A report

on

DESIGN OF SDR FOR HD VIDEO COMMUNICATION

Sashank Krishna S	2019A8PS0184P
Aditya Soni	2019A8PS1282H

Prepared in Partial Fulfillment of Practice School - 1
(Course No: BITS F221)
at
Department of Communications
Military College of Electronics and Mechanical Engineering
a Practice School - 1 Station
of

Birla Institute of Technology and Science, Pilani

July, 2021

Birla Institute of Technology and Science, Pilani
Rajasthan
Practice School Division

Station: Military College of Electronics and Mechanical Engineering, Secundrabad
Duration: Seven Weeks **Date of Start:** May 31, 2021
Date of Submission: July 23, 2021
Title: Design of SDR for HD Video communication

Students:
Sashank Krishna S
2019A8PS0184P
B.E. Electronics and Instrumentation

Aditya Soni
2019A8PS1282H
B.E. Electronics and Instrumentation

Mentors:
AEE Naveen Reddy
Military college of Electronics and Mechanical Engineering

Dr. Rahul Singhal
Assistant professor,
BITS Pilani, Pilani Campus

Keywords:
MIMO-OFDM, Polar codes, USRP, HD Video communication, DVB, H.264, H.265

Project Areas:
Wireless communications, Digital communications, Software defined Radio, Video codecs

Abstract:
The project aims to model a MIMO-OFDM + Polar/LDPC coding system using MATLAB. The system is for HD video communication, and the target hardware is the USRP B210. The video is processed using the H.264 / H.265 codecs, and the system supports both transfer of files, and streaming of the video.

Student Signature **Mentor Signature**

Contents

1 About the Project

1.1 Scope

The project aimed to implement a MIMO-OFDM system using USRP n210s [1] and b210s [2]. The link was to employ polar coding as the channel coding scheme. The video is to compressed using the H.264 / H.265 standards and transmitted through the implemented link.[3] The receiving end is to be able to both store the video as well as stream the same. Lastly, a power amplifier is to be selected for extending the range of the wireless link.
The project involved the following sub-problems:

- Modelling of a MIMO-OFDM link [4]
- Implementation of Polar coding
- Translation of the above models into HDL, and configuring the SDR
- Incorporation of H.264 and H.265 codecs
- Selection of suitable power amplifier

1.2 Modified Scope

The project implements a Monte-Carlo simulation of a MIMO-OFDM system. The link employs LDPC for the channel coding scheme. The video is compressed using the H.264 / H.265 standards using FFMPEG, and transmitted through the implemented model. File transfer as well as streaming capabilities were implemented. The model implemented is highly parameterized, and supporting scripts for analysis and visualization were also written.

Our initial objective was to implement polar codes in the system, but the 5G NR Polar codes function defined in MATLAB were very slow and were thus not at all suitable for both file transfer and live-streaming.

The following sub-problems were solved

- Modelling of an OFDM link
- Implementation of channel inversion
- Upgradation to a MIMO-OFDM link
- Implementation of LDPC error correction
- Integration with MIMO-OFDM model
- Parameterization of the model
- Enabling of streaming and file transfer support
- Incorporation of H.264 and H.265 via the use of FFMPEG
- Writing of supplementary scripts

1.3 Learning outcomes

1. Proficiency with MATLAB, and the communications toolbox
2. A strong background in wireless communications
3. A strong background in digital communications
4. A good understanding of 4G, and 5G communication standards
5. A strong understanding of MIMO-OFDM
6. A good understanding of Polar coding and LDPC [5]
7. Insight into video compression and digital image processing [6]
8. Familiarity with the theory behind SDRs [7] [8]

1.4 Project work plan

The project was decoupled into two independent sets of problem statements, and the work was done as shown below :

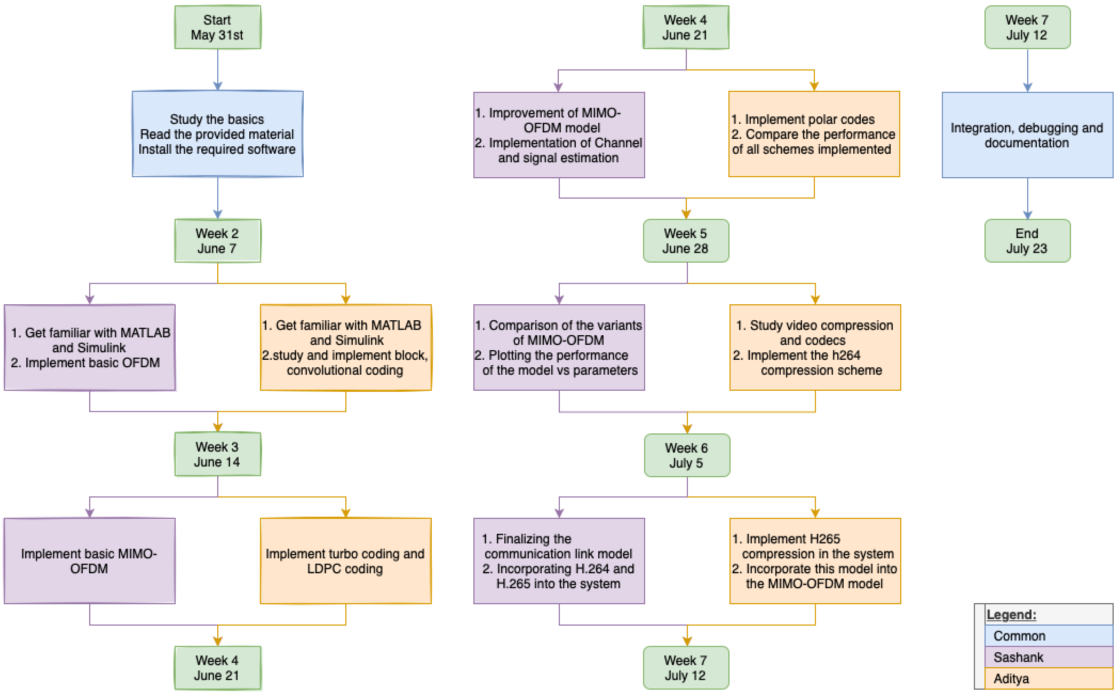

Figure 1: The work split and plan followed by the team

Sashank modelled and parameterized the MIMO-OFDM system, while Aditya implemented and tested an LDPC system. Sashank integrated the 2 systems, while Aditya worked on introducing HEVC and AVC into the system using FFMPEG.

