Project Based Learning Report

on

Submitted in the partial fulfillment of the requirements
For the Project based learning in **5G Architecture**
in
Electronics & Communication Engineering

PRN	Name of the Student
AASHISH KUMAR	PRN-2014111065
SAURAV KUMAR	PRN-2014111071
NISHAL KUMAR	PRN- 2014111072

Under the guidance of Course In-charge

Prof. S.V. Dhole

Department of Electronics & Communication Engineering

Bharati Vidyapeeth
(Deemed to be University)
College of Engineering,
Pune – 411043

Academic Year: 2023-24

Bharati Vidyapeeth
(Deemed to be University)
College of Engineering,
Pune – 411043

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the Project Based Learning report entitled. "Realistic Multiuser MIMO OFDM Simulations (Transmitter) & The performance over different models"
is work done by

PRN	Name of the Student
AASHISH KUMAR	PRN-2014111065
SAURAV KUMAR	PRN-2014111071
NISHAL KUMAR	PRN- 2014111072

in partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in "**5G Architecture"** of Bachelor of Technology Semester VIII , in ELECTRONICS AND COMMUNICATION ENGINEERING

Date:

Prof. S.V Dhole
Course In-charge

Dr. Arundhati A. Shinde
Professor & Head ECE

INDEX

Problem Statement:

Understanding how various channel models impact Multiuser MIMO OFDM system performance from the transmitter perspective is crucial for optimizing system design. However, this aspect remains underexplored. This study aims to investigate the performance variations of Multiuser MIMO OFDM systems across different channel models, focusing on transmitter strategies and signal processing techniques. By assessing the impact of channel models on system performance, this research seeks to provide insights for designing more robust and adaptive wireless communication networks.

Problem Solution:

This study seeks to delve into the impact of various channel models on Multiuser MIMO OFDM system performance, with a specific focus on transmitter strategies and signal processing techniques. By conducting comprehensive simulations and analyses, we aim to uncover the performance variations across different channel models and identify optimal approaches to mitigate channel-induced impairments. Through this research, we intend to provide valuable insights that can inform the design of more resilient and adaptable wireless communication networks, ultimately enhancing their efficiency and reliability in diverse real-world environments.

