# Project Based Learning Report

on

UMTS Uplink and Downlink Waveform Generation using LTE Toolbox in Matlab

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

in

Electronics & Communication Engineering

By

### PRN NAME

|  |  |
| --- | --- |
| 2114110480 | Manisha Kumari |
| 2114110481 | Summi Kumari |
| 2214110606 | Arpita Shree |

Under the guidance of Course In-charge

### Prof. Dr.Tanuja S.Dhope

Department of Electronics & Communication Engineering Bharati Vidyapeeth

(Deemed to be University)

College of Engineering, Pune – 411043

**Academic Year: 2024-25**

**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, “UMTS Uplink and Downlink Waveform Generation using LTE Toolbox in Matlab”. Is work done by

### PRN NAME

|  |  |
| --- | --- |
| 2114110480 | Manisha Kumari |
| 2114110481 | Summi Kumari |
| 2214110606 | Arpita Shree |

in partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in “**Cellular Technology & 4G**” of Bachelor of Technology Semester VI, in Electronics and Communication.

### Date:

**Prof.Dr.Tanuja S.Dhope Dr. Arundhati A. Shinde Course In-charge Professor & Head**

# INDEX: -

|  |  |  |
| --- | --- | --- |
| **SR.No.** | **Title** | **Page No.** |
| 1. | Problem Statement with Solution | 2 |
| 2. | Description about Project | 3 |
| 3. | Advantage and Disadvantage | 4 |
| 4. | Applications | 5 |
| 5. | Introduction to Matlab & LTE Toolbox | 6 |
| 6. | Flowchart | 7 |
| 7. | Algorithm | 8 |
| 8. | Code and Output | 9-16 |
| 9. | Course Outcome & Conclusion | 17 |
| 10. | References & Github-Link | 18 |

## Problem Statement: -

**UMTS Uplink and Downlink Waveform Generation using LTE Toolbox in Matlab.**

**Before UMTS, mobile networks faced slow data speeds, capacity constraints, spectrum limitations, and inadequate support for multimedia services. UMTS addressed these issues, offering higher speeds and improved capabilities.**

## Solution:-

1. UMTS provides high-speed data and voice communication.
2. It ensures seamless mobility and broad coverage.
3. Supports advanced services like video calling and mobile TV.
4. Offers international roaming and backward compatibility.
5. UMTS serves as a foundation for future mobile technologies.

## DESCRIBTION ABOUT PROJECT: -

The project aims to generate UMTS uplink and downlink waveforms using MATLAB's LTE Toolbox. By configuring parameters such as carrier frequency, spreading factor, code, and modulation scheme, the goal is to produce realistic representations of UMTS transmissions. This capability enables comprehensive analysis and simulation of UMTS communication systems within the MATLAB environment. Key methodologies include leveraging LTE Toolbox functions tailored to UMTS standards and validating generated waveforms through visualization and analysis. The project's outcomes include accurate UMTS waveforms for various applications, including network planning, performance evaluation, and research in wireless communication..

### Objective:

* Generate UMTS uplink and downlink waveforms using MATLAB's LTE Toolbox.

### Methodology:

* Utilize functions within the LTE Toolbox tailored to UMTS specifications.
* Define parameters such as carrier frequency, spreading factor, code, and modulation scheme for both uplink and downlink channels.
* Configure these parameters to accurately represent UMTS transmission standards.

### Importance:

* Facilitates analysis and simulation of UMTS communication systems.
* Enables testing of signal processing algorithms within the MATLAB environment.
* Supports simulation of various network scenarios for performance evaluation.

### Implementation:

* Utilize MATLAB's LTE Toolbox to streamline waveform generation.
* Customize parameters according to UMTS standards to ensure accurate representation.
* Validate generated waveforms through visualization and analysis.

### Potential Applications:

* Network planning and optimization.
* Performance evaluation of UMTS systems.
* Research and development of UMTS technologies and protocols.

### Outcome:

* Accurate UMTS uplink and downlink waveforms for further analysis and simulation.
* Enhanced understanding of UMTS communication systems.
* Facilitation of research and development in the field of wireless communication.

## Adavantage:-

1. **Unified Platform:** LTE Toolbox provides a unified platform for UMTS and LTE waveform generation, streamlining development and testing processes.
2. **Familiar Interface:** For engineers already proficient with LTE Toolbox, leveraging it for UMTS waveform generation reduces the learning curve and enhances productivity.
3. **Integrated Features:** LTE Toolbox may offer integrated features such as channel modeling, interference simulation, and performance analysis, enhancing the capabilities for UMTS waveform generation and evaluation.
4. **Efficiency:** Utilizing LTE Toolbox for UMTS waveform generation can lead to improved efficiency in terms of development time, computational resources, and simulation complexity.
5. **Updates and Support:** Users can benefit from ongoing updates, improvements, and technical support provided by MathWorks for LTE Toolbox, ensuring compatibility with evolving UMTS standards and requirements.