2 Research Methodologies

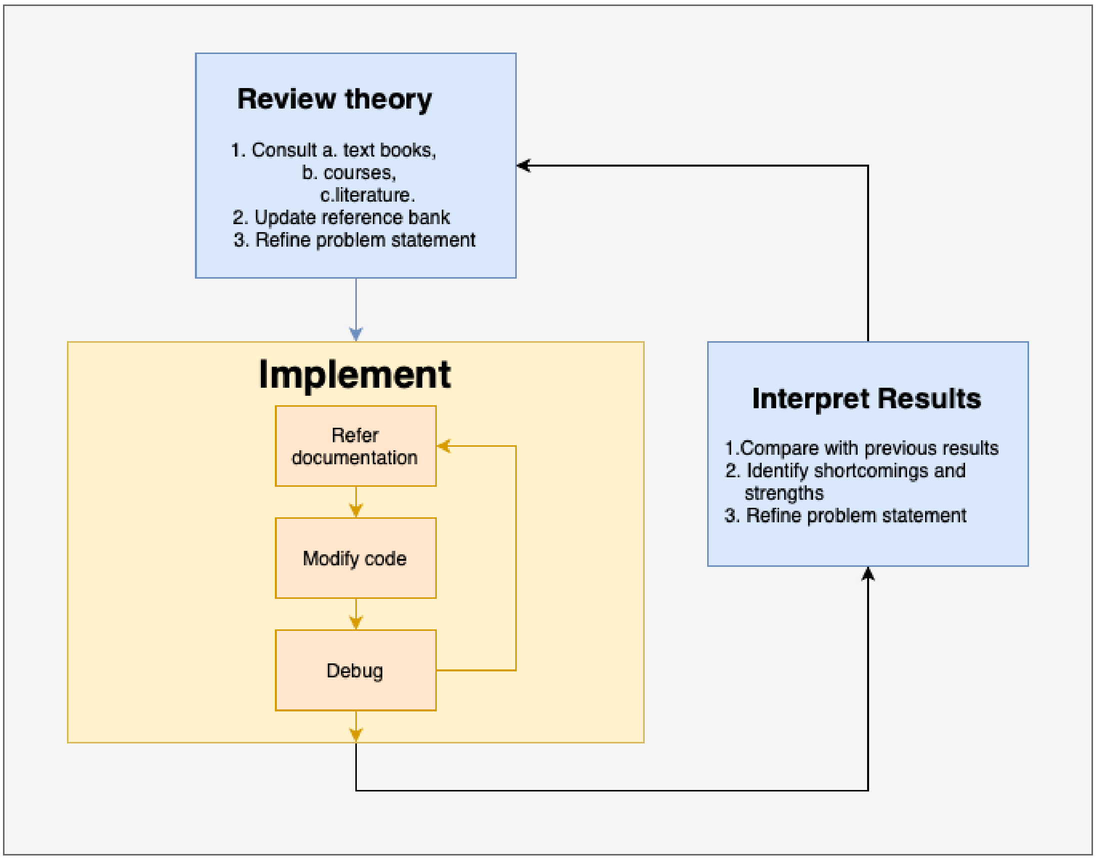

Figure 2: Research Methodology followed

- Reviewed the theory required using the provided supplementary material and other miscellaneous sources referenced to.
- Proceeded to implement the theory learnt.
  - Referred the documentations of the respective software used
  - Modified the code accordingly
  - Debugged the errors, went back to referring the documentation in case of errors
- Compared the results obtained to the previous results and identified any shortcomings, then went back to reviewing the theory.

3 Theoretical Background

3.1 QPSK and QAM

Chunks of bits are translated into symbols on a constellation diagram and are transmitted. The bits can be recovered by inverting the mapping from the constellation diagram at the receiver end. Here, the received symbol gets mapped to the nearest meaningful point on the constellation diagram.

The constellation diagram has two axes: the I (in-phase) and Q (Quadrature-phase) axes. The (I, Q) coordinates together map to a superposition of a cosine wave and a sine wave of corresponding amplitudes. Effectively, the information is encoded in the amplitude level and the phase shift of the resulting sinusoid.