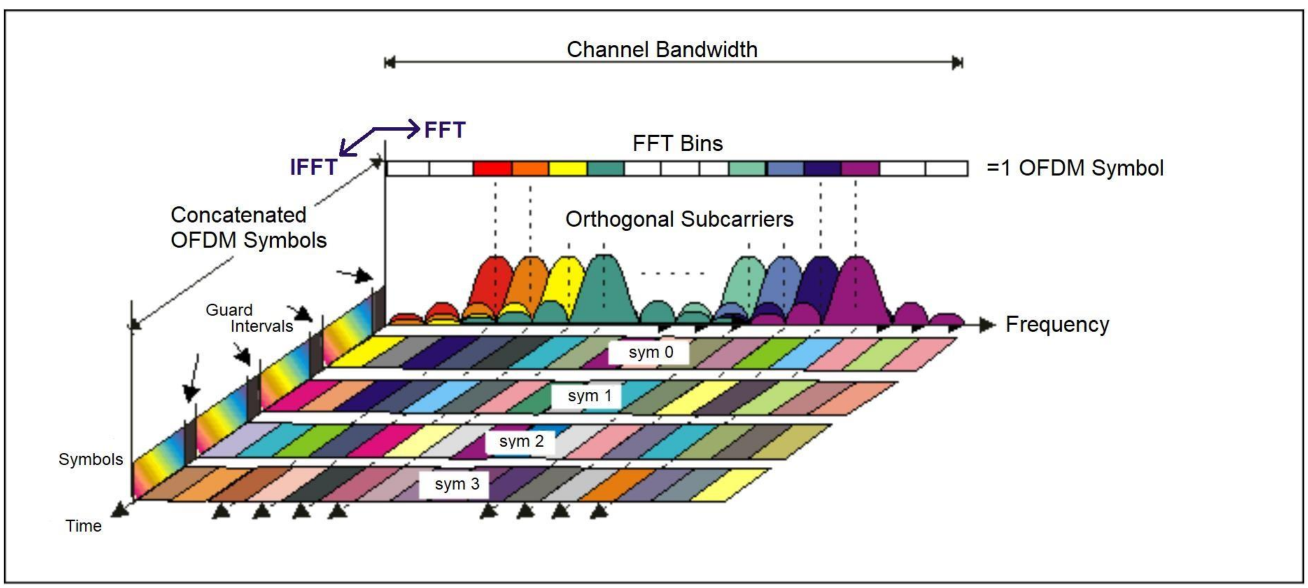
CHAPTER-1

INTRODUCTION

OFDM

Wireless communication systems based on multi-carrier modulation have become the state-of art method of broadband data transfer. OFDM is the most well-known and widely deployed scheme among them. Its favorable features such as signal generation using fast Fourier transform (FFT) algorithms, robustness against inter-symbol interference (ISI) and multipath fading, and easy application of MIMO techniques differentiate it from the rest [3, 4]. OFDM converts high-speed serial data stream into low-speed parallel data streams, also known as “subcarriers” that are orthogonal to each other. This property alleviates the impact of frequency selective fading due to multi-path. These sub-carriers are then modulated by a conventional modulation scheme such as quadrature phase shift keying (QPSK), quadrature amplitude modulation (QAM), etc., thus maintaining data rates similar to conventional single carrier modulation schemes in the same bandwidth. The main advantage of OFDM over single-carrier system is its resilience to frequency selective fading thus eliminating the use of expensive and complicated time domain equalizers which are commonly used in single-carrier systems. Moreover, OFDM is resilient to inter-symbol interference due to the presence of cyclic prefix before every symbol. Owing to these advantages, OFDM and its variants have been widely used in various wireless technologies such as IEEE 802.11 wireless local area network network(WLAN) , WiMax etc .

OFDM consists of large number of independent subcarriers which means that the resultant signal has significant PAPR . A high PAPR signal is susceptible to the non linear distortion cause due to the presence of HPA at transceivers , As a consequence the performance of the OFDM – based wireless system significantly degradation in the presence of non linear HPA .



Frequency-Time Representative of an OFDM signal

Multi-User MIMO

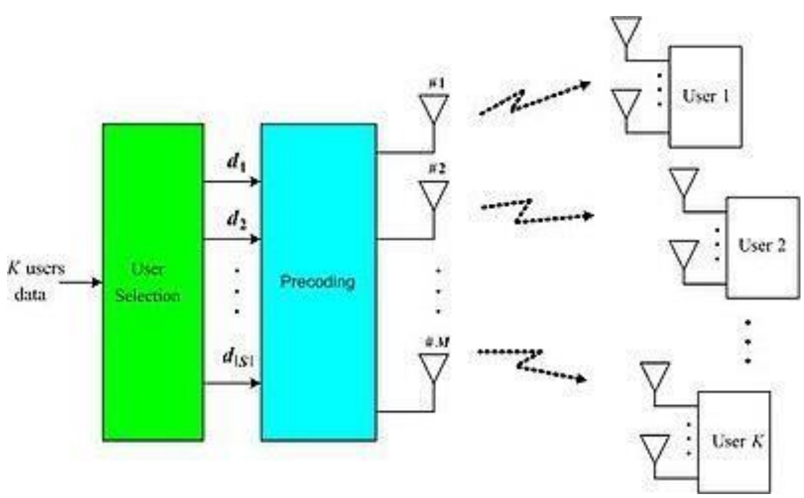
Multi-user MIMO, or MU-MIMO, is a wireless communication technology that uses multiple antennas to improve communication by creating multiple connections to the same device at the same time. MIMO is an acronym that stands for *multiple input, multiple output*.

MU-MIMO is commonly used in devices such as routers, and it works with mobile devices such as smartphones and laptops. The technology supports environments where multiple users access the same wireless network at once.

Typically, when multiple users connect to the same router, congestion will begin to build up as the router services the first device's requests while the other devices wait to be serviced. The amount of time each device waits is generally not long but can build up with enough devices. MU-MIMO helps relieve this potential congestion by creating multiple connections to a device at the same time, which increases network efficiency.

MU-MIMO takes advantage of multi-path, which is when a radio signal gets reflected and bounces around surrounding objects to be picked up by a receiver in a user's device at slightly different times and angles. MU-MIMO will typically have multiple antennas at the transmit end and one antenna at the receiving end of the signal.

MU-MIMO devices separate bandwidth into individual streams that share an equal connection. These streams typically divide as 2x2, 3x3, 4x4 or 8x8, which refers to the number of streams. The data streams directly to one device, which means an MU-MIMO router can only send and receive data from one device at a time .



Realistic Multiuser MIMO-OFDM

Realistic Multiuser MIMO OFDM (Multiple Input Multiple Output - Orthogonal Frequency Division Multiplexing) refers to a wireless communication system that combines multiple antenna elements at both the transmitter and receiver sides (MIMO) with orthogonal frequency division multiplexing modulation (OFDM) to achieve high data rates and spectral efficiency in multiuser environments while accounting for real-world factors and conditions.

