Project Based Learning Report
on

Analyze the impact of WLAN fading SISO/MIMO channels on the received waveforms using the WLAN Toolbox

Submitted in the partial fulfillment of the requirements For
the Project based learning in Cellular Technology & 4G

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Name of the student	**PRN**
Anshuman	2114110433
Ayush Ghosh	2114110449
Mayank Mankar	2114110488

Under the guidance of Course Incharge

Prof. Dr. Tanuja S. Dhope

Department of Electronics & Communication Engineering

Bharati Vidyapeeth

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

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

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the Project Based Learning report entitled, "Analyze the impact of WLAN fading SISO/MIMO channels on the received waveforms using the WLAN toolbox in Matlab" is a bonafied work done by

Name of the student	PRN
Anshuman	**2114110433**
Ayush Ghosh	**2114110449**
Mayank Mankar	**2114110488**

in partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in Information Theory And Coding Course of Bachelor of Technology Semester VI, in Electronics and Communication Branch.

Date:

Prof. Dr. Tanuja .S. Dhope	Prof. Dr A.A.Shinde
Course Incharge	Head Of Department

INDEX

Chapter no.	Title	Page No.
1.	Problem Statement with Solution	2
2.	Project Description	3
3.	Software used	4
5.	Simulation and result	6
7.	Project Outcome	8-10
8.	Conclusion	11
9.	References	12

CHAPTER 1
PROBLEM STATEMENT

Question:-

How does WLAN fading affect the received waveforms in both Single-Input Single-Output (SISO) and Multiple-Input Multiple-Output (MIMO) channels, and how can the WLAN Toolbox in MATLAB be used to analyze these effects

Solution:-

WLAN fading, caused by factors such as multipath propagation, interference, and environmental conditions, can significantly impact the quality and integrity of received waveforms in wireless communication systems. The WLAN Toolbox in MATLAB provides tools for simulating and analyzing these effects. Let's delve into how WLAN fading affects SISO and MIMO channels and how we can analyze these impacts using MATLAB:

1. WLAN Fading Effects on SISO Channels:
In SISO channels, where there is only one transmit and one receive antenna, WLAN fading can cause signal attenuation, multipath distortion, and variations in signal strength over time and frequency. These effects result in fluctuations in the received signal's amplitude and phase, leading to potential degradation in communication quality.

2. WLAN Fading Effects on MIMO Channels:
MIMO channels, with multiple transmit and receive antennas, are inherently more robust to fading compared to SISO channels. However, WLAN fading can still impact MIMO systems by introducing spatial correlation between antennas, causing fading correlation and reducing the diversity gain. This can result in variations in the received signal's quality across different spatial streams.

3. Analyzing WLAN Fading Effects using MATLAB:
To analyze the impact of WLAN fading on received waveforms, we can use the WLAN Toolbox in MATLAB. This toolbox allows us to simulate fading scenarios by modeling channel characteristics such as path loss, shadowing, and multipath fading. By generating WLAN waveforms for both SISO and MIMO channels and applying fading effects using appropriate channel models, we can observe how the received waveforms are affected.

4. Observations and Analysis:
Through MATLAB simulations, we can observe changes in the received waveforms, such as amplitude variations, phase shifts, and fluctuations in signal strength. We can compare the impact of fading on SISO and MIMO channels, evaluating factors like signal quality, robustness, and diversity gain. Additionally, we can analyze performance metrics such as signal-to-noise ratio (SNR) and bit error rate (BER) to quantify the effects of fading on WLAN system performance.

In summary, WLAN fading significantly influences the received waveforms in both SISO and MIMO channels, impacting signal quality and system performance. By leveraging the capabilities of the WLAN Toolbox in MATLAB, we can simulate and analyze these effects better understand how fading affects WLAN communication systems and devise strategies to mitigate its impact.

CHAPTER 2:-
PROJECT DESCRIPTION

MATLAB code is aimed at addressing the problem statement . Let's break down the project:

Introduction:

In the modern era of wireless communication, understanding the behavior of wireless local area network (WLAN) channels is crucial for designing robust communication systems. WLAN channels are subject to various impairments, including fading, which can significantly affect the quality of received signals. In this project, we aim to investigate the impact of fading in WLAN channels on received waveforms using MATLAB's WLAN Toolbox.