## Disadvantage:-

1. **Complexity:** UMTS and LTE are different standards with distinct waveform characteristics. Attempting to generate UMTS waveforms using LTE Toolbox might require complex conversions and adjustments, increasing the overall complexity of the process.
2. **Accuracy:** While LTE Toolbox may provide functionalities for generating UMTS-like waveforms, the accuracy might not be as high as using dedicated UMTS simulation tools or models. This could lead to discrepancies in the generated waveforms compared to real-world UMTS signals.
3. **Limited Features:** LTE Toolbox is primarily designed for LTE simulation and analysis. It may lack certain features specific to UMTS waveform generation, leading to compromises in fidelity or functionality.
4. **Performance:** Since LTE Toolbox is optimized for LTE, using it for UMTS waveform generation may result in suboptimal performance in terms of computational efficiency and simulation speed.
5. **Compatibility:** UMTS and LTE operate on different radio access technologies, and their waveform characteristics differ significantly. Using LTE Toolbox for UMTS waveform generation might not fully capture the intricacies of UMTS modulation, coding, and other parameters.

## Applications:-

### Wireless System Design:

* Design and optimization of wireless communication systems compatible with UMTS standards.

### Protocol Testing:

* Testing and validation of UMTS communication protocols for improved efficiency.

### Education:

* Educational tool for hands-on learning of UMTS waveform simulation and parameter impact.

### System Performance Analysis:

* Analysis of UMTS system performance under various conditions for optimization.

### Research and Development:

* Contribution to advanced wireless communication technologies and experimentation.

## Introduction to Matlab:

MATLAB stands as a cornerstone in engineering, scientific, and academic realms, offering a versatile programming language and a robust numerical computing environment. Its intuitive syntax simplifies complex computations, while its extensive library of functions empowers users to tackle diverse tasks with ease. With an array of built-in toolboxes covering various disciplines, from signal processing to machine learning, MATLAB serves as a one-stop solution for data analysis, algorithm development, and simulation. Its flexibility and efficiency make it indispensable for professionals and researchers alike, enabling groundbreaking discoveries and innovative solutions across countless domains**.**

