` supports up to 6 DM-RS ports when `DMRSLength = 1`. This is designed for multi-user MIMO (MU-MIMO).
iv. `DMRSConfigurationType = 2` supports up to 12 DM-RS ports when `DMRSLength = 2`. This is designed for MU-MIMO.

The maximum number of layers must be less than or equal to the number of DM-RS ports.

```
pdsch.DMRS.DMRSConfigurationType = 1;

pdsch.DMRS.DMRSLength = 2;

pdsch.DMRS                           % Display DM-RS properties
```

DL-SCH Configuration

Specify the code rate, the number of HARQ processes, and the redundancy version (RV) sequence values. This sequence controls the RV retransmissions in case of error. To disable HARQ retransmissions, you can set `rvSeq` to a constant value.

```
NHARQProcesses = 16;     % Number of parallel HARQ processes

rvSeq = [0 2 3 1];

codeRate= 490/1024;
```

Create the DL-SCH encoder and decoder objects. To use multiple processes, set the `MultipleHARQProcesses` property to `true` for both objects. You do not need to specify the number of HARQ processes. The DL-SCH encoder and decoder objects can model up to 16 HARQ processes. To identify the active HARQ process when performing operations with the DL-SCH encoder and decoder objects, use the `HARQprocessID` property of the HARQ entity object,

```
% Create DL-SCH encoder object

encodeDLSCH = nrDLSCH;

encodeDLSCH.MultipleHARQProcesses = true;

encodeDLSCH.TargetCodeRate = codeRate;

% Create DLSCH decoder object
decodeDLSCH = nrDLSCHDecoder;

decodeDLSCH.MultipleHARQProcesses = true;
```

```
decodeDLSCH.TargetCodeRate = codeRate;
decodeDLSCH.LDPCDecodingAlgorithm = "Normalized min-sum";
decodeDLSCH.MaximumLDPCIterationCount = 6;
```

HARQ Management

Create a HARQ entity object to manage the HARQ processes and the DL-SCH encoder and decoder buffers. For each HARQ process, a HARQ entity stores these elements:

i. HARQ ID number.
ii. RV.
iii. Transmission number, which indicates how many times a certain transport block has been transmitted.
iv. Flag to indicate whether new data is required. New data is required when a transport block is received successfully or if a sequence timeout has occurred (all RV transmissions have failed).
v. Flag to indicate whether a sequence timeout has occurred (all RV transmissions have failed).

```
harqEntity = HARQEntity(0:NHARQProcesses-1,rvSeq,pdsch.NumCodewords);
```

Channel Configuration

Specify the number of transmit and receive antennas.

```
nTxAnts = 8;
nRxAnts = 8;
% Check that the number of layers is valid for the number of antennas
if pdsch.NumLayers > min(nTxAnts,nRxAnts)
    error("The number of layers ("+string(pdsch.NumLayers)+") must be smaller than min(nTxAnts,nRxAnts) ("+string(min(nTxAnts,nRxAnts))+")")
end
```

Create a channel object.

```
Channel = nrTDLChannel;
channel.DelayProfile = "TDL-C";
channel.NumTransmitAntennas = nTxAnts;
channel.NumReceiveAntennas = nRxAnts;
```

Set the channel sample rate to that of the OFDM signal. To obtain the sampling rate of the OFDM signal, use the **nrOFDMInfo** function.

```
ofdmInfo = nrOFDMInfo(carrier);
channel.SampleRate = ofdmInfo.SampleRate;
```

Transmission and Reception

Set up a loop to simulate the transmission and reception of slots. Create a **comm.ConstellationDiagram** to display the constellation of the equalized signal.

```
constPlot = comm.ConstellationDiagram;
% Constellation diagram object

constPlot.ReferenceConstellation =
getConstellationRefPoints(pdsch.Modulation); % Reference constellation values

constPlot.EnableMeasurements = 1;
% Enable EVM measurements

% Initial timing offset
offset = 0;

estChannelGrid = getInitialChannelEstimate(channel,carrier);
newPrecodingWeight =
getPrecodingMatrix(pdsch.PRBSet,pdsch.NumLayers,estChannelGrid);
```

BER Simulation

```
% Initialize loop variables

noiseVar = 1./(10.^(SNRdB/10)); % Noise variance

numBlkErr = 0;                  % Number of block errors

numRxBits = [];        % Number of successfully received bits per transmission

txedTrBlkSizes = [];        % Number of transmitted info bits per transmission

for nTrBlk = 1:noTransportBlocks
% A transport block or transmission time interval (TTI) corresponds to 1 slot

    carrier.NSlot = carrier.NSlot+1;
```

Calculate Transport Block Size

The transport block size is the number of bits to send to the channel coding stages. This value depends on the capacity of the PDSCH. To calculate the transport block size, use the `nrTBS` function.

```
    % Generate PDSCH indices info, which is needed to calculate the transport

    % block size

    [pdschIndices,pdschInfo] = nrPDSCHIndices(carrier,pdsch);

    % Calculate transport block sizes
    Xoh_PDSCH = 0;

    trBlkSizes =
nrTBS(pdsch.Modulation,pdsch.NumLayers,numel(pdsch.PRBSet),pdschInfo.NREPerPRB
,codeRate,Xoh_PDSCH);
```

HARQ Processing (Buffer Management)

i. DL-SCH encoder buffers: Generate a new transport block if new data is required for the active HARQ process. Store the transport block in the corresponding buffer. If new data is not required, the DL-SCH encoder uses its buffered bits for retransmission.

ii. DL-SCH decoder buffers: The soft buffers in the receiver store previously received versions of the same codeword. These buffers are cleared automatically upon successful reception (no CRC error). However, if the RV sequence ends without successful decoding, flush the buffers manually by using the `resetSoftBuffer` object function.

```
    % Get new transport blocks and flush decoder soft buffer, as required
    for cwIdx = 1:pdsch.NumCodewords
        if harqEntity.NewData(cwIdx)
            % Create and store a new transport block for transmission
            trBlk = randi([0 1],trBlkSizes(cwIdx),1);
            setTransportBlock(encodeDLSCH,trBlk,cwIdx-
1,harqEntity.HARQProcessID);
            % If the previous RV sequence ends without successful
            % decoding, flush the soft buffer
            if harqEntity.SequenceTimeout(cwIdx)
                resetSoftBuffer(decodeDLSCH,cwIdx-1,harqEntity.HARQProcessID);
            end
        end
    end
```

DL-SCH Encoding

Encode the transport blocks.

```
    codedTrBlock =
encodeDLSCH(pdsch.Modulation,pdsch.NumLayers,pdschInfo.G,harqEntity.Redundancy
Version,harqEntity.HARQProcessID);
```

PDSCH Modulation and MIMO Precoding

Generate PDSCH symbols from the coded transport blocks.

```
    pdschSymbols = nrPDSCH(carrier,pdsch,codedTrBlock);
```

Get the precoding weights. This example assumes channel knowledge for precoding

```
    precodingWeights = newPrecodingWeight;
```

Precode the PDSCH symbols.

```
    pdschSymbolsPrecoded = pdschSymbols*precodingWeights;
```

PDSCH DM-RS Generation

Generate DM-RS symbols and indices.

```
dmrsSymbols = nrPDSCHDMRS(carrier,pdsch);
dmrsIndices = nrPDSCHDMRSIndices(carrier,pdsch);
```

Mapping to Resource Grid

Generate an empty resource grid. This grid represents a slot.

```
pdschGrid = nrResourceGrid(carrier,nTxAnts);
```

The `nrPDSCHIndices` function generates indices that refer to layers and not antennas.. In this case, the resulting resource grids are not precoded. This figure shows the mapping process of the PDSCH symbols to as many resource grids as layers.