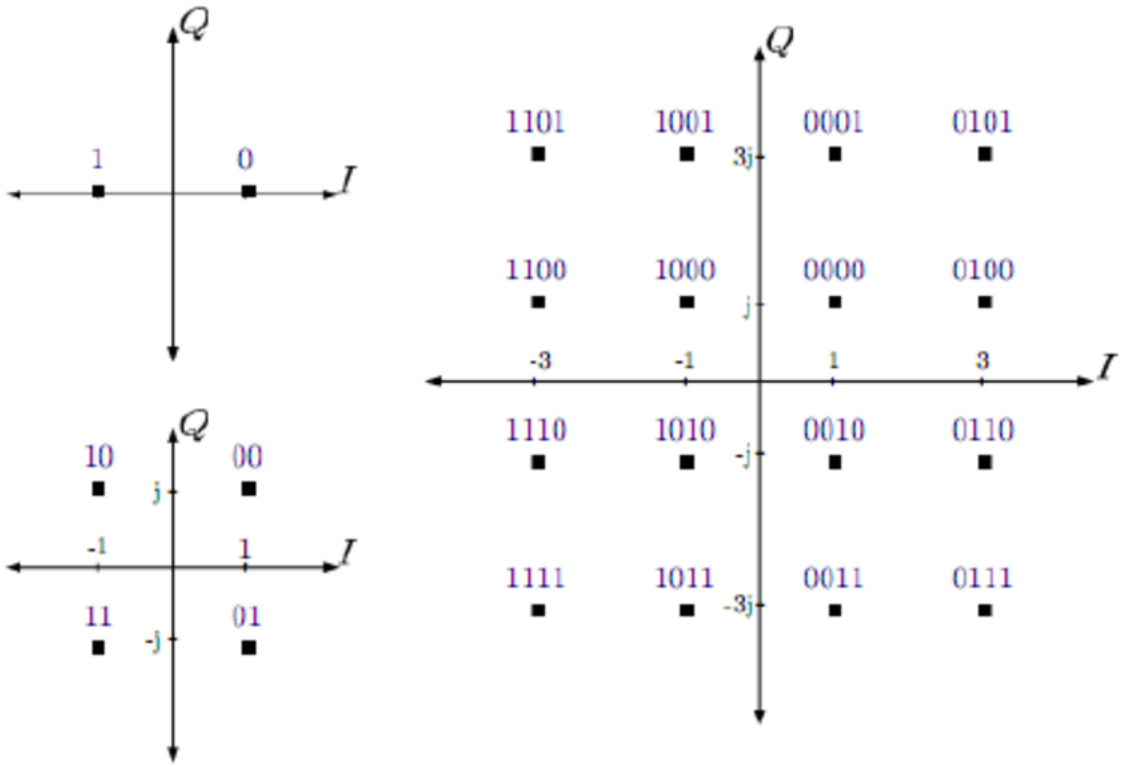

Figure 3: Mapping of bits to constellation (gray code)

3.2 Wireless channels and channel estimation

The waves transmitted through a wireless channel undergo reflections, refractions, scattering and even diffraction due to the miscellaneous obstacles making up the channel. We even get constructive and destructive interference of the waves, and can get both significant increases as well as significant drops in amplitude. Effectively, the receiver receives a superposition of time-shifted, phase-shifted, amplitude scaled copies of the signal transmitted. Wireless channels hence necessitate heavy signal processing in order to estimate the transmitted signal.

On making several mathematical assumptions and manipulating equations, we can model the channel using a complex multiplicative factor. The amplitude of the complex number following a Rayleigh distribution and the phase following a uniform distribution.

$$ae^{i\Phi} : P(a) = 2ae^{-a^2}; P(\phi) = \frac{1}{2\pi}$$

This result poses a significant problem since we encoded the information in the phase and amplitude of a sinusoid. Hence, to recover the information, we must invert the effects of the channel at the receiver end. i.e. We must estimate the channel.

The transmitter and receiver can agree upon pilot data, and in the case of OFDM, agree on pilot carriers as well, to estimate the channel matrix using mathematical techniques such as the approach of least squares. [9] Basically, the squared error between the best fit choice of channel matrix and the actual channel matrix can be minimized through matrix methods, and the optimal estimate can be carried forward. Similarly, LMMSE based inversion methods are available too. [10]

Next, the estimate of the channel matrix can used to estimate the signal that was transmitted from the received signal. Three approaches to do are LSE, LMMMSE and VBLAST [11],[12]. LMMSE is mathematically superior to least squares since the noise added is AWGN, and since the approach involves taking the mean, the noise would roughly average out. VBLAST is a recursive implementation of LMMSE, which also yields us better overall performance.

Another problem faced stems from the shape of the Rayleigh distribution. There is a chance that the scaling factor a will be uncomfortably close to zero. When this happens, the noise will dominate the signal, and the information will be lost forever. This condition is known as a deep fade. The problem of a deep fade can be alleviated by increasing the diversity order. However, in case a deep fade does occur, it plagues an entire OFDM frame with errors and results in a burst error.

Furthermore, channels in general suffer from path loss. Path loss can be modelled as follows :

$$FSPL = \frac{P_r}{P_t} = \left(\frac{4\pi df}{c}\right)^2$$

3.3 MIMO

MIMO [13] systems are used commonly to alleviate the problems posed by deep fades. If multiple transmitters and receivers are used, then the links between all transmitters and receivers must suffer from deep fade for the system to fail. If even one link survives, the message can still get through.
When we use multiple receivers, an issue that crops up is that now we have multiple copies of the signal and need to estimate the initially transmitted signal from them. For this purpose, we perform beamforming. A simple example of a beamforming technique is the maximal ratio combiner, where we weigh the various receivers by their channel coefficients.

An additional flexibility that MIMO offers is that we can employ orthogonal space-time block coding (OSTBC) to improve our performance. Space-time block coding refers to making the retrieval of the symbols easier by utilizing redundancies spread across antennas and time. i.e. across space and time. An example of the same is the Alamouti code.

3.4 OFDM

Orthogonal Frequency Division Multiplexing is a multi-carrier modulation technique that employs frequency division multiplexing but using orthogonal frequencies. I.e. with pulses of duration such that in the frequency domain, when any of the corresponding sync functions are peaking, all others are at zero. Hence, as we can pack more peaks using lower bandwidth, OFDM allows for spectrally efficient multiplexing.

3.5 LDPC

In information theory, a low-density parity-check (LDPC) code is a linear error correcting code, a method of transmitting a message over a noisy transmission channel. An LDPC is constructed using a sparse Tanner graph (subclass of the bipartite graph). LDPC codes are capacity-approaching codes, which means that practical constructions exist that allow the noise threshold to be set very close to the theoretical maximum (the Shannon limit) for a symmetric memoryless channel. The noise threshold defines an upper bound for the channel noise, up to which the probability of lost information can be made as small as desired. Using iterative belief propagation techniques, LDPC codes can be decoded in time linear to their block length.