## LTE ToolBox:-

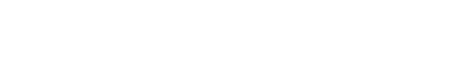
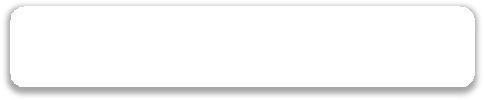
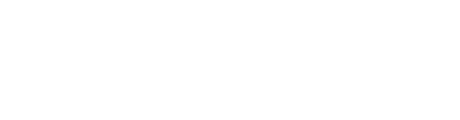
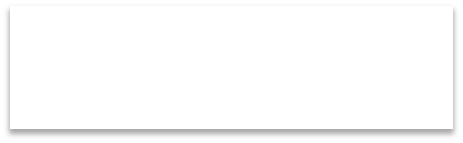
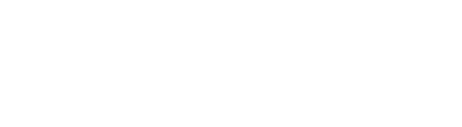
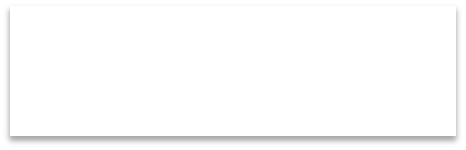
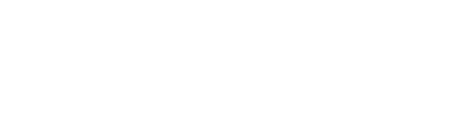
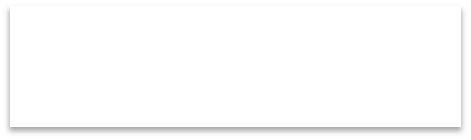
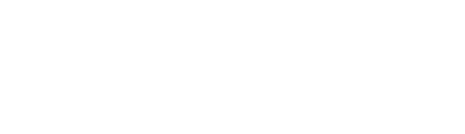
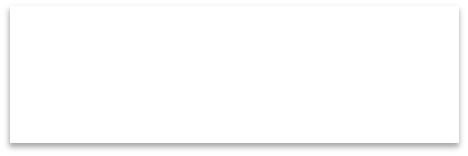
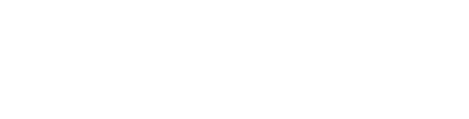
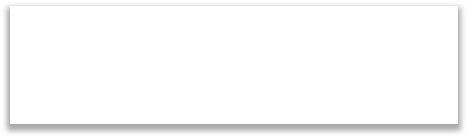
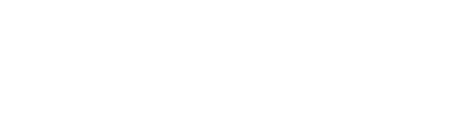
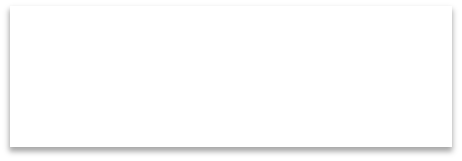
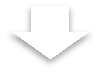
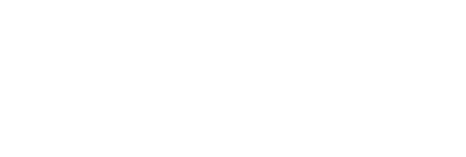
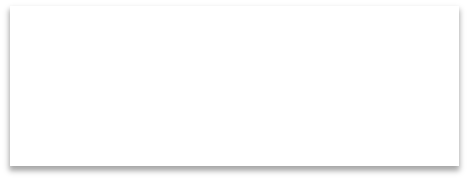
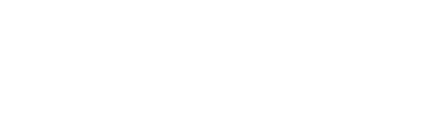
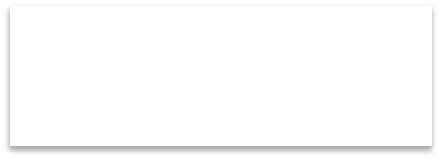
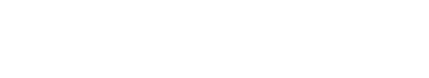
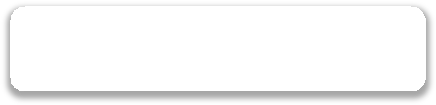
The LTE Toolbox in MATLAB is a comprehensive set of MATLAB functions and Simulink blocks for designing, simulating, and testing LTE and LTE-Advanced wireless communication systems. It provides tools for modeling the physical layer of LTE systems, including signal generation, channel modeling, link-level simulation, and performance analysis. The toolbox allows engineers and researchers to develop and evaluate LTE algorithms, analyze system performance under various conditions, and prototype LTE designs quickly and efficiently.

## Signal Generator:-

A signal generator in MATLAB is a tool used to create synthetic signals for various purposes, such as testing and simulation. It allows users to generate signals of different types, frequencies, amplitudes, and shapes. MATLAB provides built-in functions and toolboxes for signal generation, including functions for creating sinusoidal waves, square waves, sawtooth waves, random noise, and more. Additionally, MATLAB's signal processing toolbox offers advanced functionalities for designing and generating complex signals, such as modulated signals and chirp signals. Signal generators in MATLAB are essential for tasks like testing communication systems, designing filters, simulating control systems, and analyzing signal processing algorithms.

* In LTE Tool box we have used Signal Generator to generate Uplink and Downlink Waveform in Matlab.

## FLOWCHART:-



Start

Validate

Analyze

|

End

Re-validate

Adjust

Visualize

Generate

Configure

Define Parameters

**Algorithm:-**

### Define Parameters:

* Specify carrier frequency, spreading factor, code, and modulation scheme.

### Generate Downlink Waveform:

* Utilize LTE Toolbox function for downlink waveform generation.
* Use defined parameters to generate the downlink waveform.
* Output the downlink waveform.

### Generate Uplink Waveform:

* Utilize LTE Toolbox function for uplink waveform generation.
* Use defined parameters to generate the uplink waveform.
* Output the uplink waveform.

### Validate Downlink Waveform:

* Visualize the generated downlink waveform.

### Validate Uplink Waveform:

* Visualize the generated uplink waveform.

### Analyze:

* Compare generated waveforms with expected characteristics.