Key components and features of realistic Multiuser MIMO OFDM systems:

1. Multiple Input Multiple Output (MIMO) : MIMO technology employs multiple antennas at both the transmitter and receiver to improve communication performance by exploiting spatial diversity and multiplexing gains. In a multiuser scenario, MIMO allows simultaneous communication with multiple users using spatial multiplexing techniques.

2. Orthogonal Frequency Division Multiplexing (OFDM) : OFDM is a modulation scheme that divides the available spectrum into multiple orthogonal subcarriers. By transmitting data simultaneously on these subcarriers, OFDM effectively mitigates the effects of frequency-selective fading and reduces inter-symbol interference (ISI). OFDM is widely used in modern wireless communication systems due to its robustness and spectral efficiency.

3. Realistic Channel Modeling: Real-world wireless channels are subject to various impairments such as multipath propagation, fading, shadowing, and interference. Realistic Multiuser MIMO OFDM systems incorporate accurate channel models to simulate these effects, ensuring that the performance of the system is evaluated under conditions similar to those encountered in practical deployments.

4. Multiuser Communication: In multiuser scenarios, multiple users share the same spectrum and communicate simultaneously with the base station (or access point) using multiple antennas. Multiuser MIMO techniques, such as beamforming, spatial multiplexing, and interference management, are employed to enable efficient communication with multiple users while maximizing spectral efficiency and minimizing interference.

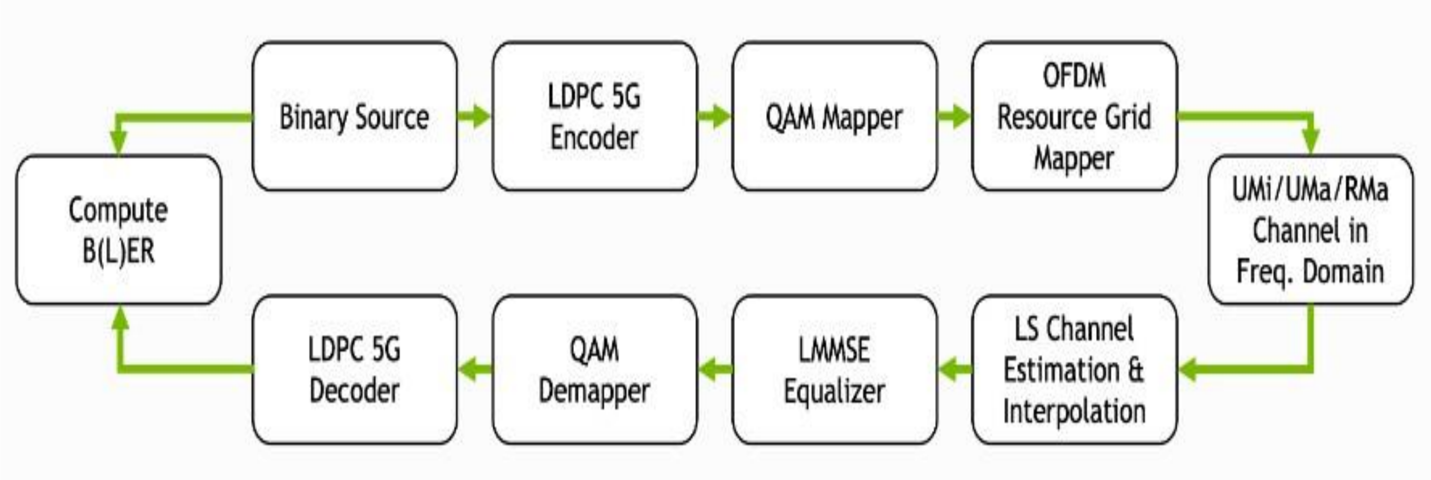
5. Resource Allocation: Efficient resource allocation is crucial in multiuser MIMO OFDM systems to optimize system performance and meet quality of service (QoS) requirements. This includes allocating subcarriers, transmit power, and antennas among different users based on channel conditions, user requirements, and fairness considerations.

6. Interference Management: Interference from other users sharing the same spectrum can degrade the performance of multiuser MIMO OFDM systems. Techniques such as interference cancellation, precoding, and spatial nulling are employed to mitigate interference and improve system throughput and reliability.

7. Performance Evaluation: Realistic Multiuser MIMO OFDM systems are evaluated based on various performance metrics, including bit error rate (BER), throughput, spectral efficiency, coverage, and user fairness. These metrics provide insights into the system's ability to support multiple users simultaneously while maintaining high data rates and reliable communication.

CHAPTER -2

Block Diagram & Working of System Model



5G LDPC FEC: Performs Forward Error Correction using Low-Density Parity-Check codes, enhancing the reliability of data transmission over the wireless channel by adding redundancy bits.