Objectives:

To simulate WLAN fading channels, both single-input single-output (SISO) and multiple-input multiple-output (MIMO), using MATLAB.

To generate WLAN waveforms under different fading conditions.

To analyze the effects of fading on received waveforms in terms of signal quality metrics such as signal-to-noise ratio (SNR), bit error rate (BER), and throughput.

To compare the performance of SISO and MIMO systems in WLAN environments under fading conditions.

To explore techniques for mitigating the effects of fading in WLAN systems.

Methodology:

Simulate SISO and MIMO WLAN fading channels using MATLAB's WLAN Toolbox.

Generate WLAN waveforms using various modulation schemes (e.g., OFDM, QAM).

Implement fading models such as Rayleigh, Rician, and Nakagami to simulate realistic channel conditions.

Analyze received waveforms under different fading scenarios by measuring SNR, BER, and throughput.

Compare the performance of SISO and MIMO systems in terms of robustness to fading and achievable data rates.

Investigate the effectiveness of techniques like diversity combining and beamforming in mitigating fading effects in WLAN systems.

Expected Outcomes:

A thorough understanding of the impact of fading on WLAN channels and its implications for system performance.

Insight into the advantages and limitations of SISO and MIMO configurations in WLAN environments.

Identification of effective strategies for combating fading effects in WLAN systems.

Quantitative analysis and comparison of different fading models and their applicability to WLAN scenarios.

Recommendations for optimizing WLAN system design to enhance reliability and throughput in fading environments.

Significance:

The findings of this project can inform the design and deployment of WLAN systems in real-world environments, where fading is a common phenomenon.

Understanding the behavior of WLAN channels under fading conditions is essential for ensuring reliable and high-performance wireless communication.

The project contributes to the body of knowledge in wireless communications and provides valuable insights for researchers, engineers, and practitioners working in the field of WLAN technology.

Applications:-

1. **Network Optimization**: Understanding network performance metrics such as bandwidth, propagation delay, and throughput allows network administrators to identify bottlenecks and optimize network infrastructure for improved efficiency and reliability.

2. **Troubleshooting**: When users experience slow network speeds or connectivity issues, analyzing network performance metrics can help pinpoint the root cause of the problem, whether it's related to bandwidth limitations, network congestion, or other factors.

3. **Capacity Planning**: By monitoring network performance over time, organizations can anticipate future capacity requirements and plan upgrades or expansions to accommodate growing demands without compromising performance.

4. **Quality of Service (QoS) Management**: Assessing parameters like latency and jitter is crucial for ensuring consistent and reliable network performance, particularly in applications where real-time data transmission is critical, such as voice and video communication.

5. **Service Level Agreement (SLA) Compliance**: For service providers, measuring network performance against predefined SLA parameters is essential for maintaining customer satisfaction and meeting contractual obligations.

6. **Security Analysis**: Abnormalities in network performance metrics can sometimes indicate security threats such as denial-of-service (DoS) attacks or unauthorized access attempts, prompting proactive security measures to safeguard network integrity.

7. **Research and Development**: Engineers and researchers in the field of networking can use such tools to conduct experiments, validate theoretical models, and explore new technologies aimed at enhancing network performance and efficiency.

CHAPTER 4
SOFTWARE USED

MATLAB:

1. Programming Language:

MATLAB provides an easy-to-understand programming language with syntax that is conducive to mathematical and scientific computing.
It supports various programming paradigms, including procedural, functional, and object-oriented programming.
MATLAB code is interpreted, which allows for rapid prototyping and quick development cycles.

2. Numerical Computing:

MATLAB's core strength lies in its numerical computing capabilities.
It provides built-in functions and libraries for performing a wide range of mathematical operations, including linear algebra, optimization, statistics, signal processing, and differential equations solving.
MATLAB's matrix-based approach to computation makes it particularly efficient for handling large-scale numerical problems.

3. Data Analysis and Visualization:

MATLAB offers powerful tools for importing, manipulating, analyzing, and visualizing data.
It supports importing data from various file formats, such as spreadsheets, text files, and databases.
MATLAB's plotting functions enable users to create publication-quality plots, charts, graphs, and animations to visualize data effectively.

4. Toolboxes:

MATLAB's functionality can be extended using toolboxes, which are collections of specialized functions and algorithms for specific application areas.
MathWorks provides a wide range of toolboxes covering domains such as image processing, control systems, machine learning, signal processing, and more.
These toolboxes allow users to access advanced features and algorithms tailored to their specific needs.