### Adjust Parameters (if necessary):

* Modify parameters based on analysis results.
* **End the algorithm.**

## CODE:- Uplink Waveform

frc = 'FRC1'; % FRC number

preconfigParams = umtsUplinkReferenceChannels(frc); % Get FRC parameters frcWaveform = umtsUplinkWaveformGenerator(preconfigParams);% Generate FRC waveform

% FRC definition from scratch

% General settings

uplinkParams.TotFrames = 1; % Number of frames to be generated uplinkParams.ScramblingCode = 1; % Scrambling code uplinkParams.FilterType = 'RRC'; % Enable the RRC filter uplinkParams.OversamplingRatio = 4; % Oversampling set to 4 uplinkParams.NormalizedPower = 'Off'; % No power normalization

% Define Uplink Dedicated Physical Data Channel (DPDCH) uplinkParams.DPDCH.Enable = 'On'; % Enable DPDCH uplinkParams.DPDCH.SlotFormat = 2; % DPDCH slot format uplinkParams.DPDCH.CodeCombination = 64; % DPDCH spreading factor uplinkParams.DPDCH.Power = 0; % Power in dB uplinkParams.DPDCH.DataSource = 'CCTrCH'; % DPDCH data source is CCTrCH

% DPDCH carries the Coded Composite Transport Channel (CCTrCH) containing

% one or more transport channels. Since DPDCH source is specified as

% CCTrCH, define the CCTrCH containing DTCH and DCCH transport channels

% Build DTCH definition

TrCH(1).Name = 'DTCH'; % Name of the transport channel TrCH(1).CRC = '16'; % CRC type

TrCH(1).CodingType = 'conv3'; % The coding type and rate TrCH(1).RMA = 256; % Rate matching attribute TrCH(1).TTI = 20; % TTI in ms

TrCH(1).DataSource = 'PN9-ITU'; % Tr channel data source TrCH(1).ActiveDynamicPart = 1; % Index to active dynamic part TrCH(1).DynamicPart(1) = struct('BlockSize',244,'BlockSetSize',244); % 1x244 blocks

% Build DCCH definition

TrCH(2).Name = 'DCCH'; % Name of the transport channel TrCH(2).CRC = '12'; % CRC type

TrCH(2).CodingType = 'conv3'; % The coding type and rate TrCH(2).RMA = 256; % Rate matching attribute TrCH(2).TTI = 40; % TTI in ms

TrCH(2).DataSource = 'PN9-ITU'; % Tr channel data source TrCH(2).ActiveDynamicPart = 1; % Index to active dynamic part TrCH(2).DynamicPart(1) = struct('BlockSize',100,'BlockSetSize',100); % 1x100 blocks

% Finalize CCTrCH structure array using the TrCH structures defined above uplinkParams.DPDCH.CCTrCH.Name = 'DCH'; % Name of the CCTrCH uplinkParams.DPDCH.CCTrCH.TrCH = TrCH; % Assign DTCH/DCCH to CCTrCH

% Define DPCCH

uplinkParams.DPCCH.Enable = 'On'; % Enable DPCCH uplinkParams.DPCCH.SlotFormat = 0; % Slot format number uplinkParams.DPCCH.Power = -5.46; % Power in dB

uplinkParams.DPCCH.TPCData = 1; % TPC value uplinkParams.DPCCH.TFCI = 0; % TFCI value uplinkParams.DPCCH.FBIData = 0; % FBI value

% Define HSUPA channels

uplinkParams.HSUPA.Enable = 'On'; % Enable HSUPA channels uplinkParams.HSUPA.CodeCombination = [4 4]; % E-DPDCH spreading factors uplinkParams.HSUPA.EDPDCHPower = -5.46+12.04; % Power in dB uplinkParams.HSUPA.EDPCCHPower = -5.46+6.02; % Power in dB uplinkParams.HSUPA.RSNSequence = 0; % RSN value uplinkParams.HSUPA.ETFCI = 0; % E-TFCI value uplinkParams.HSUPA.HappyBit = 0; % Happy Bit value uplinkParams.HSUPA.DataSource = 'EDCH'; % Data source is E-DCH uplinkParams.HSUPA.EDCH.BlockSize = 2706; % E-DCH transport block size uplinkParams.HSUPA.EDCH.TTI = 2; % E-DCH TTI in ms uplinkParams.HSUPA.EDCH.Modulation = 'BPSK'; % Modulation scheme uplinkParams.HSUPA.EDCH.DataSource = 'PN9-ITU'; % E-DCH Data source

% Define HS-DPCCH, but disable for FRC1 generation uplinkParams.HSDPCCH.Enable = 'Off'; % Disable HS-DPCCH uplinkParams.HSDPCCH.Power = 0; % Power in dB uplinkParams.HSDPCCH.CQI = 0; % CQI value uplinkParams.HSDPCCH.HARQACK = 1; % HARQ ACK bit value uplinkParams.HSDPCCH.UEMIMO = 0; % UE not in MIMO mode

% The structure defined above can be used to generate the waveform: frcWaveform2 = umtsUplinkWaveformGenerator(uplinkParams);