3.6 Video Compression

Video data may be represented as a series of still image frames. Such data usually contains abundant amounts of spatial and temporal redundancy. Video compression algorithms attempt to reduce redundancy and store information more compactly.

Uncompressed video requires a very high data rate. Most video coding standards, such as the H.26x and MPEG formats, typically use motion-compensated DCT video coding (block motion compensation) to compress the videos. Although lossless video compression codecs perform at a compression factor of 5 to 12, a typical H.264 lossy compression video has a compression factor between 20 and 200. H.265 takes this one step further by performing at a compression factor twice of that of H.264.

4 System Overview

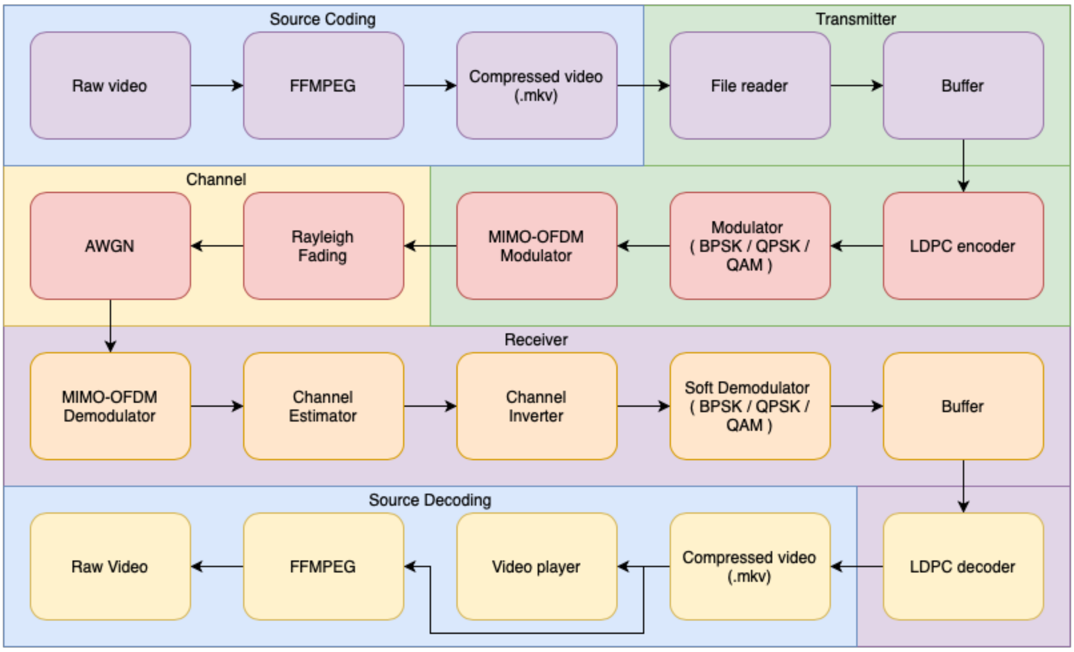

Figure 4: A block diagram representation of the implemented system

The scripts implemented are summarized below, and the results they generate, along with their scope, has also been highlighted.

4.1 Video compression using FFmpeg

The FFmpeg script encodes the videos in H.265, and the parameters are chosen according to the constraints of the user. The system supports encoding in H.264 but this is not used in the system because we place more importance on the compression factor than the video compression computational complexity.[14]

4.2 LDPC

This system used the predefined LDPC functions defined in the MATLAB communications toolbox. The LDPC encoder uses the standard DVB.S2 parity matrix, which has a code rate of 1/2. The encoder accepts messages in blocks of 32400 bits and the decoder inputs a soft-demodulated block of length 64800.

Since the performance of the CPU implementation was insufficient for both file transfer and streaming, we replaced the LDPC Decoder function with its GPU counterpart that can decode 30 blocks simultaneously. [15] This greatly sped up the system and made it possible to live-stream at low bitrates.[16],[17].

4.3 MIMO-OFDM System

The script implements a monte-carlo simulator [18]. A large block of bits is generated at the start, and transmitted through the model, and the output is compared with the input, and the BER is plotted.

The script allows for easy switching between BPSK, QPSK and M-QAM by just specifying modulation order. We can also specify the number of transmitter and receiver antennas in the MIMO system, and the model would accordingly change. Also, one can set the centre frequency and distance to effectively model the path loss.

Furthermore, all the parameters of the OFDM system can also be changed. The number of subcarriers, Number of Pilot carriers, Number of left and right guard carriers, and the cyclic prefix length can be changed easily. Additionally, pulse shaping can be toggled, and the windowing length can optionally be specified.

The script uses a Rayleigh channel model [19], and also adds AWGN in the order : Rayleigh channel, FSPL, AWGN.