5. Interactivity:

MATLAB's interactive environment enables users to execute commands and scripts, visualize data, and explore algorithms in real-time.
The command window allows for quick experimentation and exploration of MATLAB functions and features.
MATLAB also supports the creation of graphical user interfaces (GUIs) for building custom interactive applications.

6. Deployment:

MATLAB code can be deployed to various platforms and formats for sharing or integration into larger systems.

MATLAB Compiler allows users to convert MATLAB code into standalone executables or shared libraries for deployment on systems without MATLAB installed.

MATLAB code can also be deployed to web applications, hardware devices, and cloud platforms.

7. Integration:

MATLAB can be integrated with other programming languages and tools, enabling seamless interoperability.

MATLAB supports calling external functions and libraries written in languages such as C/C++, Java, Python, and .NET.

MATLAB also provides APIs and interfaces for interacting with external software, databases, and hardware devices.

8. Community and Support:

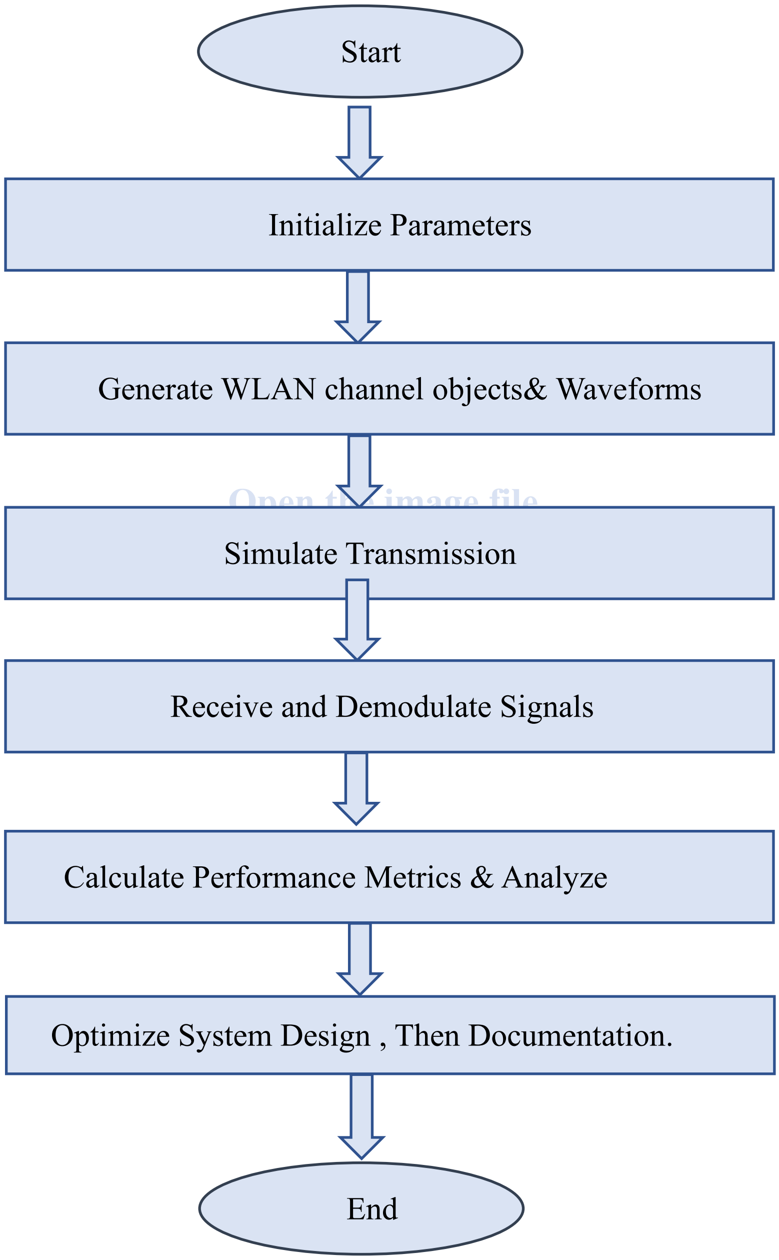
MATLAB has a large and active user community, with resources such as documentation, forums, and online tutorials available for learning and troubleshooting.