% For completeness we can see that the FRC definition structures obtained

% by the above two parameterization approaches are identical if(isequal(uplinkParams,preconfigParams))

disp(['FRC1 definitions generated with and without using ' ... 'umtsUplinkReferenceChannels function are the same.']);

end if(isequal(frcWaveform,frcWaveform2))

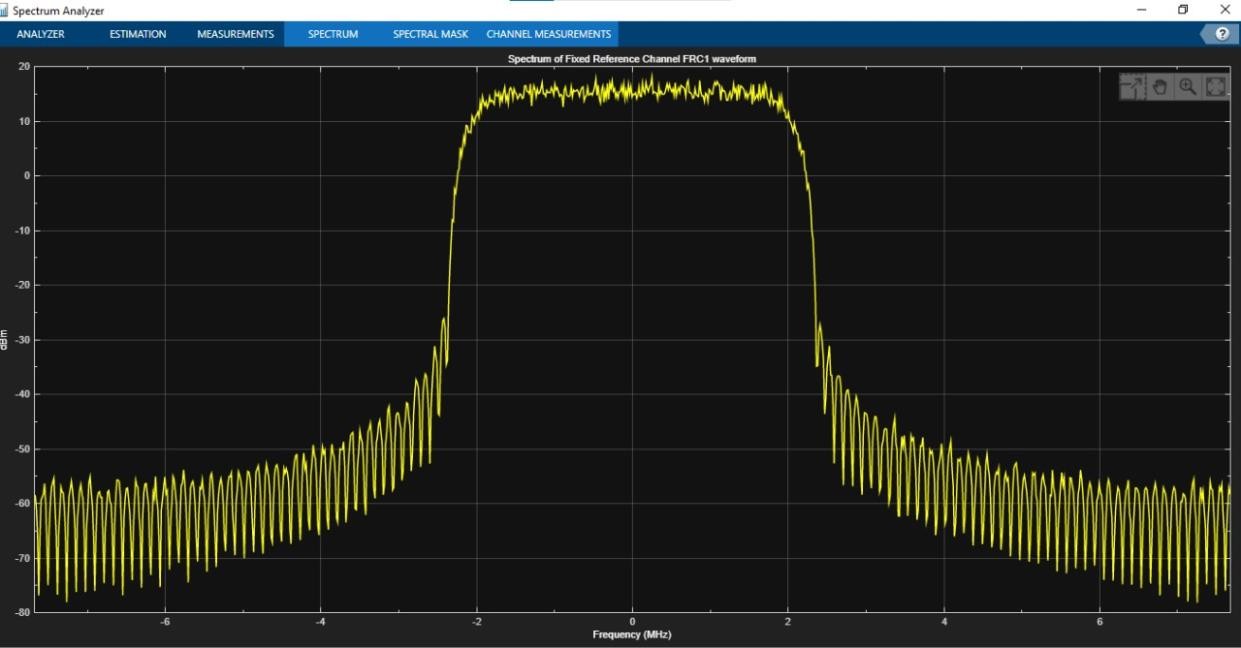
disp(['FRC1 waveforms generated with and without using' ... ' umtsUplinkReferenceChannels function are the same.']);

end

chiprate = 3.84e6; % Chip rate of the baseband waveform

spectrumPlot = spectrumAnalyzer(SampleRate=chiprate\*uplinkParams.OversamplingRatio); spectrumPlot.Title = sprintf('Spectrum of Fixed Reference Channel %s waveform', frc); spectrumPlot(frcWaveform);

## Output:-



**Code:- Downlink Waveform**

hset = 'H-Set1'; % H-Set number modulation = 'QPSK'; % Modulation scheme

preconfigParams = umtsDownlinkReferenceChannels(hset,modulation); % Get H-Set parameters

frcWaveform = umtsDownlinkWaveformGenerator(preconfigParams); % Generate H-Set waveform

% H-Set parameter structure definition from scratch

% General settings

downlinkParams.TotFrames = 1; % Number of 10ms frames to be generated downlinkParams.PrimaryScramblingCode = 0; % Primary scrambling code downlinkParams.FilterType = 'RRC'; % Enable the RRC filter downlinkParams.OversamplingRatio = 4; % Oversampling set to 4 downlinkParams.NormalizedPower = 'Off'; % Power normalization disabled

% Define Downlink Dedicated Physical Channel (DPCH) downlinkParams.DPCH.Enable = 'On'; % Enable DPCH downlinkParams.DPCH.SlotFormat = 11; % DPCH slot format downlinkParams.DPCH.SpreadingCode = 6; % DPCH spreading code downlinkParams.DPCH.NMulticodes = 1; % Number of DPCH downlinkParams.DPCH.SecondaryScramblingCode = 1; % Secondary scrambling code downlinkParams.DPCH.TimingOffset = 0; % Timing Offset downlinkParams.DPCH.Power = 0; % Power in dB downlinkParams.DPCH.TPCData = 0; % TPC value downlinkParams.DPCH.TFCI = 0; % TFCI value downlinkParams.DPCH.DataSource = 'CCTrCH'; % DPCH data source is CCTrCH