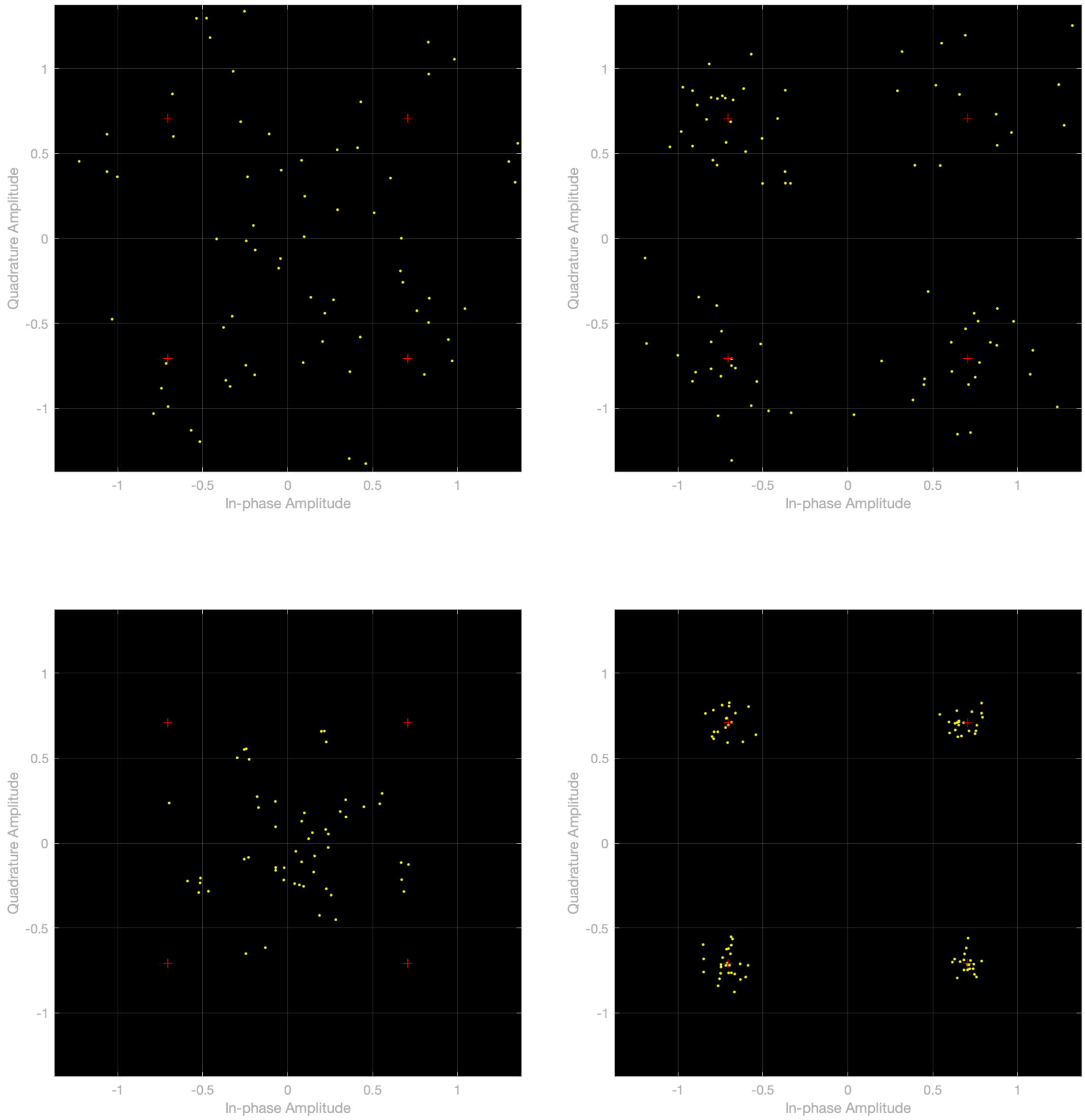

Figure 5: Plots illustrating the need for channel estimation

4.4 Channel & Signal Estimation

Least squares based channel estimation was implemented. The effect of the same is demonstrated in figure ??. The method of least squares, as well as the method of linear minimum mean squared error was employed for signal estimation. The different approaches were compared, and LMMSE is observed to be superior as show in figure 6. This observation is consistent with the theory.

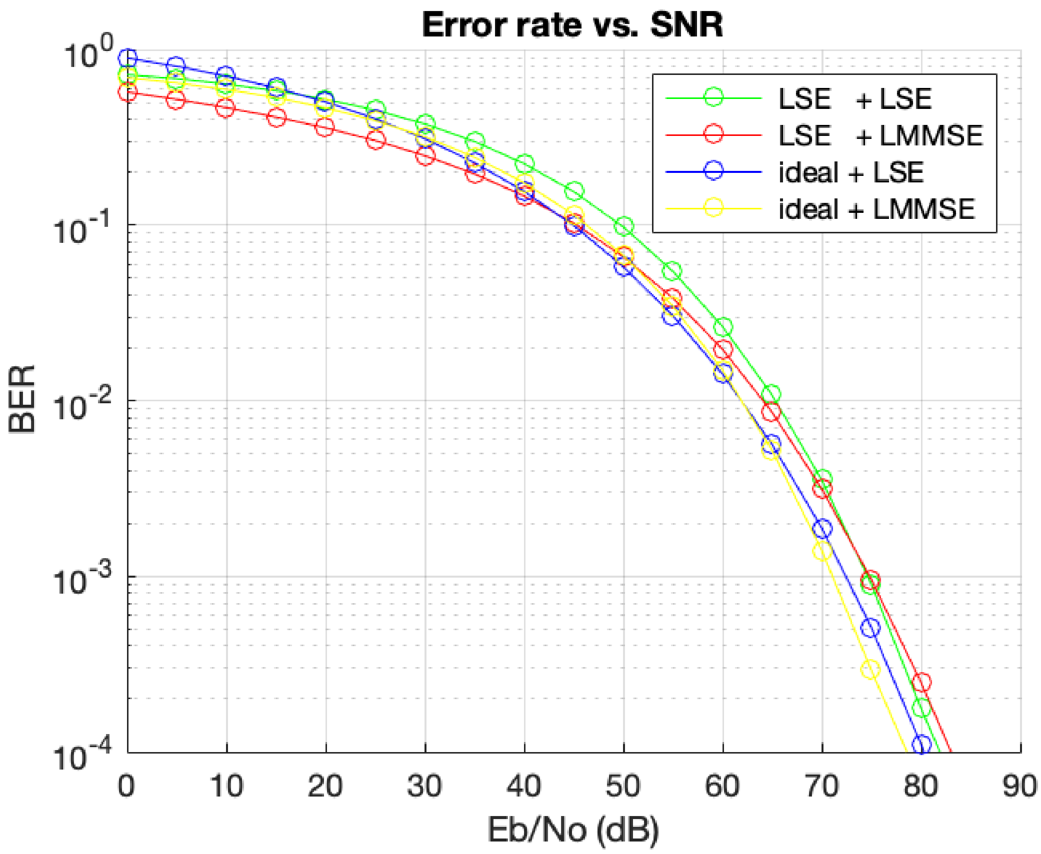

Figure 6: Comparison of the performance of the LSE and LMMSE approaches

4.5 Supplementary scripts

Multiple supplementary scripts were written for the easy visualization of the performance of the systems, and visual analyses of the same. Additionally, demo scripts were written to illustrate the impact of features such as the channel and signal estimation, as well as the modulation, and the channel estimation approach used. Furthermore, any comparison can be performed using the Grapher package written.