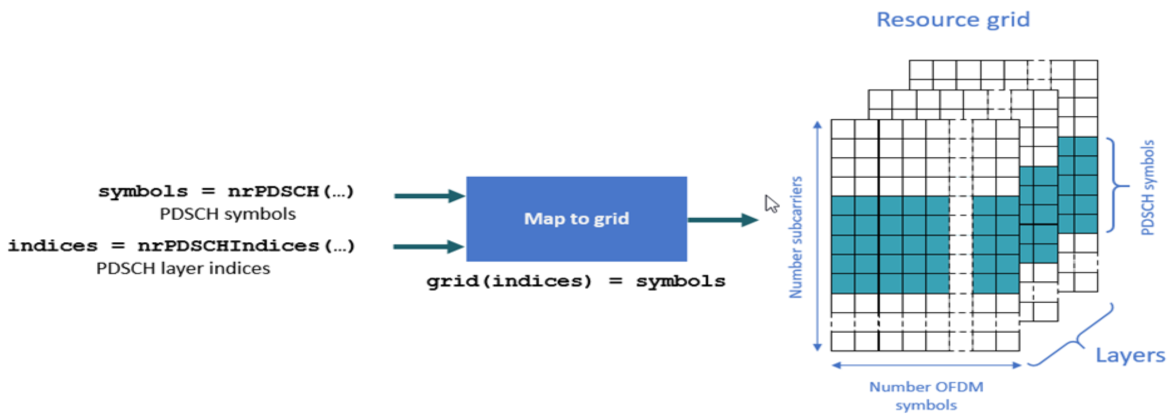


Figure 2. Mapping process of the PDSCH symbols to as many resource grids as layers.

Because we applied MIMO precoding to the PDSCH symbols before mapping them to the resource grids, the MIMO-precoded PDSCH symbols refer to antennas and not layers. To

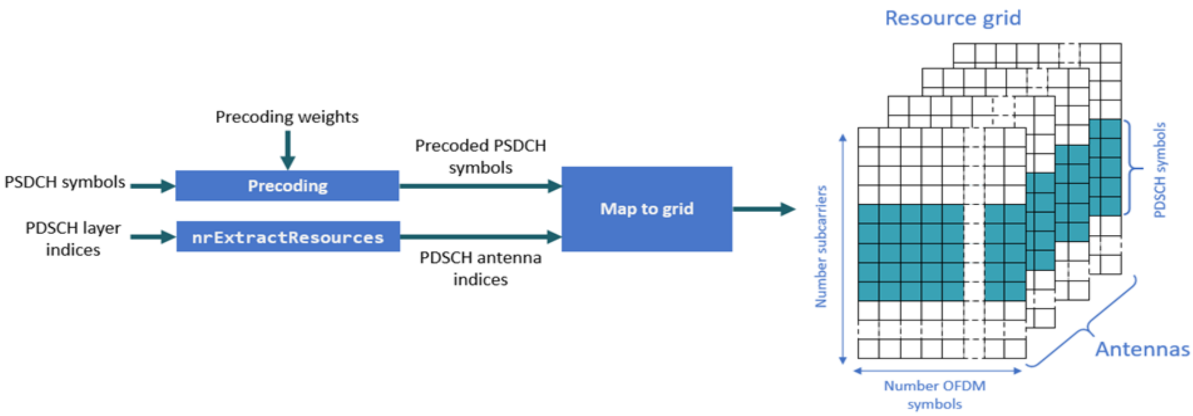


Figure 3. Mapping process of MIMO-precoded symbols to as many resource grids as transmit antennas.

convert layer indices to antenna indices, we use the **`nrExtractResources`** function. This figure shows the mapping process of MIMO-precoded symbols to as many resource grids as transmit antennas.

```
[~,pdschAntIndices] = nrExtractResources(pdschIndices,pdschGrid);
pdschGrid(pdschAntIndices) = pdschSymbolsPrecoded;
```

MIMO-precode and map the DM-RS symbols to the resource grid. Similar to the PDSCH indices, the DM-RS indices refer to layers. To convert these layer indices to antenna indices, we use the **nrExtractResources** function again.

```
% PDSCH DM-RS precoding and mapping
for p = 1:size(dmrsSymbols,2)
    [~,dmrsAntIndices] = nrExtractResources(dmrsIndices(:,p),pdschGrid);
    pdschGrid(dmrsAntIndices) = pdschGrid(dmrsAntIndices) + 
dmrsSymbols(:,p)*precodingWeights(p,:);
end
```

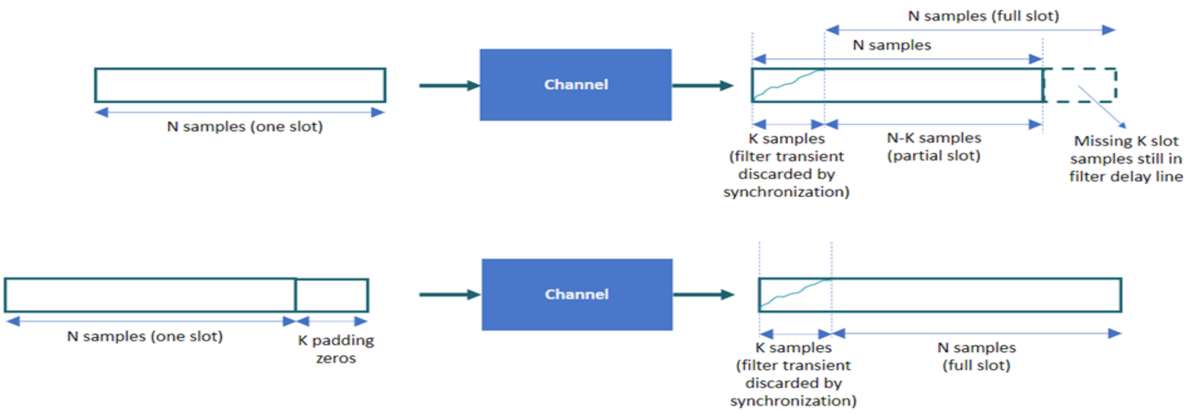
OFDM Modulation

OFDM-modulate the resource grid.

```
[txWaveform,waveformInfo] = nrOFDMModulate(carrier,pdschGrid);
```

Propagation Channel

The propagation channel generates *N* output samples for an input with *N* samples. However, the block of *N* output samples includes the channel filter transient (*K* samples).. *N-K* samples are not enough to decode a slot-worth of data. Part of the slot samples are in the channel filter delay line and are not flushed yet. To flush all relevant samples out of the channel filter, we padded the input signal with zeros.