% DPCH carries the Coded Composite Transport Channel (CCTrCH) containing

% one or more transport channels. Since DPCH source is specified as CCTrCH,

% define the CCTrCH containing DTCH and DCCH transport channels

% Build DTCH definition

TrCH(1).Name = 'DTCH'; % Name of the transport channel TrCH(1).CRC = '16'; % CRC type

TrCH(1).TTI = 20; % TTI in ms

TrCH(1).CodingType = 'conv3'; % The coding type and rate TrCH(1).RMA = 256; % Rate matching attribute TrCH(1).DataSource = 'PN9-ITU'; % Tr channel data source TrCH(1).ActiveDynamicPart = 1; % Index to active dynamic part TrCH(1).DynamicPart(1) = struct('BlockSize',244,'BlockSetSize',244); % 1x244 blocks

% Build DCCH definition

TrCH(2).Name = 'DCCH'; % Name of the transport channel TrCH(2).CRC = '12'; % CRC type

TrCH(2).TTI = 40; % TTI in ms

TrCH(2).CodingType = 'conv3'; % The coding type and rate TrCH(2).RMA = 256; % Rate matching attribute TrCH(2).DataSource = 'PN9-ITU'; % Tr channel data source TrCH(2).ActiveDynamicPart = 1; % Index to active dynamic part TrCH(2).DynamicPart(1) = struct('BlockSize',100,'BlockSetSize',100); % 1x100 blocks

% Finalize CCTrCH structure array using the TrCH structures defined above downlinkParams.DPCH.CCTrCH.Name = 'DCH'; % Name of the CCTrCH downlinkParams.DPCH.CCTrCH.DTXPosition = 'fixed'; % DTX position downlinkParams.DPCH.CCTrCH.TrCH = TrCH; % Assign DTCH/DCCH to CCTrCH

% Define P-CCPCH

downlinkParams.PCCPCH.Enable = 'On'; % Enable P-CCPCH downlinkParams.PCCPCH.Power = 0; % Set power to be 0dB downlinkParams.PCCPCH.DataSource = 'CCTrCH'; % P-CCPCH data source is CCTrCH

% P-CCPCH CCTrCH carries the BCH transport channel. Since P-CCPCH source is

% CCTrCH, define CCTrCH containing BCH clear TrCH;

TrCH(1).Name = 'BCH'; % Name of the Tr channel TrCH(1).CRC = '16'; % CRC type

TrCH(1).TTI = 20; % TTI in ms

TrCH(1).CodingType = 'conv2'; % The coding type and rate TrCH(1).RMA = 256; % Rate matching attribute TrCH(1).DataSource = 'PN9-ITU'; % Tr channel data source TrCH(1).ActiveDynamicPart = 1; % Index to active dynamic part TrCH(1).DynamicPart(1) = struct('BlockSize',246,'BlockSetSize',246); % 1x246 block

% Finalize CCTrCH structure array using the TrCH structure defined above downlinkParams.PCCPCH.CCTrCH.Name = 'BCH'; % Name of the CCTrCH downlinkParams.PCCPCH.CCTrCH.DTXPosition = 'fixed';% DTX position downlinkParams.PCCPCH.CCTrCH.TrCH = TrCH; % Assign BCH to CCTrCH

% Define S-CCPCH, but this channel is not required for H-Set1 generation downlinkParams.SCCPCH.Enable = 'Off'; % Disable S-CCPCH downlinkParams.SCCPCH.SlotFormat = 7; % Slot format number downlinkParams.SCCPCH.SpreadingCode = 3; % S-CCPCH spreading code downlinkParams.SCCPCH.SecondaryScramblingCode = 3; % Secondary scrambling code downlinkParams.SCCPCH.TimingOffset = 0; % Timing Offset downlinkParams.SCCPCH.Power = 0; % Power in dB downlinkParams.SCCPCH.TFCI = 0; % TFCI value downlinkParams.SCCPCH.DataSource = 'CCTrCH'; % S-CCPCH data source is CCTrCH

% S-CCPCH CCTrCH can carry PCH and FACH transport channels. Since S-CCPCH

% source is CCTrCH, define CCTrCH containing PCH and FACH

% Build PCH definition

TrCH(1).Name = 'PCH'; % Name of the Tr channel

TrCH(1).CRC = '16'; % CRC type

TrCH(1).TTI = 10; % TTI in ms

TrCH(1).CodingType = 'conv2'; % The coding type TrCH(1).RMA = 256; % Rate matching attribute TrCH(1).DataSource = 'PN9-ITU'; % Tr channel data source TrCH(1).ActiveDynamicPart = 1; % Index to active dynamic part