QAM modulation: Maps incoming data bits into symbols from a Quadrature Amplitude Modulation constellation, allowing for efficient data transmission by varying both amplitude and phase of the carrier signal.

OFDM resource grid with configurable pilot pattern: Organizes subcarriers into a grid structure, with flexibility to configure pilot symbols for channel estimation. Pilots aid in accurately estimating the channel response at different frequencies.

Multiple single-antenna transmitters and a multi-antenna receiver: Multiple transmitters each with a single antenna send data simultaneously to a receiver equipped with multiple antennas, enabling spatial multiplexing and diversity gain.

3GPP 38.901 UMi, UMa, and RMa channel models and antenna patterns: Provides standardized channel models and antenna patterns specified by 3GPP for various scenarios (urban macro, urban micro, rural macro), aiding in realistic simulation of wireless communication environments.

LS Channel estimation with nearest-neighbor interpolation as well as perfect CSI: Estimates the channel response using Least Squares (LS) method and interpolates between pilot symbols using nearest-neighbor interpolation. Perfect Channel State Information (CSI) assumes ideal knowledge of the channel conditions.

LMMSE MIMO equalization: Performs Linear Minimum Mean Square Error (LMMSE) equalization to mitigate the effects of channel distortions and noise in the received signal, improving data recovery in multi-antenna systems.

Path Loss Model in 5G Architecture :

The signal power in wireless communication systems is influenced by its surroundings; primarily, it will be affected by the path difference, operating frequency, and environmental effects. This makes it extremely challenging to plan any communication system that will provide better signal strength. Therefore, large-scale path loss models are considered to estimate the path loss at various frequencies, distances, and in various environments. In this paper, we considered UMi, UMa, and RMa environments to estimate the LOS and NLOS path loss for frequencies from 0.5 to 100 GHz. In the millimeter wave frequency range, a comparison is made between the path loss observed and the path loss models created by different standard organizations. The simulation results demonstrate that the 5GCM model is an optimized path loss model in the urban micro-environment, similarly 3GPP model is an optimized path loss model in UMa and RMa environments. These optimized models produce enhanced path loss compared to the other path loss models. These optimized models could be used by the service providers to enhance the quality of service in 5G wireless networks.

Some of these model are :

1. UMi (Urban Microcell):

Environment: UMi is primarily designed for dense urban microcell environments where buildings are closely spaced.

Key Features: It takes into account the effect of both large-scale and small-scale fading, including multipath reflections and diffractions.

Applications: Suitable for scenarios where small cells are deployed in urban areas, such as dense city centers or shopping districts.

Path Loss Characteristics : Generally exhibits higher path loss due to the presence of obstacles like buildings and street furniture, which cause significant signal attenuation and multipath propagation.

2. UMa (Urban Macrocell) :

Environment: UMa is tailored for urban macrocell environments characterized by fewer obstacles and larger cell sizes compared to UMi.

Key Features: It considers the impact of both line-of-sight (LOS) and non-line-of-sight (NLOS) propagation, as well as shadowing effects from buildings.

Applications: Suitable for urban areas where macrocell base stations are deployed to cover larger areas with fewer infrastructure elements compared to microcells.

Path Loss Characteristics: Typically exhibits lower path loss compared to UMi due to fewer obstacles, but still experiences attenuation from buildings and other structures.

3. RMa (Rural Macrocell) :

Environment :RMa is designed for rural macrocell environments characterized by sparse population density and wide open spaces.

Key Features : It considers terrain irregularities, such as hills and valleys, as well as vegetation attenuation.

Applications : Suitable for providing coverage in rural areas where the distance between base stations is relatively large and there are fewer obstacles compared to urban environments.

Path Loss Characteristics :Generally exhibits the lowest path loss among the three models due to fewer obstructions and lower propagation attenuation from vegetation compared to urban environments.

Software Used :

Google Colab : Google Colab, short for Google Colaboratory, is a free cloud-based platform provided by Google that allows users to write and execute Python code collaboratively. It's particularly popular among researchers, students, and data scientists for its ease of use, integration with Google Drive, and access to powerful hardware resources, including GPUs and TPUs (Tensor Processing Units).

Here are some key features and benefits of Google Colab:

1. Free Access : Google Colab is entirely free to use. It provides a Jupyter notebook environment without any cost, making it accessible to anyone with a Google account.

2. Cloud-Based : Being cloud-based, Google Colab does not require any setup or installation. Users can access it directly through a web browser, eliminating the need for local installations of Python or other dependencies.

3. Jupyter Notebook Integration : Google Colab supports Jupyter notebooks, which allow users to create and share documents containing live code, equations, visualizations, and explanatory text.