Pad the input signal with enough zeros to ensure that the generated signal is flushed out of the channel filter.

```
chInfo = info(channel);
```

```
    maxChDelay = ceil(max(chInfo.PathDelays*channel.SampleRate)) +
chInfo.ChannelFilterDelay;

    txWaveform = [txWaveform; zeros(maxChDelay,size(txWaveform,2))];
```

Send the signal through the channel and add noise.

```
    [rxWaveform,pathGains,sampleTimes] = channel(txWaveform);

    noise = generateAWGN(SNRdB,nRxAnts,waveformInfo.Nfft,size(rxWaveform));

    rxWaveform = rxWaveform + noise;
```

Timing Synchronization

One can perform perfect or practical synchronization.

- Perfect synchronization assumes channel knowledge (**`nrPerfectTimingEstimate`**). The channel returns information on path gains and path filters impulse response. You can use this information to determine the offset that is associated with the strongest multipath component across all channel snapshots and across all transmit and receive antennas.
- Practical synchronization performs a cross-correlation of the received signal with the PDSCH DM-RS symbols in the time domain (**`nrTimingEstimate`**). In some adverse cases, this cross-correlation can be weak due to fading or noise, resulting in an erroneous timing offset. The function **`hSkipWeakTimingOffset`** checks the magnitude of the cross-correlation `mag`. If the cross-correlation is weak, the function ignores the current timing estimate and instead uses the previous estimate (`offset`).

Perform perfect or practical timing estimation and synchronization.

```
    if perfectEstimation

        % Get path filters for perfect timing estimation

        pathFilters = getPathFilters(channel);

        [offset,mag] = nrPerfectTimingEstimate(pathGains,pathFilters);

    else

        [t,mag] =
nrTimingEstimate(carrier,rxWaveform,dmrsIndices,dmrsSymbols);

        offset = hSkipWeakTimingOffset(offset,t,mag);

    end

    rxWaveform = rxWaveform(1+offset:end,:);
```

OFDM Demodulation

OFDM-demodulate the synchronized signal.

```
    rxGrid = nrOFDMDemodulate(carrier,rxWaveform);
```

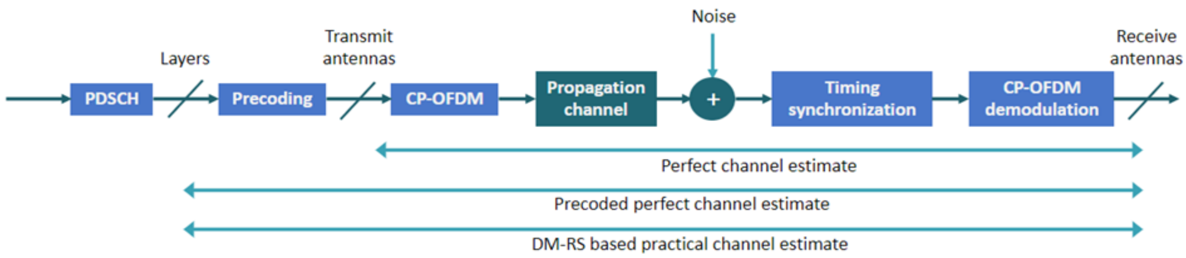
Channel Estimation

Channel estimation provides a representation of the channel effects per resource element (RE). The equalizer uses this information to compensate for the distortion introduced by the channel.

One can perform perfect or practical channel estimation.

i. Perfect channel estimation assumes channel knowledge (**`nrPerfectChannelEstimate`**). The perfect channel estimate represents the channel conditions between the transmit and receive antennas. Because the equalizer requires channel knowledge between the transmit layers and the receive antennas, you must apply precoding to the perfect channel estimate.
ii. Practical channel estimation uses the PDSCH DM-RS to estimate the channel conditions and uses noise averaging and interpolation to obtain an estimate for all REs in the slot. Because the DM-RSs are specified per layer, the resulting practical channel estimate represents the channel conditions between the transmit layers and the receive antennas. The practical channel estimate includes the effect of the MIMO precoding operation.

This figure shows the reference points of the channel estimates in the downlink processing chain.



Perform perfect or practical channel estimation.

```
    if perfectEstimation
 % Perform perfect channel estimation between transmit and receive antennas.

        estChGridAnts =
nrPerfectChannelEstimate(carrier,pathGains,pathFilters,offset,sampleTimes);

        % Get perfect noise estimate (from noise realization)
        noiseGrid = nrOFDMDemodulate(carrier,noise(1+offset:end ,:));

        noiseEst = var(noiseGrid(:));

        % Get precoding matrix for next slot
        newPrecodingWeight =
getPrecodingMatrix(pdsch.PRBSet,pdsch.NumLayers,estChGridAnts);

        % Apply precoding to estChGridAnts. The resulting estimate is for
        % the channel estimate between layers and receive antennas.

        estChGridLayers =
precodeChannelEstimate(estChGridAnts,precodingWeights.');

    else

% Perform practical channel estimation between layers and receive antennas.
```

```
        [estChGridLayers,noiseEst] =
nrChannelEstimate(carrier,rxGrid,dmrsIndices,dmrsSymbols,'CDMLengths',pdsch.DMRS.CDMLengths);

    % Remove precoding from estChannelGrid before precoding matrix calculation
        estChGridAnts =
precodeChannelEstimate(estChGridLayers,conj(precodingWeights));

        % Get precoding matrix for next slot
        newPrecodingWeight =
getPrecodingMatrix(pdsch.PRBSet,pdsch.NumLayers,estChGridAnts);

    end
```

Plot the channel estimate between the first layer and the first receive antenna.

```
    mesh(abs(estChGridLayers(:,:,1,1)));

    title('Channel Estimate');

    xlabel('OFDM Symbol');

    ylabel("Subcarrier");

    zlabel("Magnitude");
```

At this point, one can use the channel estimate to obtain the precoding matrix for transmission in the next slot

Equalization

The equalizer uses the channel estimate to compensate for the distortion introduced by the channel.

Extract the PDSCH symbols from the received grid and associated channel estimates. The `csi` output has channel state information (CSI) for each of the equalized PDSCH symbols. The CSI is a measure of the channel conditions for each PDSCH symbol. Use the CSI to weight the decoded soft bits after PDSCH decoding, effectively increasing the importance of symbols experiencing better channel conditions.

```
     [pdschRx,pdschHest] =
nrExtractResources(pdschIndices,rxGrid,estChGridLayers);

    [pdschEq,csi] = nrEqualizeMMSE(pdschRx,pdschHest,noiseEst);
```

Plot the constellation of the equalized symbols. The plot includes the constellation diagrams for all layers.

```
    constPlot.ChannelNames = "Layer "+(pdsch.NumLayers:-1:1);

    constPlot.ShowLegend = true;

  % Constellation for the first layer has a higher SNR than that for the
  % last layer. Flip the layers so that the constellations do not mask
  % each other.
    constPlot(fliplr(pdschEq));
```

PDSCH Decoding

Decode the equalized PDSCH symbols and obtain the soft bit codewords.

```
    [dlschLLRs,rxSymbols] = nrPDSCHDecode(carrier,pdsch,pdschEq,noiseEst);
```