TrCH(1).DynamicPart(1) = struct('BlockSize',64,'BlockSetSize',64); % 1x64 block

% Build FACH definition

TrCH(2).Name = 'FACH'; % Name of the Tr channel TrCH(2).CRC = '16'; % CRC type

TrCH(2).TTI = 10; % TTI in ms

TrCH(2).CodingType = 'turbo'; % The coding type TrCH(2).RMA = 256; % Rate matching attribute TrCH(2).DataSource = 'PN9-ITU'; % Tr channel data source TrCH(2).ActiveDynamicPart = 1; % Index to active dynamic part

TrCH(2).DynamicPart(1) = struct('BlockSize',360,'BlockSetSize',360); % 1x360 block

% Finalize CCTrCH using the above downlinkParams.SCCPCH.CCTrCH.Name = ''; % Name of the CCTrCH downlinkParams.SCCPCH.CCTrCH.DTXPosition = 'fixed';% DTX position

downlinkParams.SCCPCH.CCTrCH.TrCH = TrCH; % Assign PCH/FACH to CCTrCH

% Define P-CPICH

downlinkParams.PCPICH.Enable = 'On'; % Enable P-CPICH downlinkParams.PCPICH.Power = 0; % Power in dB

% Define S-CPICH

downlinkParams.SCPICH.Enable = 'Off'; % Disable S-CPICH downlinkParams.SCPICH.SpreadingCode = 4; % S-CPICH spreading code downlinkParams.SCPICH.SecondaryScramblingCode = 4; % Secondary scrambling code downlinkParams.SCPICH.Power = 0; % Power in dB

% Define P-SCH

downlinkParams.PSCH.Enable = 'On'; % Enable P-SCH downlinkParams.PSCH.Power = 0; % Power in dB

% Define S-SCH

downlinkParams.SSCH.Enable = 'On'; % Enable S-SCH downlinkParams.SSCH.Power = 0; % Power in dB

% Define PICH

downlinkParams.PICH.Enable = 'On'; % Enable PICH downlinkParams.PICH.SpreadingCode = 16; % PICH spreading code downlinkParams.PICH.TimingOffset = 0; % Timing offset downlinkParams.PICH.Power = 0; % Power in dB downlinkParams.PICH.DataSource = 'PagingData'; % PICH data source downlinkParams.PICH.Np = 144; % Number of paging indicators

% Define HSDPA

downlinkParams.HSDPA.Enable = 'On'; % Enable HSDPA channels downlinkParams.HSDPA.CodeGroup = 5; % Number of HS-PDSCHs downlinkParams.HSDPA.CodeOffset = 1; % Code offset to first HS-PDSCH downlinkParams.HSDPA.Modulation = 'QPSK'; % Modulation scheme downlinkParams.HSDPA.VirtualBufferCapacity = 9600; % Buffer capacity downlinkParams.HSDPA.InterTTIDistance = 3; % Inter TTI interval downlinkParams.HSDPA.NHARQProcesses = 2; % Number of HARQ processes downlinkParams.HSDPA.XrvSequence = [0 2 5 6]; % The XRV sequence downlinkParams.HSDPA.UEId = 0; % UE Identity downlinkParams.HSDPA.TransportBlockSizeId = 41; % The transport block size id downlinkParams.HSDPA.HSSCCHSpreadingCode = 9; % Shared channel spreading code downlinkParams.HSDPA.SecondaryScramblingCode = 6; % Secondary scrambling code downlinkParams.HSDPA.HSPDSCHPower = 0; % HS-PDSCH power in dB downlinkParams.HSDPA.HSSCCHPower = 0; % HS-SCCH power in dB downlinkParams.HSDPA.DataSource = 'HSDSCH'; % Data source is HS-DSCH

% HS-DSCH transport channel definition downlinkParams.HSDPA.HSDSCH.BlockSize = 3202; % The transport block size downlinkParams.HSDPA.HSDSCH.DataSource = 'PN9-ITU';% HS-DSCH data source

% Define OCNS channels as defined in TS25.101 Table C.13 downlinkParams.OCNS.Enable = 'On'; % Enable OCNS channels downlinkParams.OCNS.Power = 0; % OCNS power scaling in dB downlinkParams.OCNS.OCNSType = 'H-Set\_6DPCH'; % OCNS definition

% The structure defined above can be used to generate the waveform: frcWaveform2 = umtsDownlinkWaveformGenerator(downlinkParams);