4.5.1 Plotter

The plotter package is a set of scripts that allow for the easy plotting and comparison of various different systems. The system can be implemented by simply changing the parameters listed in one file. The computations can be performed by running the Monte Carlo simulator script multiple times, after changing 2 lines before each run. The computed data can lastly be plotted by running the plotter script, after setting up the legend and the colour scheme.

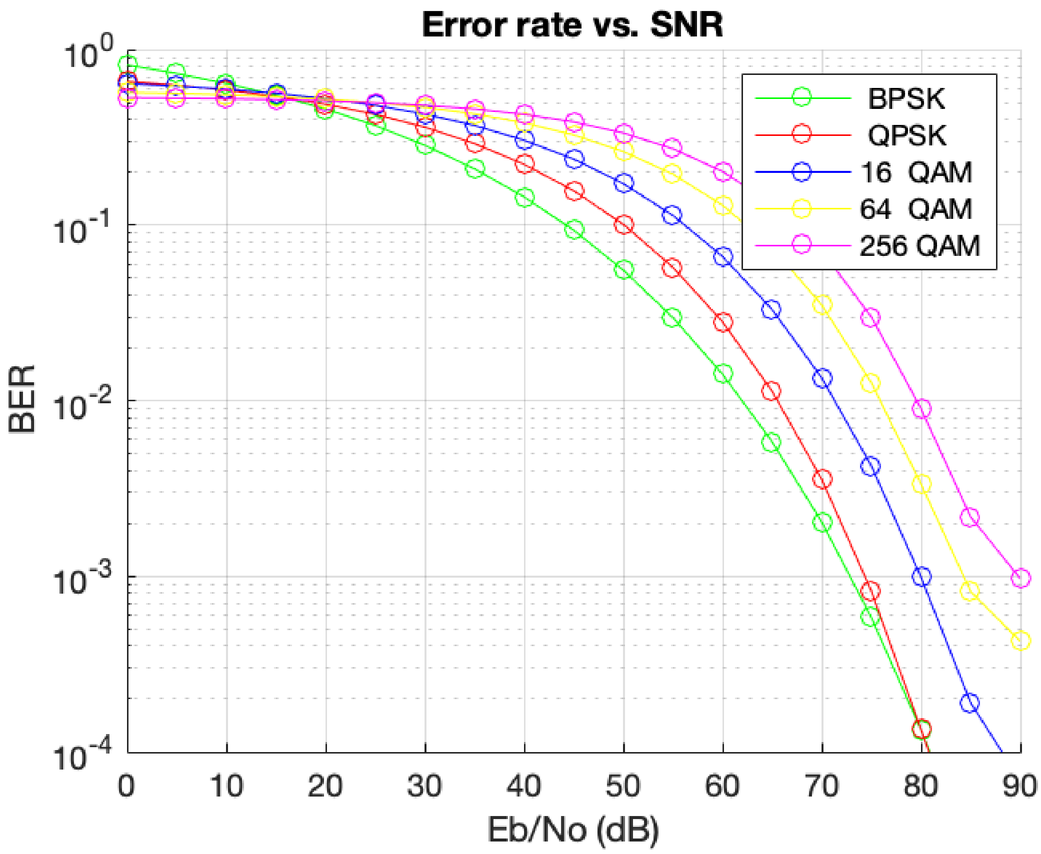

Figure 7: Comparison of the performance of BPSK, QPSK and QAM

Figure 7 shows a plot that compares different M-QAM implementations. 5 system configuration files were written, the difference between all of them being only a single variable. Then, the main file was run 5 times, once per system. The plotting was finally done by using the plotter file.

4.5.2 Burst Error Detector

The package has 2 scripts. The first script is a normal LDPC + MIMO-OFDM system, except that it records the channel matrix. The second script uses the recorded variation of the channel matrix to pinpoint the deep fades of all links, ascertain the failures of transmitters and receievers, and also prints the average number of errors in every OFDM frame.

The plot of the errors shows sharp spikes in places indicating that a burst error occurred. On examining the plot alongside the plot that marks the deep fades of the links, one can see the correlation visually. Large spikes tend to stem from deep fades, whereas small spikes tend to stem from the random generation of high values by the normal distribution.

The following plots were generated by one such Monte-Carlo simulation, and illustrate this correlation. As you can see, the deep fades align with the spikes in at least 2 place.