4. Collaboration : One of the key features of Google Colab is its ability to facilitate collaboration. Users can share their Colab notebooks just like they share Google Docs, enabling real-time collaboration with colleagues or classmates.

5. Integration with Google Drive : Google Colab seamlessly integrates with Google Drive, allowing users to save their notebooks directly to Google Drive, access datasets stored in Drive, and import libraries and files from Drive.

6. High-Performance Computing : Google Colab provides access to powerful hardware resources, including GPUs and TPUs, which can significantly accelerate computations for tasks such as machine learning model training.

7. Pre-installed Libraries : Google Colab comes with many popular Python libraries pre-installed, including TensorFlow, PyTorch, Keras, OpenCV, and more. Users can install additional libraries using pip commands.

8. Educational Resources : Google Colab is often used in educational settings due to its accessibility and ease of use. Many educational institutions leverage Colab for teaching programming, data science, and machine learning courses.

9. Support for Markdown: Google Colab supports Markdown cells, allowing users to add formatted text, headings, lists, and hyperlinks to their notebooks, making them more readable and informative.

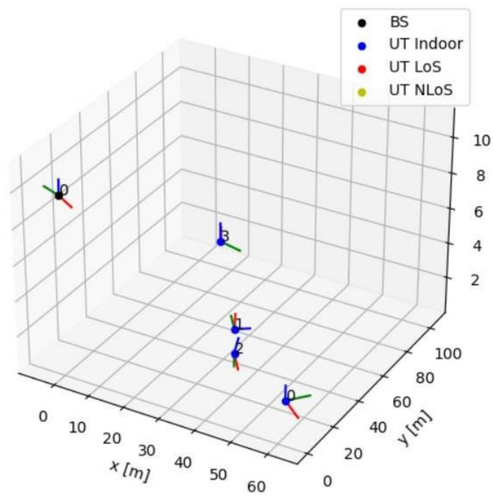
CHAPTER- 3

Program & Output:

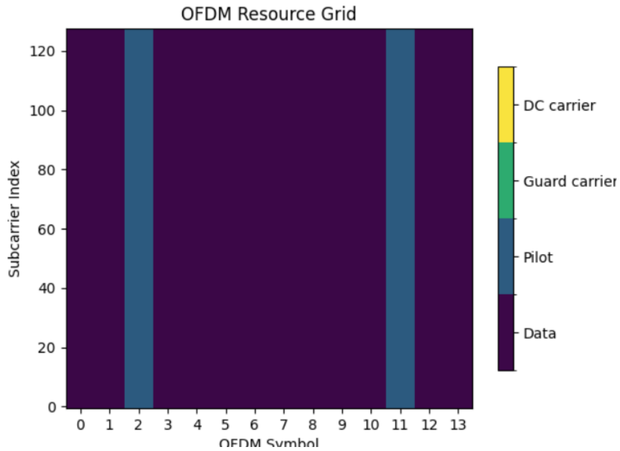
Code:

```
import os
gpu_num = 0 # Use "" to use the CPU
os.environ["CUDA_VISIBLE_DEVICES"] = f"{gpu_num}"
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
# Import Sionna
try:
    import sionna
except ImportError as e:
    # Install Sionna if package is not already installed
    import os
    os.system("pip install sionna")
    import sionna
import tensorflow as tf
gpus = tf.config.list_physical_devices('GPU')
if gpus:
    try:
        tf.config.experimental.set_memory_growth(gpus[0], True)
    except RuntimeError as e:
        print(e)
# Avoid warnings from TensorFlow
tf.get_logger().setLevel('ERROR')
tf.random.set_seed(1)
# Define the UT antenna array
ut_array = Antenna(polarization="single",
```