Scale the soft bits or log-likelihood ratios (LLRs) by the CSI. This scaling applies a larger weight to the symbols in the REs with better channel conditions.

```
    % Scale LLRs by CSI

     csi = nrLayerDemap(csi);                                  % CSI layer demapping

    for cwIdx = 1:pdsch.NumCodewords

    Qm = length(dlschLLRs{cwIdx})/length(rxSymbols{cwIdx}); % Bits per symbol

       csi{cwIdx} = repmat(csi{cwIdx}.',Qm,1); % Expand by each bit per symbol

        dlschLLRs{cwIdx} = dlschLLRs{cwIdx} .* csi{cwIdx}(:);   % Scale

    end
```

DL-SCH Decoding

Decode the LLRs and check for errors.

```
    decodeDLSCH.TransportBlockLength = trBlkSizes;

    [decbits,blkerr] = decodeDLSCH(dlschLLRs,pdsch.Modulation,pdsch.NumLayers,
...harqEntity.RedundancyVersion,harqEntity.HARQProcessID);
```

Results

Store the results to calculate the BLER.

```
    % Store values to calculate throughput (only for active transport blocks)

    if(any(trBlkSizes ~= 0))

        numRxBits = [numRxBits trBlkSizes.*(1-blkerr)];

        txedTrBlkSizes = [txedTrBlkSizes trBlkSizes];

    end

    if blkerr

        numBlkErr = numBlkErr + 1;

    end
```

HARQ Process Update

Update the current HARQ process with the resulting block error status, and then advance to the next process. This step updates the information related to the active HARQ process in the HARQ entity.

```
statusReport = updateAndAdvance(harqEntity,blkerr,trBlkSizes,pdschInfo.G);
```

Summarize HARQ and decoding information for the present slot.

```
    disp("Slot "+(nTrBlk)+". "+statusReport);
end % for nTrBlk = 1:noTransportBlocks
```

BLER Results

Calculate the BLER and the throughput (percentage of successfully received transport blocks). To provide statistically meaningful results, run this simulation for many transport blocks.

```
maxThroughput = sum(txedTrBlkSizes); % Maximum possible throughput

totalNumRxBits = sum(numRxBits,2);  % Number of successfully received bits

disp("Block Error Rate: "+string(numBlkErr/noTransportBlocks))
disp("Throughput: " + string(totalNumRxBits*100/maxThroughput) + "%")
```

Plots

```
plot(mag) %magnitude
plot(pdschSymbolsPrecoded) % PDSCH precoded symbol

plot(pdschRx) %PDSCH recieved signal

plot(txWaveform) % PDSCH transmitted waveform

plot(noise) %AWGN

plot(rxWaveform) % recieved waveform
```

Local Functions

```
function noise = generateAWGN(SNRdB,nRxAnts,Nfft,sizeRxWaveform)

% Generate AWGN for a given value of SNR in dB (SNRDB), which is the
% receiver SNR per RE and antenna, assuming the channel does
% not affect the power of the signal. NRXANTS is the number of receive
% antennas. NFFT is the FFT size used in OFDM demodulation. SIZERXWAVEFORM
% is the size of the receive waveform used to calculate the size of the
% noise matrix.
% Normalize noise power by the IFFT size used in OFDM modulation, as
% the OFDM modulator applies this normalization to the transmitted
% waveform. Also normalize by the number of receive antennas, as the
% channel model applies this normalization to the received waveform by
```

```
% default. The SNR is defined per RE for each receive antenna (TS
% 38.101-4).
    SNR = 10^(SNRdB/10); % Calculate linear noise gain

    N0 = 1/sqrt(2.0*nRxAnts*double(Nfft)*SNR);

    noise = N0*complex(randn(sizeRxWaveform),randn(sizeRxWaveform));

end

function wtx = getPrecodingMatrix(PRBSet,NLayers,hestGrid)

% Calculate precoding matrix given an allocation and a channel estimate

    % Allocated subcarrier indices

    allocSc = (1:12)' + 12*PRBSet(:).';

    allocSc = allocSc(:);

    % Average channel estimate

    [~,~,R,P] = size(hestGrid);

    estAllocGrid = hestGrid(allocSc,:,:,:);

    Hest = permute(mean(reshape(estAllocGrid,[],R,P)),[2 3 1]);

    % SVD decomposition

    [~,~,V] = svd(Hest);

    wtx = V(:,1:NLayers).';

    wtx = wtx/sqrt(NLayers); % Normalize by NLayers

end

function estChannelGrid = getInitialChannelEstimate(channel,carrier)
% Obtain an initial channel estimate for calculating the precoding matrix.
% This function assumes a perfect channel estimate
    % Clone of the channel
    chClone = channel.clone();

    chClone.release();

    % No filtering needed to get channel path gains

    chClone.ChannelFiltering = false;

    % Get channel path gains

    [pathGains,sampleTimes] = chClone();

    % Perfect timing synchronization

    pathFilters = getPathFilters(chClone);

    offset = nrPerfectTimingEstimate(pathGains,pathFilters);

    % Perfect channel estimate

    estChannelGrid =
nrPerfectChannelEstimate(carrier,pathGains,pathFilters,offset,sampleTimes);

end
```

```
function refPoints = getConstellationRefPoints(mod)
% Calculate the reference constellation points for a given modulation
% scheme.
    switch mod
        case "QPSK"
            nPts = 4;
        case "16QAM"
            nPts = 16;
        case "64QAM"
            nPts = 64;
        case "256QAM"
            nPts = 256;
    end
    binaryValues = int2bit(0:nPts-1,log2(nPts));
    refPoints = nrSymbolModulate(binaryValues(:),mod);
end
function estChannelGrid = precodeChannelEstimate(estChannelGrid,W)
% Apply precoding matrix W to the last dimension of the channel estimate.
    % Linearize 4-D matrix and reshape after multiplication

    K = size(estChannelGrid,1);
    L = size(estChannelGrid,2);
    R = size(estChannelGrid,3);
    estChannelGrid = reshape(estChannelGrid,K*L*R,[]);
    estChannelGrid = estChannelGrid*W;
    estChannelGrid = reshape(estChannelGrid,K,L,R,[]);
end
```

REFRENCES

1. 5G Toolbox Reference (https://in.mathworks.com/help/pdf_doc/5g/5g_ref.pdf)
2. 5G Channel Models: Requirements and Deployment Scenarios (https://www.techplayon.com/5g-channel-models-requirements-deployment-scenarios/)
3. Model 5G NR Transport Channels with HARQ (Model 5G NR Transport Channels with HARQ - MATLAB & Simulink (mathworks.com))
4. DL-SCH and PDSCH Transmit and Receive Processing Chain(DL-SCH and PDSCH Transmit and Receive Processing Chain - MATLAB & Simulink (mathworks.com)

Appendix -2 Research Paper

5G NR DOWNLINK TRANSPORT CHANNEL MODELLING USING MATLAB

KIET Group of Institution **Mridul Nigam[1], Dr. Narendra Kumar[2]**

***Abstract*- The fifth generation (5G) of mobile communication networks offers significant improvements in terms of data rates, latency, and reliability in comparison to 4G. One critical component of the 5G system is the downlink transport channel, which facilitates data transmission between the base station and the user equipment. Accurate modelling of the downlink transport channel is essential to optimize network performance and ensure seamless communication. In this study, we investigate the modelling of the downlink transport channel in 5G NR using MATLAB. We employ simulation-based techniques to evaluate the transport channel's performance under various scenarios and configurations. Our findings demonstrate the efficacy of the MATLAB-based approach in accurately predicting downlink transport channel characteristics, facilitating optimization of network design and deployment. This research contributes to the field of 5G NR downlink transport channel modelling by validating the effectiveness of MATLAB-based simulation techniques, enabling further study and refinement of network performance optimization.**

KEYWORDS

5G, 5G NR, Transport Channel, Downlink Transport Channel, 5G Equalization, MATLAB

1. INTRODUCTION

The 5G New Radio (NR) [1] standard introduces several new features and techniques to provide high data rates, low latency, and efficient use of radio resources. One of the key components of the 5G NR system is the downlink transport channel, which is responsible for transferring user data and control information between the base station and user equipment.