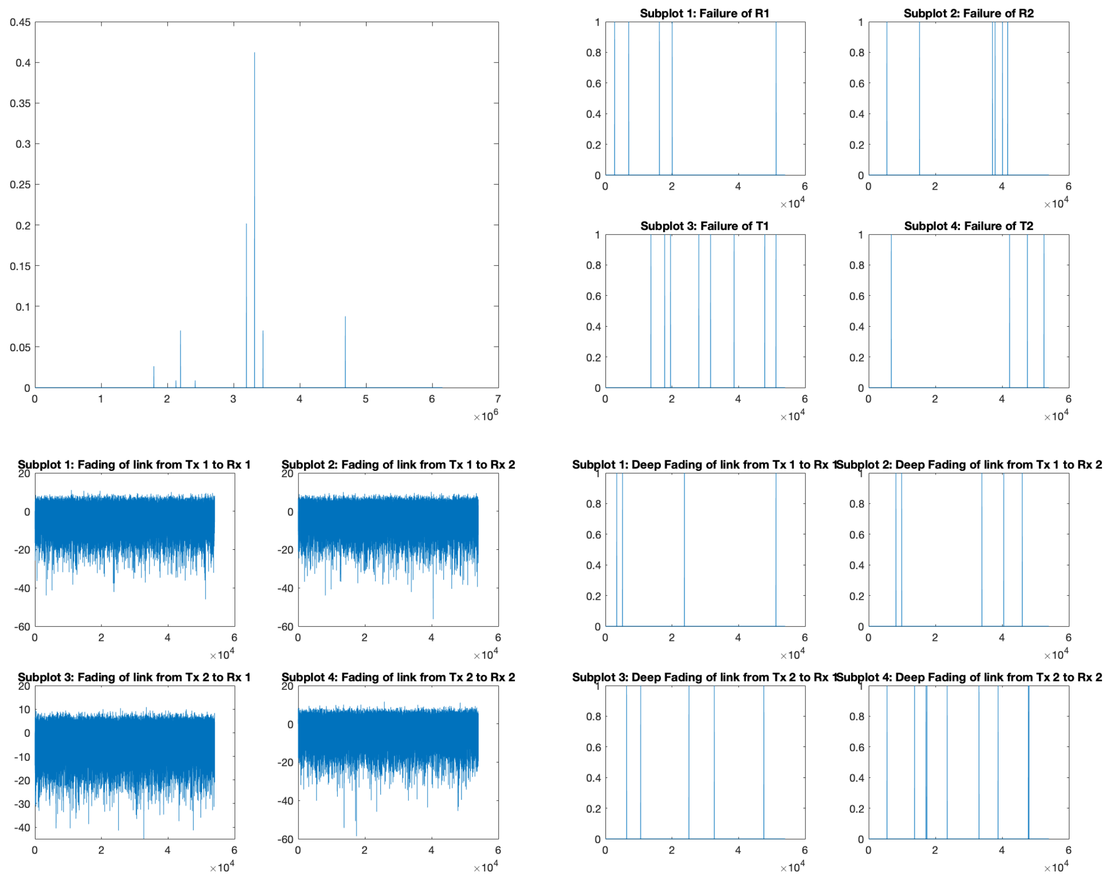

Figure 8: Plots illustrating the relationship between burst errors and deep fades

4.5.3 OFDM Spectrum plotter

A script was written to plot the individual subcarriers used in the OFDM part of the system. [20] The logic is simple. The symbols allocated to all subcarriers except one is zero'd out, and then the IFFT is run. This effectively gives us the frequency response of that isolated subcarrier. The process is repeated for all carriers, and a plot is generated. The plot can be compared with the resource allocation map for stronger insight into the system.

As mentioned earlier, all the parameters of the OFDM system can also be changed. The number of subcarriers, Number of Pilot carriers, Number of left and right guard carriers, and the cyclic prefix length can be changed easily. Additionally, pulse shaping can be toggled, and the windowing length can optionally be specified. Furthermore, the bandwidth of the system can also be altered with ease in this implementation.

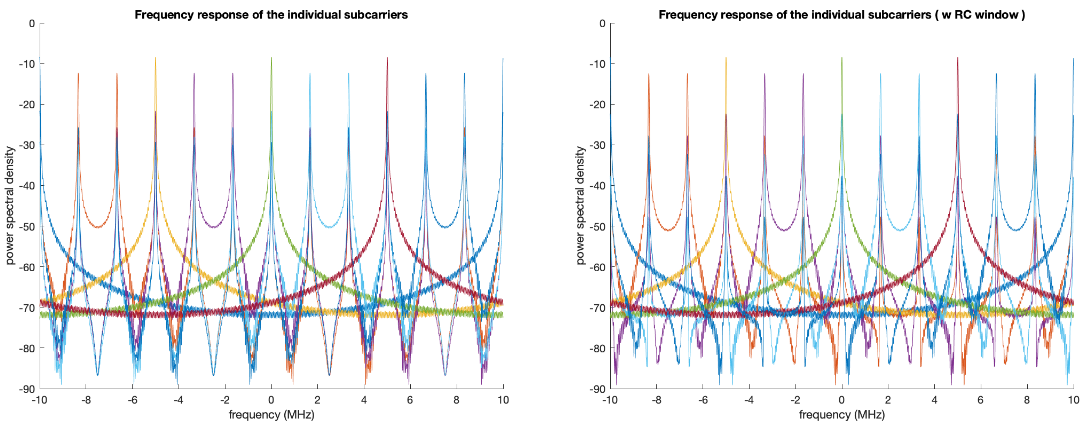

Figure 9: Frequency spectrum of the subcarriers used in OFDM

The scripts can be used for the visualization of the spectrum, and can potentially be used for visually tuning the same.

5 Results

The project aimed to simulate a MIMO-OFDM system that used polar codes, for HD video communication. The report listed the targets we have met, how we met the same, and the things we learnt along the way. It also highlights and presents the results achieved.