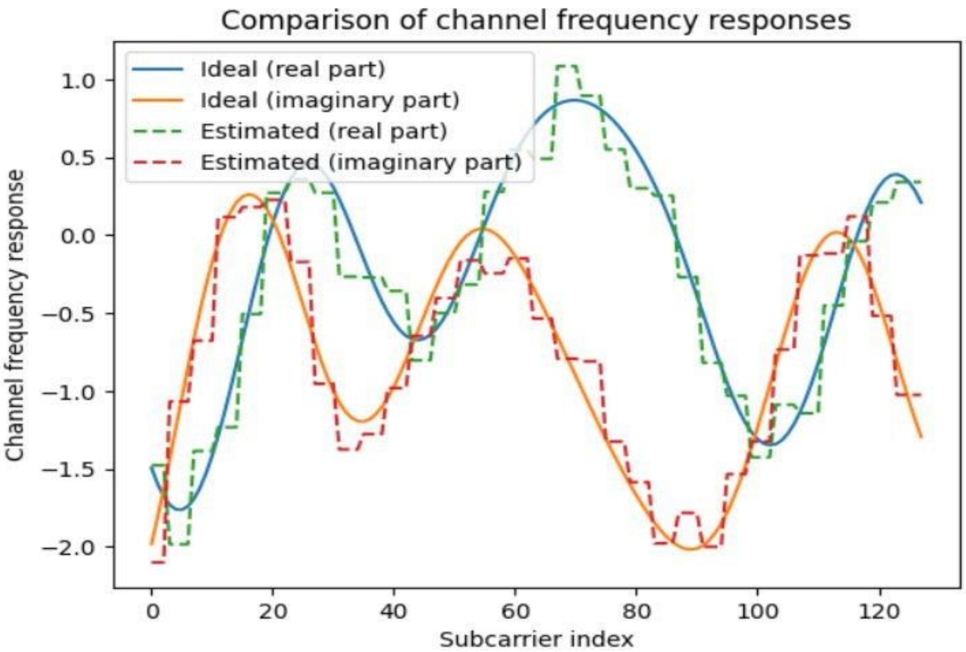
```
                polarization_type="V",
                antenna_pattern="omni",
                carrier_frequency=carrier_frequency)
# Define the BS antenna array
bs_array = AntennaArray(num_rows=1,
                    num_cols=4,
                    polarization="dual",
                    polarization_type="VH",
                    antenna_pattern="38.901",
                    carrier_frequency=carrier_frequency)
# Create channel model
channel_model = UMi(carrier_frequency=carrier_frequency,
                o2i_model="low",
                ut_array=ut_array,
                bs_array=bs_array,
                direction=direction,
                enable_pathloss=False,
                enable_shadow_fading=False)
# Generate the topology
topology = gen_topology(batch_size, num_ut, scenario)
# Set the topology
channel_model.set_topology(*topology)
# Visualize the topology
channel_model.show_topology()
```




```
rg = ResourceGrid(num_ofdm_symbols=14,
                  fft_size=128,
                  subcarrier_spacing=30e3,
                  num_tx=num_ut,
                  num_streams_per_tx=num_streams_per_tx,
                  cyclic_prefix_length=20,
                  pilot_pattern="kronecker",
                  pilot_ofdm_symbol_indices=[2,11])
rg.show()
```



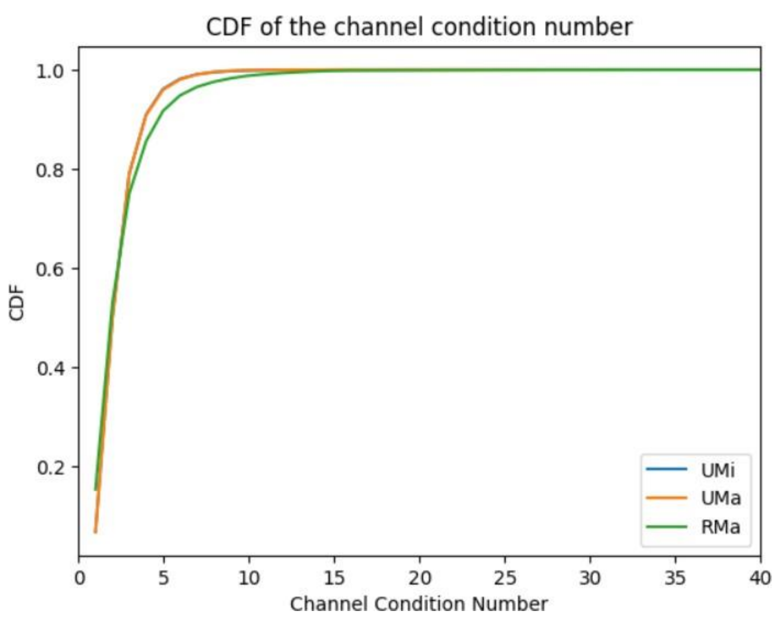

```
# In the example above, we assumed perfect CSI, i.e.,

# h_hat correpsond to the exact ideal channel frequency response.

h_perf = remove_nulled_scs(h_freq)[0,0,0,0,0,0]


# We now compute the LS channel estimate from the pilots.

h_est = h_hat[0,0,0,0,0,0]


plt.figure()

plt.plot(np.real(h_perf))

plt.plot(np.imag(h_perf))

plt.plot(np.real(h_est), "--")

plt.plot(np.imag(h_est), "--")

plt.xlabel("Subcarrier index")

plt.ylabel("Channel frequency response")

plt.legend(["Ideal (real part)", "Ideal (imaginary part)", "Estimated (real part)", "Estimated (imaginary part)"]);

plt.title("Comparison of channel frequency responses");
```




```
def cond_hist(scenario):
    """Generates a histogram of the channel condition numbers"""
# Setup a CIR generator
    if scenario == "umi":
        channel_model = UMi(carrier_frequency=carrier_frequency,
                              o2i_model="low",
                              ut_array=ut_array,
                              bs_array=bs_array,
                              direction="uplink",
                              enable_pathloss=False,
                              enable_shadow_fading=False)
    elif scenario == "uma":
        channel_model = UMa(carrier_frequency=carrier_frequency,
                              o2i_model="low",
                              ut_array=ut_array,
                              bs_array=bs_array,
```

```
                                direction="uplink",
                                enable_pathloss=False,
                                enable_shadow_fading=False)
    elif scenario == "rma":
        channel_model = RMa(carrier_frequency=carrier_frequency,
                                ut_array=ut_array,
                                bs_array=bs_array,
                                direction="uplink",
                                enable_pathloss=False,
                                enable_shadow_fading=False)

    topology = gen_topology(1024, num_ut, scenario)
   # Set the topology
   channel_model.set_topology(*topology)
    # Generate random CIR realizations
    # As we nned only a single sample in time, the sampling_frequency
    # does not matter.
    cir = channel_model(1, 1)
    # Compute the frequency response
    h = cir_to_ofdm_channel(frequencies, *cir, normalize=True)
    h = tf.squeeze(h)
    h = tf.transpose(h, [0,3,1,2])

    # Compute condition number
    c = np.reshape(np.linalg.cond(h), [-1])
  # Compute normalized histogram
```