The 5G NR downlink transport channel uses a flexible structure that adapts to varying channel conditions and different types of services. The channel can be configured with different numerologies, which define the size and spacing of the subcarriers and the duration of the transmission time interval (TTI). This flexibility allows the system to support a wide range of use cases, from low-latency applications like vehicle-to-vehicle communication to high-throughput applications like video streaming.

The 5G NR [2] downlink transport channel also incorporates advanced error correction techniques to improve reliability and reduce latency. Forward error correction (FEC) is used to add redundancy to the transmitted data, allowing the receiver to correct errors without the need for retransmission. The channel coding scheme used in 5G NR is polar coding, which provides excellent error correction performance with low complexity.

In addition to the above features, the 5G NR downlink transport channel also supports beamforming and multiple input multiple output (MIMO) techniques to improve spectral efficiency and increase data rates. Beamforming enables the base station to direct radio energy towards specific users or areas, while MIMO allows multiple antennas to be used to transmit and receive data simultaneously.

Overall, the 5G NR downlink transport channel is a key component of the 5G system that enables high-speed, reliable, and low latency communication.

between the base station and user equipment. Its flexibility and advanced features make it suitable for a wide range of use cases and applications, and it is expected to play a crucial role in the development of the 5G ecosystem.

In this paper, we develop a MATLAB-based model for simulating the 5G NR downlink transport channel and evaluate the performance of the model under various transport block size and modulation schemes. We also compare the performance of the model with a model that make use of channel improvement methods like equalization, precoding, and channel estimation.

1.1 Channel Modelling

The modelling of the downlink transport channel is of utmost importance for the successful implementation of 5G NR technology. This is because the downlink transport channel is responsible for transmitting the user data and control information between the **UE** and the **gNB.** The efficient design and analysis of the downlink transport channel are essential to ensure that the data transmission is reliable, secure, and can support the massive data rates that 5G NR promises.

Accurate modelling of the 5G NR downlink transport channel is crucial for system-level simulations that evaluate the performance of the entire 5G system, including the radio access network (RAN) and the core network. This modelling enables the optimization of different parameters and features of the downlink transport channel, such as the modulation scheme, coding, and multiplexing techniques, to achieve the desired quality of service and user experience.

Moreover, modelling the 5G NR downlink transport channel also helps in the development and testing of advanced techniques such as MIMO, beamforming, and massive connectivity that are the key enablers of 5G technology. By accurately modelling these techniques, it is possible to analyse their impact on the performance of the downlink transport channel and optimize them for better system-level performance.

In summary, the importance of downlink transport channel modelling for 5G NR cannot be overstated. It plays a critical role in the successful implementation and optimization of 5G technology, enabling high-speed data transmission and providing a seamless user experience.

1.2 Research Methodology

The methodology for "5G NR Downlink transport channel modelling using MATLAB" involves several steps. The first step is to study the 5G NR standard and understand the different types of downlink transport channels used in the downlink directions.

The next step is to identify the parameters that need to be considered in the modelling process. These parameters include channel bandwidth, modulation scheme, channel coding rate, and channel characteristics.

The third step is to use MATLAB to model the downlink transport channel. This involves using functions provided by the MATLAB 5G Toolbox, such as **nrDLSCH**, **nrPDSCH**, and **nrOFDMModulate**, to simulate the transmission and reception of data over the downlink transport channel. The model should also include the effects of channel resource mapping (**nrResourceGrid)**, channel estimation **(nrChannelEstimate)** , and channel equalization **(nrEqualizeMMSE)**

The fourth step is to validate the model by comparing the simulation results with the expected results based on the 5G NR standard. This involves analysing metrics such as error rate, throughput, and spectral efficiency.

The final step is to use the validated model to study the performance of the 5G NR downlink transport channel under different scenarios. This includes varying the transport block size and modulation schemes and studying the impact on the performance of the downlink transport channel and compare the changes in the presence of channel improvement techniques like channel estimation, precoding, and equalization.

Overall, the methodology involves a combination of theoretical analysis and practical simulation using MATLAB to develop a comprehensive understanding of the behaviour and performance of the 5G NR downlink transport channel.

2. LITERATURE REVIEW

Transport channel modelling in 5G NR has been a subject of extensive research in recent years. Various studies have proposed different models and techniques for modelling the downlink and uplink transport channels in 5G NR. The most common methods include the use of channel coding, modulation schemes, and spatial diversity techniques to optimize the performance of the downlink transport channel.

One study [3] proposed a 5G NR transport channel model based on the 3GPP standard. The model considers the effects of multi-path fading, shadowing, and interference on the downlink transport channel, and it has been validated through simulations. Another study proposed a novel deep learning-based approach for modelling the 5G NR transport channel, which achieved better performance compared to traditional models.

Other studies [4] have focused on specific aspects of the transport channel modelling, such as the impact of channel estimation errors, mobility, and interference. The results of these studies have contributed to the development of more accurate and robust downlink transport channel models for 5G NR.

Overall, the previous studies [5] on transport channel modelling in 5G NR have provided valuable insights into the characteristics of the channel and the most effective methods for modelling it. However, there is still a need for further research to improve the accuracy and efficiency of the models and to address new challenges that may arise with the deployment of 5G networks.

The theoretical foundation for modelling the 5G NR Downlink transport channel using MATLAB involves understanding the key concepts of the 5G NR standard and the principles of wireless communication. The 5G NR standard is based on the Orthogonal Frequency Division Multiplexing (OFDM) modulation scheme, which is widely used in wireless communication systems. The OFDM scheme divides the available frequency band into smaller subcarriers and transmits data across multiple subcarriers simultaneously.

The 5G NR Downlink transport channel is responsible for transmitting user data between the User Equipment (UE) and the base station. It operates on top of the Physical layer, which is responsible for transmitting and receiving radio signals. The Downlink transport channel uses different modulation and coding schemes to ensure reliable data transmission.

To model the 5G NR Downlink transport channel using MATLAB, a fundamental understanding of the concepts of channel modelling, modulation and coding, and signal processing is essential. This involves knowledge of how to design and implement algorithms for channel estimation, equalization, and demodulation. Additionally, knowledge of MATLAB programming and 5G NR toolbox is necessary to implement the models accurately.

Overall, the theoretical foundation for modelling the 5G NR Downlink transport channel using MATLAB involves a comprehensive understanding of wireless communication principles, modulation and coding schemes, and signal processing algorithms.