% For completeness we can see that the H-Set definition structures obtained

% by the above two parameterization approaches are identical if(isequal(preconfigParams,downlinkParams))

disp(['H-Set1 configuration structures generated with and without using' ... ' umtsDownlinkReferenceChannels function are the same.']);

end if(isequal(frcWaveform,frcWaveform2))

disp(['H-Set1 waveforms generated with and without using' ... ' umtsDownlinkReferenceChannels function are the same.']);

end

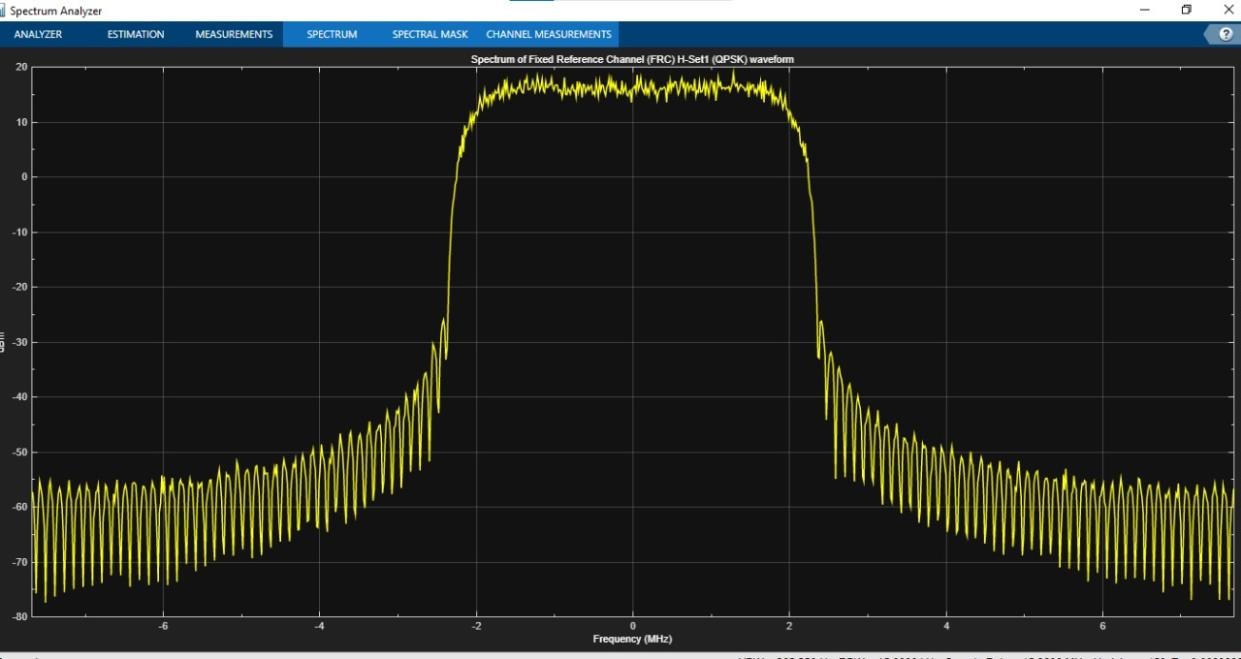
chiprate = 3.84e6; % Chip rate of the baseband waveform

spectrumPlot = spectrumAnalyzer(SampleRate=chiprate\*downlinkParams.OversamplingRatio);

spectrumPlot.Title = sprintf('Spectrum of Fixed Reference Channel (FRC) %s (%s) waveform', hset, modulation);

spectrumPlot(frcWaveform);

## Output:-



**COURSE OUTCOME:-**

**CO1:-** Understand the basics of Mobile Communication System.

**CO5:**- Examine the 3G and Future Communication Technology’s Evolution.

Hence CO1 & CO5 has been satisfied.

## CONCLUSION:-

In conclusion, utilizing the LTE Toolbox in MATLAB for UMTS uplink and downlink waveform generation provides a robust framework for simulating and analyzing UMTS communication systems. By leveraging the toolbox's comprehensive functionality, users can accurately model various aspects of UMTS waveforms, including channel characteristics, modulation schemes, and coding techniques. This enables researchers and engineers to evaluate system performance, optimize parameters, and develop advanced UMTS communication algorithms with confidence. Overall, the LTE Toolbox in MATLAB serves as a valuable tool for studying and designing UMTS communication systems.

## References:-

[**https://www.mathworks.com/help/lte/ug/umts-uplink-waveform-generation.html**](https://www.mathworks.com/help/lte/ug/umts-uplink-waveform-generation.html)

[**https://www.mathworks.com/help/lte/ug/umts-downlink-waveform-generation.html**](https://www.mathworks.com/help/lte/ug/umts-downlink-waveform-generation.html)

## Github -Link:-