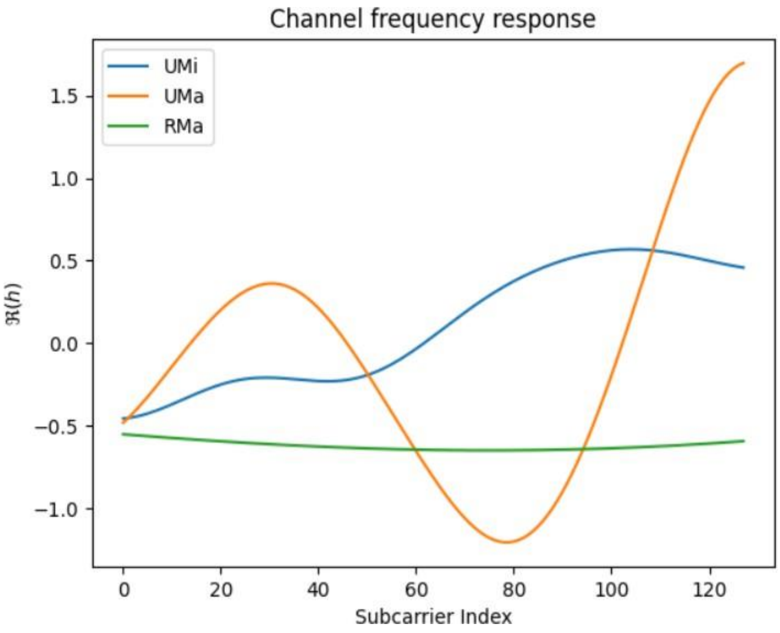
```
    hist, bins = np.histogram(c, 100, (1, 100))
    hist = hist/np.sum(hist)
    return bins[:-1], hist

plt.figure()
for cdl_model in ["umi", "uma", "rma"]:
    bins, hist = cond_hist(cdl_model)
    plt.plot(bins, np.cumsum(hist))
plt.xlim([0,40])
plt.legend(["UMi", "UMa", "RMa"]);
plt.xlabel("Channel Condition Number")
plt.ylabel("CDF")
plt.title("CDF of the channel condition number");
```




```
def freq_response(scenario):
    """Generates an example frequency response"""
    tf.random.set_seed(2)
```