MathWorks offers comprehensive technical support, training courses, and consulting services for MATLAB users.

MATLAB Central, MathWorks' online community platform, provides a wealth of user-contributed code, examples, and resources for MATLAB users.

Overall, MATLAB is a versatile and powerful platform that offers a wide range of capabilities for numerical computing, data analysis, visualization, and application development. Its ease of use, extensive functionality, and strong community support make it a popular choice for researchers, engineers, educators, and students worldwide.

CHAPTER 4
FLOWCHART



CHAPTER 5
ALGORITHM

Initialize Parameters:
Define parameters such as simulation duration, channel model (Rayleigh, Rician, Nakagami), modulation scheme, number of transmit and receive antennas for MIMO, etc.

Generate WLAN Channel:
Use MATLAB's WLAN Toolbox to create WLAN fading channel objects for both SISO and MIMO configurations.
Configure the channel objects with appropriate parameters like center frequency, bandwidth, and channel model parameters.

Generate WLAN Waveforms:
Generate WLAN waveforms using MATLAB's WLAN Toolbox functions.
Specify modulation scheme, data rate, and other waveform parameters.

Simulate Transmission:
Transmit the generated waveforms through the WLAN channels.
Account for channel impairments such as fading, noise, and interference.

Receive and Demodulate Signals:
Receive the transmitted signals at the receiver end.
Apply appropriate demodulation techniques to extract the transmitted data symbols.

Calculate Performance Metrics:
Measure signal quality metrics such as signal-to-noise ratio (SNR), bit error rate (BER), and throughput.
Compare the performance of SISO and MIMO systems under different fading conditions.

Analyze Results:
Analyze the impact of fading on received waveforms and system performance.
Evaluate the effectiveness of SISO and MIMO configurations in mitigating fading effects.
Identify trends and patterns in the data.

Optimize System Design:
Based on the analysis, optimize WLAN system parameters such as antenna configuration, modulation scheme, and channel coding to improve performance in fading channels.

Repeat and Validate:
Repeat the simulation with different parameter settings and channel conditions to validate the results.
Ensure consistency and reliability of the findings.

CHAPTER 6
PROGRAM & OUTPUT

Code:-

Pass S1G Waveform Through TGah SISO Channel

```
bits = randi([0 1],1000,1);
s1g = wlanS1GConfig(APEPLength=1000);
preChS1G = wlanWaveformGenerator(bits,s1g);
cbw = s1g.ChannelBandwidth;
fs = 2e6; % Channel model sampling frequency equals the channel bandwidth
tgahChan = wlanTGahChannel('SampleRate',fs,'ChannelBandwidth',cbw, ...
    'LargeScaleFadingEffect','Pathloss and shadowing', ...
    'DelayProfile','Model-D');
preChSigPwr_dB = 20*log10(mean(abs(preChS1G)));
sigPwr = 10^((preChSigPwr_dB-tgahChan.info.Pathloss)/10);

chNoise = comm.AWGNChannel('NoiseMethod','Signal to noise ratio (SNR)',...
    'SNR',10,'SignalPower', sigPwr);
postChS1G = chNoise(tgahChan(preChS1G));
rxNoise = comm.AWGNChannel('NoiseMethod','Variance', ...
    'VarianceSource','Input port');
nVar = 10^((-228.6 + 10*log10(290) + 10*log10(fs) + 9)/10);

rxS1G = rxNoise(postChS1G,nVar);
title = '2 MHz S1G Waveform Before and After TGah Channel';
saScope = spectrumAnalyzer(SampleRate=fs,ShowLegend=true,...
    AveragingMethod='exponential',ForgettingFactor=0.99,Title=title,...
    ChannelNames={'Before','After'});
saScope([preChS1G,rxS1G])
```