3. KEY CONCEPTS

Key concepts for "5G NR Downlink transport channel modelling using MATLAB" may include the various components involved in the 5G NR downlink transport channel, such as the control and data channels, channel coding and modulation schemes, channel estimation and equalization, and the effects of fading and interference. Additionally, key concepts involve the use of MATLAB and 5G toolbox for simulation and modelling of the downlink transport channel, including the various functions and parameters used for channel modelling, such as the channel type, bandwidth, number of subcarriers, and path loss models. Understanding these key concepts is essential for developing an accurate and efficient model of the 5G NR downlink transport channel using MATLAB.

State-of-the-art techniques and tools for Downlink transport channel modelling in 5G NR using MATLAB include the use of the 5G Toolbox and the Communications Toolbox, which provide functions and blocks for modelling 5G NR waveforms, channels, and antennas. These tools are built on top of the MATLAB environment and allow for easy and efficient development and testing of 5G NR communication systems. Additionally, the use of channel models such as the 3GPP TR 38.901 3GPP TR 38.901 channel model and the COST 2100 channel model can provide realistic channel conditions for testing and evaluation of 5G NR systems. Furthermore, techniques such as MIMO (multiple-input, multiple-output) and beamforming can be used to improve the performance of 5G NR systems, and MATLAB provides built-in functions for simulating these techniques.

A detailed description of all the Key concepts and terms behind this paper is beyond the scope of this research, but almost all of them are well explained in chapter 9 of "5G NR: The Next Generation Wireless Access Technology." [6]

4. RESULTS

Table 1 lists input parameters and their values that we used for the purpose of this research. The input values used here for 5G NR downlink transport channel is as per standard 3GPP specifications **38.212**.[13].

SNR stands for Signal to noise ratio which is an important parameter to transmit signal. As the name suggests it is the measure of the strength of the signal relative to background noise.

S. NO.	PARAMETER	VALUE
1	SNR (in db)	7
2	No. of Slots	20
3	Transport block size	100
4	No. of Layers	20
5	DMRS length	2
6	No. of HARQ process	16
7	Code rate	490/1024
8	Maximum LDPC count	6
9	LDPC decoding algorithm	Normalized min-sum
10	No. of antennas	8
11	Delay profile	"TDL-C"
12	Modulation scheme	QPSK, 16QAM, 64QAM, 256QAM

Table 1. List of input parameters for simulation

On simulating using the parameters value listed in Table 1 and "**16QAM**" as Modulation scheme, we calculated the block error rate to be **0.42** and throughput to be **58 %.** The obtained values are as per the 3GPP specifications, and this confirms the validity of our model (Detailed MATLAB code is explained in Appendix 1).

The obtained plots of Channel Estimate, Signal Magnitude, Precoded signal (PDSCH), received signal (PDSCH), Transmitted waveform (PDSCH), AWGN (Noise), Received Waveform and Constellation diagram of Equalization is shown as follows.

4.1 Channel Estimate

Channel estimation [9] provides a representation of the channel effects per resource element (RE). The equalizer uses this information to compensate for the distortion introduced by the channel. We used Perfect channel estimation that assumes channel knowledge **nrPerfectChannelEstimate** and represents the channel conditions between the transmit antennas and receive antennas.

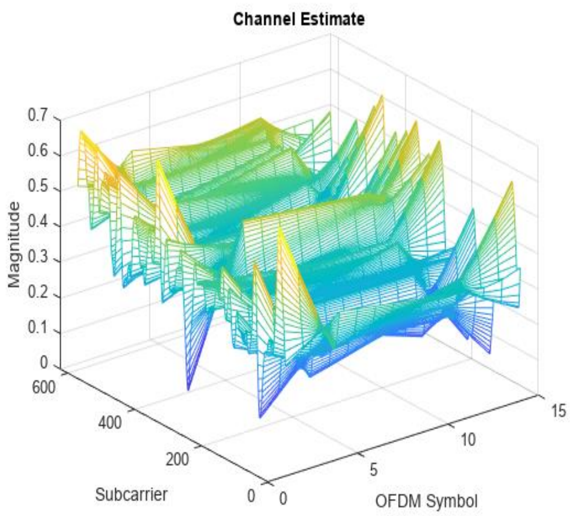


Figure 1. Channel Estimate

4.2 Magnitude

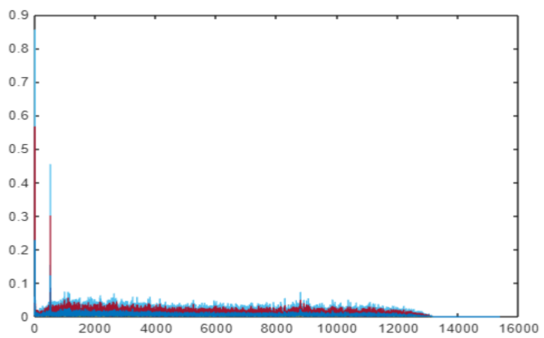


Figure 2. Magnitude plot

Figure 2 gives the overview of the magnitude obtained using perfect timing estimate.

4.3 Symbols Precoded (PDSCH)

The plot shown in Figure 3 gives an idea of the precoded symbols that are encoded in Physical downlink shared channel (PDSCH).

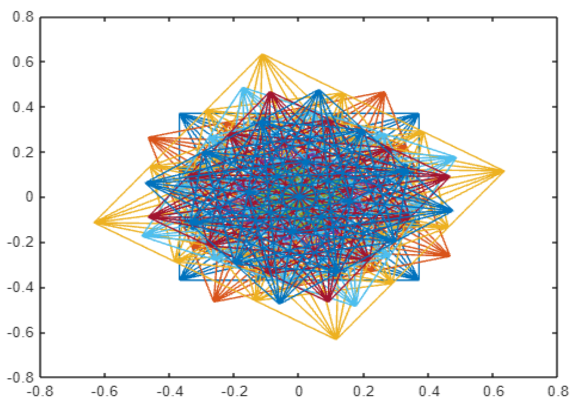


Figure 3. Precoded Symbols Plot

4.4 Signal Received (PDSCH)

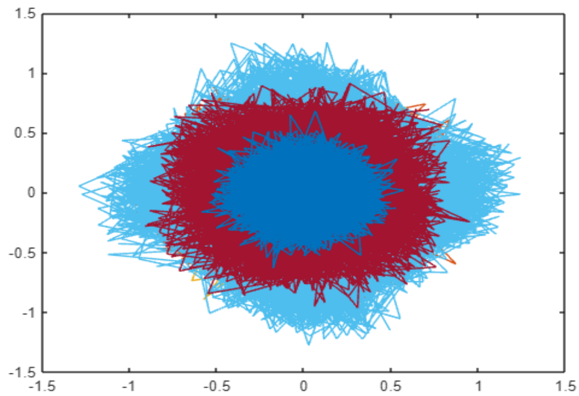


Figure 4. Received Signal Plot

The above plot in Figure 4 shows the received Signal after Equalization in Physical downlink shared channel (PDSCH).