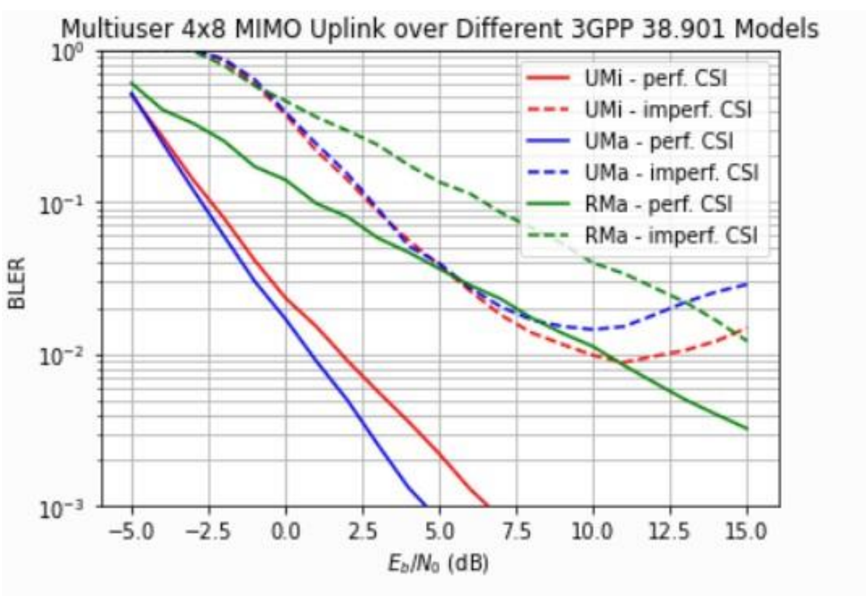
```
# Setup a CIR generator
if scenario == "umi":
    channel_model = UMi(carrier_frequency=carrier_frequency,
                        o2i_model="low",
                        ut_array=ut_array,
                        bs_array=bs_array,
                        direction="uplink",
                        enable_pathloss=False,
                        enable_shadow_fading=False)
elif scenario == "uma":
    channel_model = UMa(carrier_frequency=carrier_frequency,
                        o2i_model="low",
                        ut_array=ut_array,
                        bs_array=bs_array,
                        direction="uplink",
                        enable_pathloss=False,
                        enable_shadow_fading=False)
elif scenario == "rma":
    channel_model = RMa(carrier_frequency=carrier_frequency,
                        ut_array=ut_array,
                        bs_array=bs_array,
                        direction="uplink",
                        enable_pathloss=False,
                        enable_shadow_fading=False)
```

```
    topology = gen_topology(1, num_ut, scenario)

    # Set the topology
    channel_model.set_topology(*topology)

    # Generate random CIR realizations
    # As we nned only a single sample in time, the sampling_frequency
    # does not matter.
    cir = channel_model(1, 1)

    # Compute the frequency response
    h = cir_to_ofdm_channel(frequencies, *cir, normalize=True)
    h = tf.squeeze(h)

    return h[0,0]
plt.figure()
for cdl_model in ["umi", "uma", "rma"]:
    h = freq_response(cdl_model)
    plt.plot(np.real(h))
plt.legend(["UMi", "UMa", "RMa"]);
plt.xlabel("Subcarrier Index")
plt.ylabel(r"$\Re(h)$")
plt.title("Channel frequency response");
```




```
# Load results (uncomment to show saved results from the cell above)

plt.figure()

plt.xlabel(r"$E_b/N_0$ (dB)")

plt.ylabel("BLER")

plt.grid(which="both")

i=0

legend = []

for scenario in SIMS["scenario"]:

    for perfect_csi in SIMS["perfect_csi"]:

        if scenario=="umi":

            r = "r"

            t = "UMi"

        elif scenario=="uma":

            r = "b"

            t = "UMa"
```

```
        else:
            r = "g"
            t = "RMa"
        if perfect_csi:
            r += "-"
        else:
            r += "--"
plt.semilogy(SIMS["ebno_db"], SIMS["bler"][i], r);
        s = "{} - {} CSI".format(t,"perf." if perfect_csi else "imperf.")
        legend.append(s)
        i += 1
plt.legend(legend)
plt.ylim([1e-3, 1])
plt.title("Multiuser 4x8 MIMO Uplink over Different 3GPP 38.901 Models")
```



Conclusion :

Thus Due to the worse channel conditioning, the RMa model achieves the worst performance with perfect CSI. However, as a result of the smaller frequency selectivity, imperfect channel estimation only leads to a constant 5dB performace loss. For the UMI and UMa models, the used channel estimator with nearest-neighbor interpolation is not accurate enough so that the BER curves saturate at high SNR. This could, for example, be circumvented with another interpolation method (e.g., linear interpolation with time averaging) or a different pilot pattern .

Some applications of these models are :

1. Coverage Prediction
2. Site Selection and Deployment
3. Interference Analysis
4. Capacity Planning
5. Handover Optimization
6. Beamforming and MIMO
7. Network Planning and Optimization

Project Outcome:

Hence, we have achieved C04 (Implement the 5G wireless propagation channel models and MIMO.)

Appendix

1. https://ieeexplore.ieee.org/document/1371737
2. https://www.researchgate.net/publication/352039948_Performance_analysis_of_multiuser_MIMO_OFDM_systems_incorporating_feedback_delay_and_feedback_error
3. https://www.researchgate.net/publication/2998286_Multiuser_MIMO-OFDM_for_next-generation_wireless_systems