Pass VHT Waveform Through TGac SISO Channel

```
bits = randi([0 1],1000,1);
vht = wlanVHTConfig;
preChVHT = wlanWaveformGenerator(bits,vht);
cbw = vht.ChannelBandwidth;
fs = 80e6; % Channel model sampling frequency equals the channel bandwidth
tgacChan = wlanTGacChannel('SampleRate',fs,'ChannelBandwidth',cbw, ...
    'LargeScaleFadingEffect','Pathloss and shadowing', ...
    'DelayProfile','Model-D');
preChSigPwr_dB = 20*log10(mean(abs(preChVHT)));
sigPwr = 10^((preChSigPwr_dB-tgacChan.info.Pathloss)/10);

chNoise = comm.AWGNChannel('NoiseMethod','Signal to noise ratio (SNR)',...
    'SNR',10,'SignalPower', sigPwr);
```

```
postChVHT = chNoise(tgacChan(preChVHT));
rxNoise = comm.AWGNChannel('NoiseMethod','Variance', ...
    'VarianceSource','Input port');
nVar = 10^((-228.6 + 10*log10(290) + 10*log10(fs) + 9)/10);

rxVHT = rxNoise(postChVHT,nVar);
title = '80 MHz VHT Waveform Before and After TGac Channel';
saScope = spectrumAnalyzer(SampleRate=fs,ShowLegend=true,...
    AveragingMethod='exponential',ForgettingFactor=0.99,Title=title,...
    ChannelNames={'Before','After'});
saScope([preChVHT,rxVHT])
```

Pass HT Waveform Through TGn SISO Channel

```
bits = randi([0 1],1000,1);
ht = wlanHTConfig;
preChHT = wlanWaveformGenerator(bits,ht);
fs = 20e6; % Channel model sampling frequency equals the channel bandwidth
tgnChan = wlanTGnChannel('SampleRate',fs,'LargeScaleFadingEffect', ...
    'Pathloss and shadowing','DelayProfile','Model-F');
postChHT = awgn(tgnChan(preChHT),10,'measured');
rxNoise = comm.AWGNChannel('NoiseMethod','Variance', ...
    'VarianceSource','Input port');
nVar = 10^((-228.6 + 10*log10(290) + 10*log10(fs) + 9)/10);

rxHT = rxNoise(postChHT, nVar);
title = '20 MHz HT Waveform Before and After TGn Channel';
saScope = spectrumAnalyzer(SampleRate=fs,ShowLegend=true,...
    AveragingMethod='exponential',ForgettingFactor=0.99,Title=title,...
    ChannelNames={'Before','After'});
saScope([preChHT,postChHT])
```

Pass Non-HT Waveform Through 802.11g Channel

```
ssbits = randi([0 1],1000,1);
nht = wlanNonHTConfig;
preChNonHT = wlanWaveformGenerator(bits,nht);
dist = 3;
fc = 2.4e9;
pathLoss = 10^(-log10(4*pi*dist*(fc/3e8)));
fs = 20e6; % Channel model sampling frequency equals the channel bandwidth
maxDoppShift = 3;
trms = 2/fs;
ch802 =
comm.RayleighChannel('SampleRate',fs,'MaximumDopplerShift',maxDoppShift,'PathDelays',t
rms);
postChNonHT = awgn(ch802(preChNonHT),10,'measured');
rxNoise = comm.AWGNChannel('NoiseMethod','Variance', ...
    'VarianceSource','Input port');
nVar = 10^((-228.6 + 10*log10(290) + 10*log10(fs) + 9)/10);
```

```
rxNonHT = rxNoise(postChNonHT, nVar)* pathLoss;
title = '20 MHz Non-HT Waveform Before and After 802.11g Channel';
saScope = spectrumAnalyzer(SampleRate=fs,ShowLegend=true,...
   AveragingMethod='exponential',ForgettingFactor=0.99,Title=title,...
   ChannelNames={'Before','After'});
saScope([preChNonHT,rxNonHT])
```