4.5 Transmitted Waveform (PDSCH)

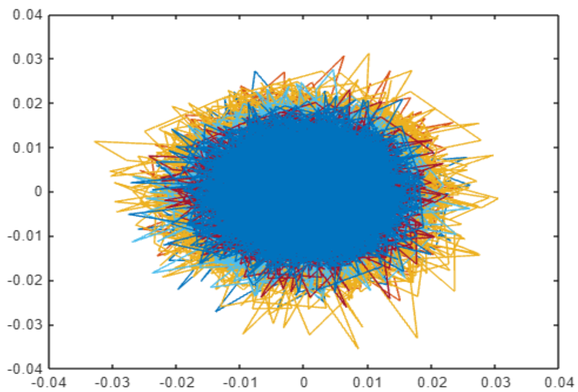


Figure 5. Transmitted Waveform Plot

The plot in Figure 5 depicts the waveform transmitted after OFDM modulation of the resource grid using **nrOFDMModulate** function.

4.6 AWGN (NOISE)

Additive White Gaussian Noise (AWGN)[10] which is type of noise that is commonly encountered in communication systems and signal processing. It is characterized by its flat frequency response, which means that the noise power is evenly distributed across all frequencies, and its statistical properties follow a Gaussian or normal distribution. It is called additive because it is added to the original signal and white because it has equal power density at all frequencies. Figure 6 shows the plot of AWGN.

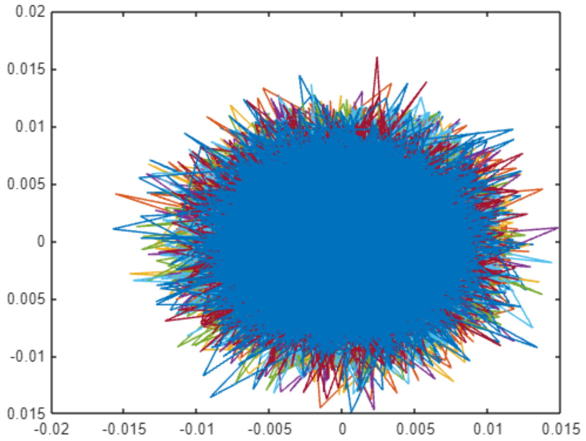


Figure 6. AWGN (noise) plot

4.7 Received Waveform

The plot shown in Figure 7 shows the received waveform after OFDM demodulation of the synchronised signal using the MATLAB function **nrOFDMDemodulate**.

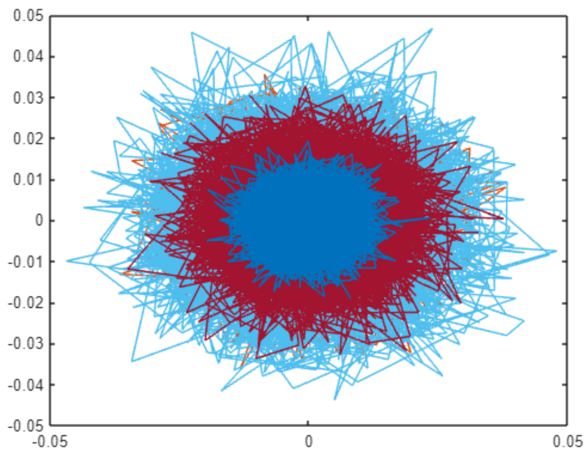


Figure 7. Received Waveform Plot

4.8 Equalization

The equalizer uses the channel estimate to compensate for the distortion introduced by the channel. The following plot is the constellation of the equalized symbols. This plot includes the constellation diagrams for all layers.

The plots shown in Figure 1-8 are obtained for parameter values defined in Table 1 and Modulation scheme of '16QAM'. Similar, plots can be obtained for different modulation scheme and parameter values using this model (refer Appendix 1 for detailed MATLAB code).

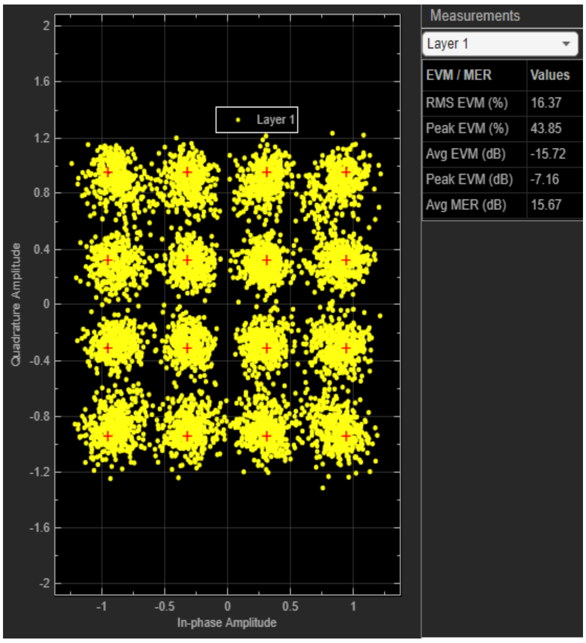


Figure 8. Equalization constellation diagram

5. INTERPRETATION

After developing a MATLAB-based model for the simulation of 5G NR downlink transport channel, we studied the channel for different parameters values including the transport block size and modulation schemes to understand the variation of the performance of the proposed model with different transport block size and modulation schemes.

It was observed that varying the code rate and Sub-carrier spacing had no change on the performance of the model. However, as expected, the change in Transport block size and Modulation Scheme did result in variation of the block error rate and throughput of the model.

The block error rate was as low as 0.377 for transport block size of 1000 blocks for '16QAM' modulation scheme and was as high as 1.0 for Transport block size of 10 blocks and modulation scheme of '64QAM' and '256QAM'. Also, the throughput became as low as 0% for modulation scheme of '64QAM' and '256QAM' at Transport block size of 10 blocks and it became as High as 62% for transport block size of 1000 blocks at '16QAM' modulation scheme.

Transport Block size	Block error ratio			Throughput		
	16QAM	64QAM	256QAM	16QAM	64QAM	256QAM
10	0.7	1.0	1.0	30	0	0
50	0.44	0.64	0.68	56	36	32
100	0.42	0.52	0.68	58	48	32
200	0.43	0.52	0.68	56.5	48	32
300	0.41	0.53	0.68	58.6	48	32
500	0.388	0.512	0.68	61.2	48.8	32
1000	0.377	0.504	0.672	62.3	49.6	32.8

Table 2. Block error ratio and Throughput for variable Transport Block size

The recorded data is as shown in the Table 2. It shows the block error ratio and throughput for different modulation scheme for various transport block size.

It can be inferred that as modulation scheme become complex from '16QAM' to '64QAM' to finally '256QAM', the block error rate increases. The block error rate also increases with decrease in size of Transport blocks. Similarly, the throughput decreases as the modulation scheme become complex from '16QAM' to '64QAM' to finally '256QAM'. The throughput also increases with increase in the Transport block size.

5.1 Channel Improvements

After we evaluated the performance of the model under various transport block size and modulation schemes, we tried to study the effects of Channel improvements techniques [6] on channel performance.

The recorded data is as shown in Table 3. It shows the block error ratio and throughput for different modulation scheme for various transport block sizes when we used various channel improvement techniques.

We inferred that Channel improvement techniques did work and reduce the block error rate to as low as 0 and Throughput as high as 100%. However, the effect of complexity of modulation scheme and Transport block size was still observed after these channel improvements techniques. The variation behaviour did not change at all with the previously obtained result.