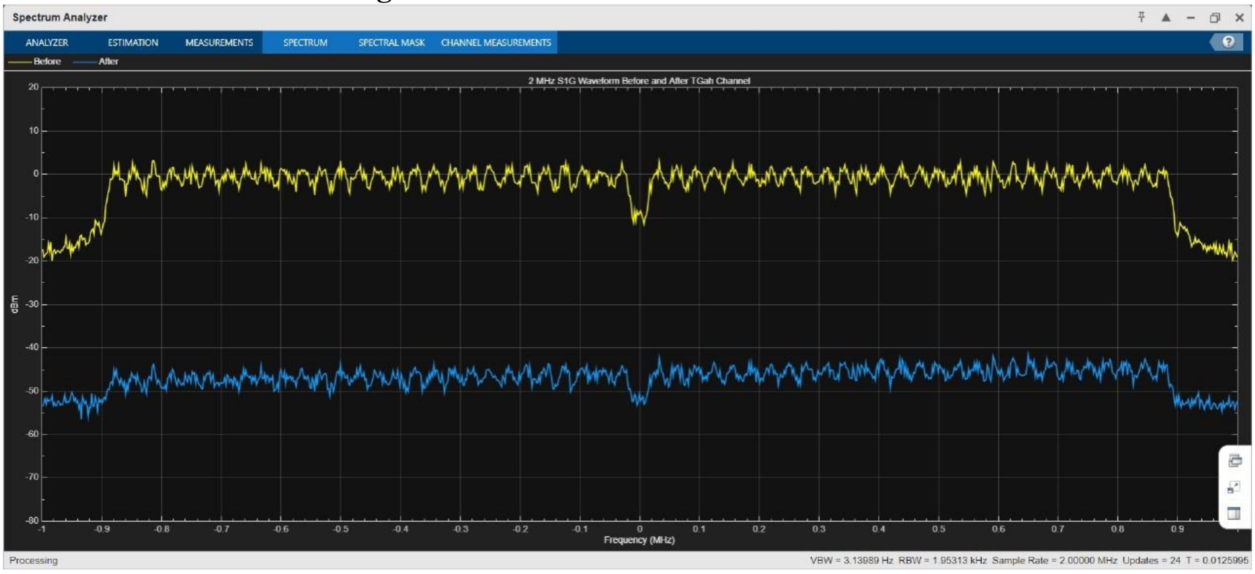
Pass VHT Waveform Through TGac MIMO Channel

```
ssbits = randi([0 1],1000,1);
ntx = 4;
nsts = 3;
nrx = 3;
vht = wlanVHTConfig('NumTransmitAntennas',ntx, ...
   'NumSpaceTimeStreams',nsts,'SpatialMapping','Hadamard');
preChVHT = wlanWaveformGenerator(bits,vht);
cbw = vht.ChannelBandwidth;
fs = 80e6; % Channel model sampling frequency equals the channel bandwidth
tgacChan = wlanTGacChannel('SampleRate',fs,'ChannelBandwidth',cbw,...
   'NumTransmitAntennas',ntx,'NumReceiveAntennas',nrx);
tgacChan.LargeScaleFadingEffect = 'None';
postChVHT = awgn(tgacChan(preChVHT),10,'measured');
rxNoise = comm.AWGNChannel('NoiseMethod','Variance', ...
   'VarianceSource','Input port');
nVar = 10^((-228.6 + 10*log10(290) + 10*log10(fs) + 9)/10);

rxVHT = rxNoise(postChVHT,nVar);
title = '80 MHz VHT 4x3 MIMO Waveform After TGac Channel';
saScope = spectrumAnalyzer(SampleRate=fs,ShowLegend=true,...
   AveragingMethod='exponential',ForgettingFactor=0.99,Title=title,...
   ChannelNames={'RX1','RX2','RX3'});
saScope(rxVHT)
```

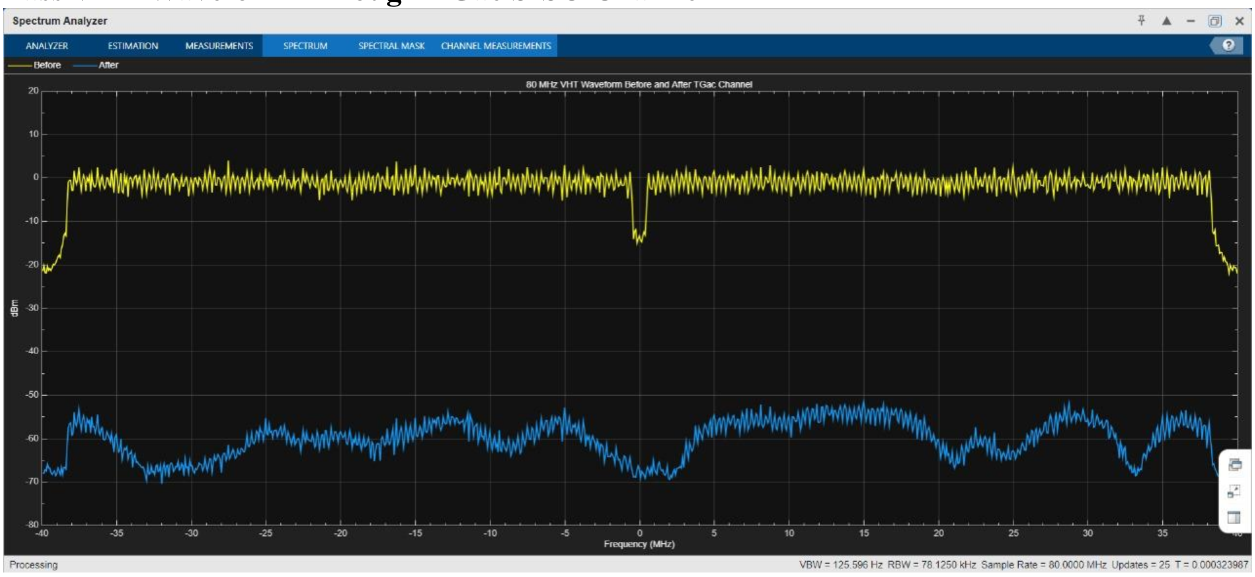
OUTPUT:-

Pass S1G Waveform Through TGah SISO Channel



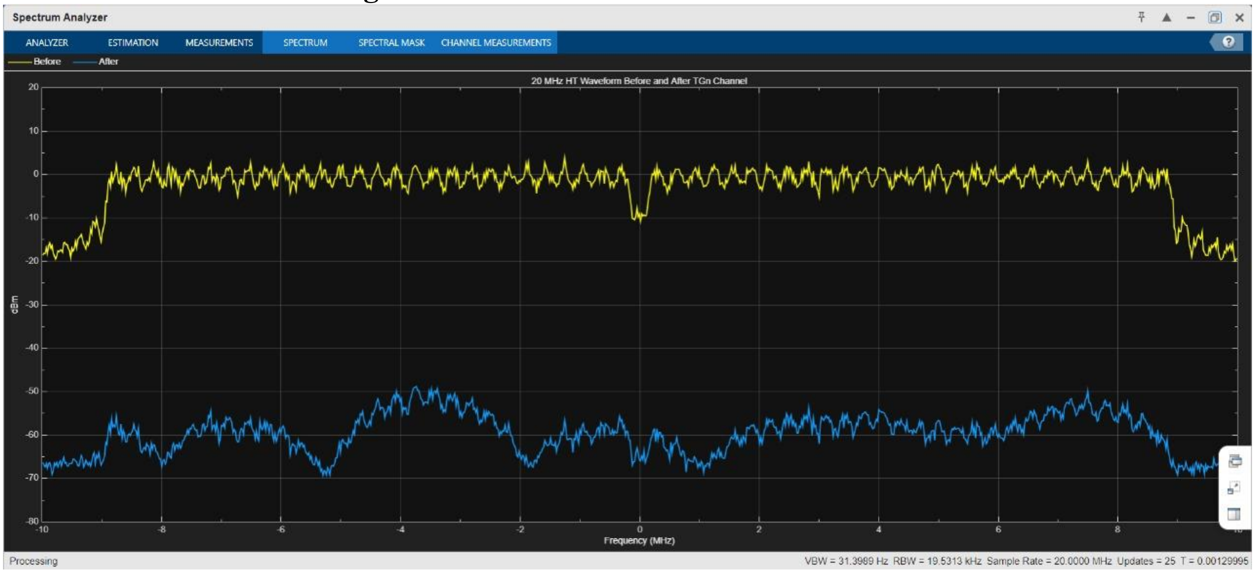
Path loss accounts for the roughly 50 dB of separation between the waveform before and after it passes through the TGah channel. The path loss results from the default transmitter-to-receiver distance of 3 meters, and from shadowing effects. The signal level variation shows the frequency selectivity of the delay profile across the frequency spectrum.