Transport Block size	Block error ratio			Throughput		
	16QAM	64QAM	256QAM	16QAM	64QAM	256QAM
10	0	0	0.5	100	100	50
50	0	0	0.36	100	100	64
100	0	0	0.48	100	100	52
200	0	0	0.475	100	100	52.5
300	0	0.056	0.476	100	94.33	52.3
500	0	0.086	0.464	100	91.4	53.6
1000	0	0.126	0.472	100	87.4	52.8

Table 3. Block error ratio and Throughput for variable Transport Block size for improved channel mode

.6. CONCLUSION

In this paper, we developed a MATLAB-based model for simulating the 5G NR downlink transport channel. Also, we evaluated its performance under various transport block size and modulation schemes. Finally, after thorough simulations we were able to compare the performance of the model with a model that make use of channel improvement methods like equalization, precoding, and channel estimation.

We observed That as modulation scheme become complex from '16QAM' to '64QAM' to finally '256QAM', the block error rate increases which also increases with decrease in size of Transport blocks. Similarly, the throughput decreases as the modulation scheme become complex from '16QAM' to '64QAM' to finally '256QAM' whereas the throughput also increases with increase in the Transport block size.

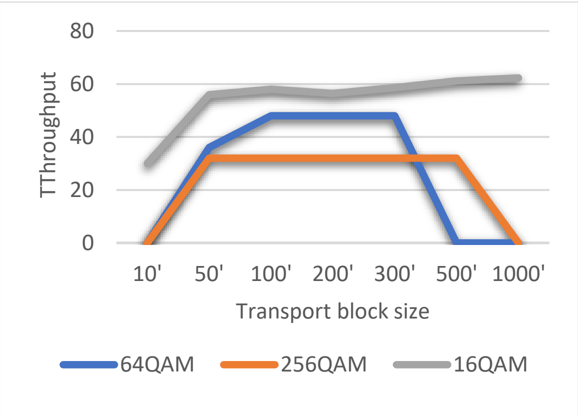


Figure 9. Throughput at variable transport block size

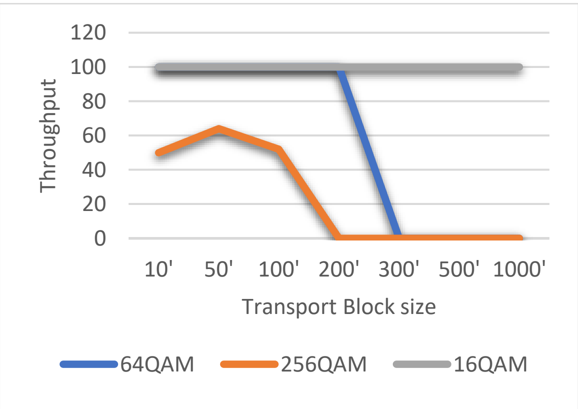


Figure 10. Throughput at variable transport block size with channel improvement techniques.

However, It is also worth noting that varying the code rate and Sub-carrier spacing had no change on the overall performance of the channel model

Also, after improving the channel using channel improvements techniques like Channel Estimation and Equalization, MIMO Precoding and Timing Synchronisation we observed that channel performance also improved by a considerable factor.

The charts shown in Figure 9 and 10 summarises Table 2 and 3 respectively. They compare the overall throughput of the model with a model that make use of channel improvement methods like equalization, precoding, and channel estimation at different modulation scheme and variable transport block size.

7. REFRENCES

[1] G. Sanfilippo, O. Galinina, S. Andreev, S. Pizzi and G. Araniti: "A concise review of 5g new radio capabilities for directional access at mmwave frequencies", Internet of Things Smart Spaces and Next Generation Networks and Systems, vol. 11118, pp. 340-354, (2018).

[2] Zaidi, A., Baldemair, R., Tullberg, H., Bjo¨rkegren, H., Sundstro¨m, L., Medbo, J., et al.: "Waveform and numerology to support 5G services and requirements", IEEE Communications Magazine, vol. 54, no. 11, pp. 90–98, (2016).

[3] Bhushan, N., Ji, T., Koymen, O., Smee, J., Soriaga, J., Subramanian, S., & Wei, Y: Industry perspectives: "5G air interface system design principles". IEEE Wireless Communications, vol 24, no. 5, pp. 6–8, (2017).

[4] Parkvall, S., Dahlman, E., Furuska¨r, A., & Frenne, M.: "5G NR: the next generation wireless access technology", IEEE Communications Standards Magazine, vol. 1, no. 4, pp. 24–30, (2017)

[5] M. R. Avendi, M. R. Javan, and H. Jafarkhani: "Uplink channel modelling and capacity analysis for 5G cellular networks", IEEE Trans. Wireless Commun., vol. 17, no. 8, pp. 5107-5119, (2018).

[6] Erik Dahlman, Stefan Parkvall, Johan Sköld: "Chapter 9- Transport channel processing; 5G NR: the Next Generation Wireless Access Technology", Academic Press, ISBN 9780128143230, pp. 153-182, (2018).

[7] Y. Wang, F. Zhu, X. You, and L. Liu: "Channel models for 5G communication systems", IEEE Wireless Commun., vol. 24, no. 4, pp. 136-143, (2017).

[8] Youngrok Jang: "Study on channel model for frequencies from 0.5 to 100 GHz", Technical Report, 3GPP (3rd Generation Partnership Project), TR 38.901, Version 14.0.0, (2017)

[9] S. Sun and T. S. Rappaport: "Path loss models for 5G millimeter wave propagation channels in urban microcells", IEEE Trans. Wireless Commun., vol. 16, no. 9, pp. 6171-6183, (2017).

[10] P. Fan, Y. Liu, F. Gao, Y. Cai, and X. Yin: "Uplink channel modelling for 5G heterogeneous networks", IEEE Commun. Mag., vol. 54, no. 6, pp. 62-68, (2016).

[11] Y. Wang, F. Gao, H. Zhang, W. Zhang, and L. Su: "Uplink channel modelling and capacity analysis for 5G cellular networks", IEEE Access, vol. 6, pp. 11759-11769, (2018).

[12] Rinaldi, F., Raschellà, A. & Pizzi, S.: "5G NR system design: a concise survey of key features and capabilities.", Wireless Network, vol. **27**, pp. 5173–5188, (2021).

[13] Yan Cheng: "5G; NR; Multiplexing and channel coding", Technical Specification Group Radio Access Network, 3GPP (3rd Generation Partnership Project), TS 38.212 version 16.2.0 Release 16, (2020).

[14] J. W. Wallace and M. A. Jensen: "Statistical characteristics of measured MIMO wireless channel data and comparison to conventional models", IEEE 54th Vehicular Technology Conference. VTC Fall 2001, vol.2, pp. 1078-1082, (2001).

[15] M. Lauridsen, L. C. Gimenez, I. Rodriguez, T. B. Sorensen, and P. Mogensen: "From lte to 5g for connected mobility", IEEE Communications Magazine, vol. 55, no. 3, pp. 156-162, (2017).

[16] Z. Ding, Y. Liu, J. Choi, Q. Sun, M. Elkashlan, I. Chih-Lin, et al.: "Application of non-orthogonal multiple access in lte and 5g networks, IEEE Communications Magazine", vol. 55, no. 2, pp. 185-191, (2017).

[17] A. Rahman, S. Arabi and R. Rab: "Feasibility and challenges of 5g network deployment in least developed countries (ldc)", Wireless Sensor Network, vol. 13, no. 1, pp. 1-16, (2021).