Pass VHT Waveform Through TGac SISO Channel



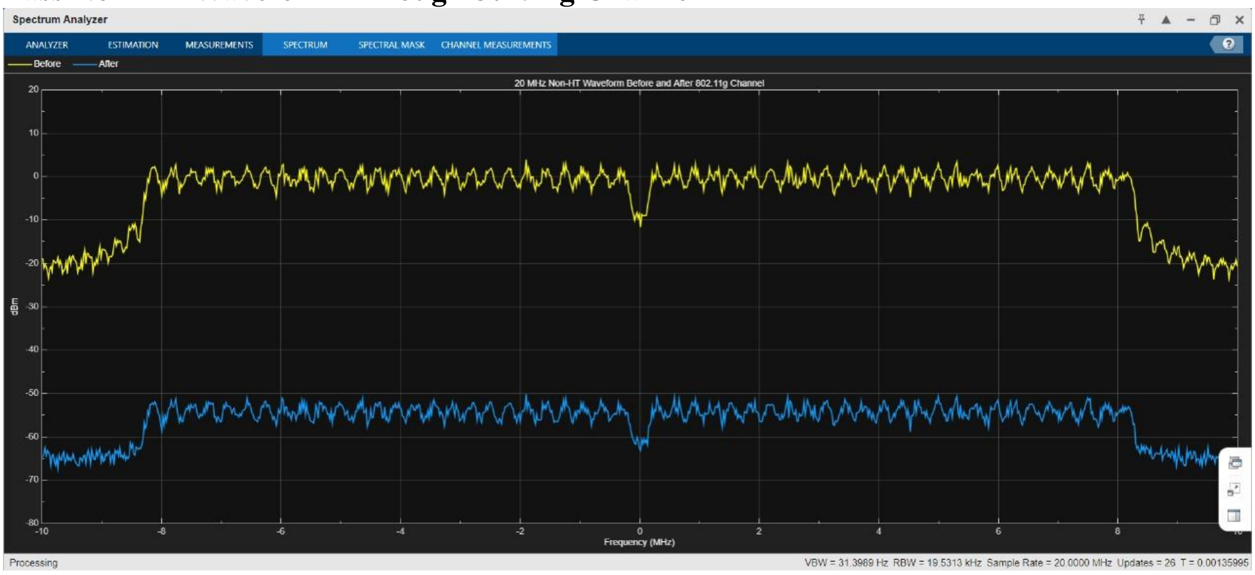
Path loss accounts for the roughly 50 to 60 dB of separation between the waveform before and after it passes through the TGac channel. The path loss results from the default transmitter-to-receiver distance of 3 meters, and from shadowing effects. The signal level variation shows the frequency selectivity of the delay profile across the frequency spectrum.

Pass HT Waveform Through TGn SISO Channel



Path loss accounts for the roughly 50 to 60 dB of separation between the waveform before and after it passes through the TGn channel. The path loss results from the default transmitter-to-receiver distance of 3 meters, and from shadowing effects. The signal level variation shows the frequency selectivity of the delay profile across the frequency spectrum.

Pass Non-HT Waveform Through 802.11g Channel



Free-space path loss accounts for the roughly 50 to 60 dB of separation between the waveform before and after it passes through the 802.11g channel. The path loss results from the specified transmitter-to-receiver distance of 3 meters, and from shadowing effects. The signal level variation shows the frequency selectivity of the delay profile across the frequency spectrum.

Pass VHT Waveform Through TGac MIMO Channel

The overlaid signals show the TGac channel variation between the received streams.

CHAPTER 7
PROJECT OUTCOME

- **CO1:** Understand the basics of mobile communication systems.
- Hence CO1 is attained.
- **CO4:** Differentiate GSM and CDMA wireless networks.
- Hence CO4 is attained.

CONCLUSION

In conclusion, this project underscores the importance of analyzing WLAN fading effects on both SISO and MIMO channels. By leveraging the capabilities of the WLAN Toolbox in MATLAB, we have gained valuable insights into the behaviour of WLAN systems under fading conditions. Moving forward, this knowledge can guide the design and optimization of WLAN networks to meet the demands of modern wireless communication systems..

CHAPTER 8
REFERENCES

- Erceg, V., L. Schumacher, P. Kyritsi, et al. *TGn Channel Models*. Version 4. IEEE 802.11-03/940r4, May 2004.
- Breit, G., H. Sampath, S. Vermani, et al. *TGac Channel Model Addendum*. Version 12. IEEE 802.11-09/0308r12, March 2010.
- WLAN Toolbox Documentation: https://www.mathworks.com/products/wlan.html
- WLAN Channel Models: https://www.mathworks.com/help/wlan/gs/wlan-channel-models.html
- WLAN (Wi-Fi) Fundamentals https://www.homenethowto.com/wireless/radio-theory-and-wi-fi/