A system for streaming and a system for file transfer were implemented. The systems implement the default LDPC + MIMO-OFDM option we offer, and log the LDPC block numbers that have bit errors in them in a file. The streaming system also uses FFMPEG as a part of it. Additionally, utility scripts for plotting and comparing systems were written. Similarly, scripts for illustrating problems and concepts, and for visualizing the implementation were also written. Furthermore, the codes were tested and neatly documented. Lastly, the entire system is customisable by varying parameters

Video compression results

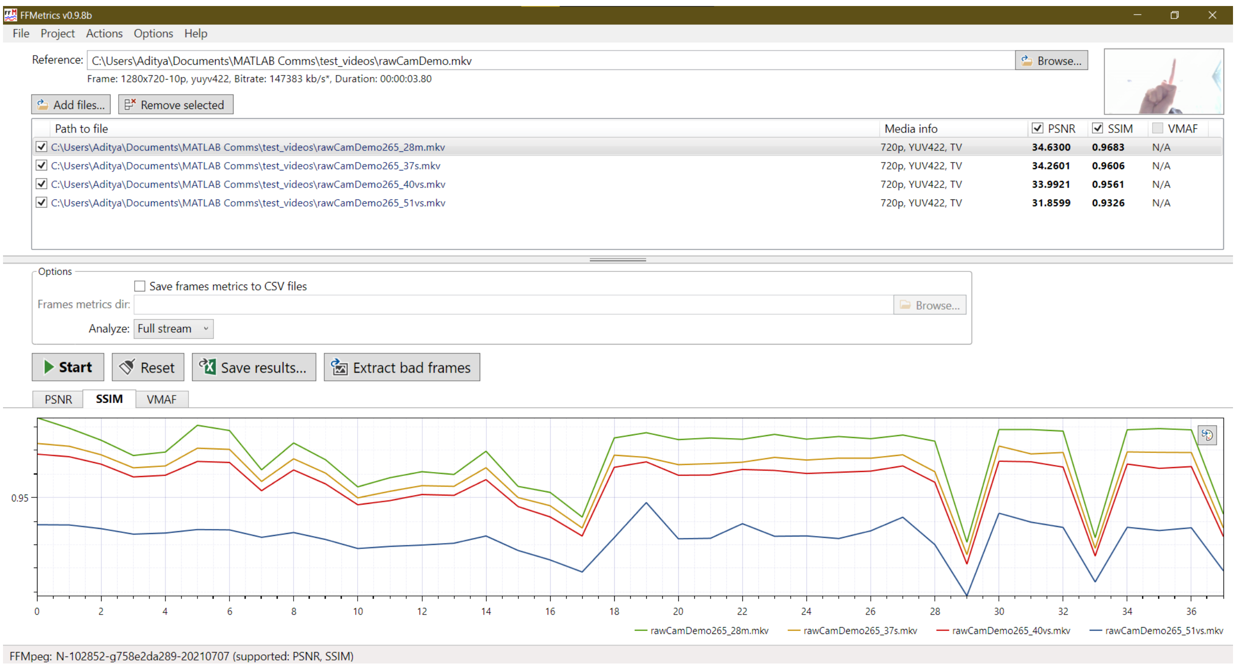

Figure 10: Comparison of different compression parameters

Figure 10 shows a comparison between a RAW video file and a few H.265 encoded versions of that file, each with different encoding parameters.

We started with a 3 seconds long raw video file which was 67MB in size. We were able to compress it down to 14KB, which translates to a compression factor of around 5000, by trading off compression time and video quality. However, we believe that a video with an SSIM below 0.96 is not close enough to the original for this project, and we should thus stick to parameters that give us an SSIM greater than 0.96. Videos with an SSIM greater than 0.96 have a compression factor of up to 1500.[21][22]

PSNR is not a reliable metric to compare how videos are perceived by humans and are thus not discussed in the context of video compression.

6 Conclusions

The burst error detector plots make it clear that LDPC, despite working for normal errors, cannot handle burst errors. However, the deep fading of Rayleigh fading channels inevitably leads to burst errors. Hence, an additional layer of BCH [23] is recommended, to retain the gain in performance offered by LDPC.

Furthermore, FFMPEG does well as a candidate for use for video compression. The compression rates achieved using FFmpeg makes HD video streaming viable.

7 Future Scope

The system designed has shown acceptable results thus far. However, there is a lot that can be done to improve the system to achieve competitive results.

A list of features that are essential for the hardware implementation that have not been implemented yet has been listed below :

- Frame synchronization
- Carrier synchronization
- Time domain signal generation
- Timing synchronization
- Translation to HDL code for use on hardware

A non-exhaustive list of potential improvements that can be made to our project is given below :

- Incorporation of the tunable parameter : coherence bandwidth for the Rayleigh fading channel
- The inclusion of ISI in the modelling
- VBLAST can be used for signal estimation, since it will offer superior performance
- Frequency interpolation can be done during channel estimation
- Optimal pilot carrier selection can be performed for all cases. [24]
- OSTBCs can be incorporated into the system.
- A protocol can be implemented for file transfer to request replacements for corrupted packets
- Polar Codes can be implemented for better BER results [25],[26],[27]
- Code generation can be used for speeding up the simulation [28]

8 Acknowledgements

We thank A.E.E. Naveen Reddy and Dr Rahul Singhal for all their guidance and support throughout the project. We also thank BITS and M.C.E.M.E. for providing us with such a rich opportunity.

9 References

[1] *usrp n200/n210 networked series*. URL: https://www.ettus.com/wp-content/uploads/2019/01/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf.

[2] *USRP B200/B210 Bus series*. URL: https://www.ettus.com/wp-content/uploads/2019/01/b200-b210_spec_sheet.pdf.

[3] Debashri Roy et al. "Adaptive Video Encoding and Dynamic Channel Access for Real-time Streaming over SDRs". In: *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. 2018, pp. 1–9. DOI: 10.1109/PCCC.2018.8710795.

[4] H. Bolcskei. "MIMO-OFDM wireless systems: basics, perspectives, and challenges". In: *IEEE Wireless Communications* 13.4 (2006), pp. 31–37. DOI: 10.1109/MWC.2006.1678163.

[5] ANDREW THANGARAJ. *Electronics & Communication Engineering - NOC:LDPC and Polar Codes in 5G Standard*. 2019. URL: https://nptel.ac.in/courses/108/106/108106137/.

[6] Aggelos K Katsaggelos. *Fundamentals of Digital Image and Video Processing*. 2016. URL: https://www.coursera.org/learn/digital.

[7] Meenakshi Rawat. *Electronics and Communication Engineering - NOC:Basics of software-defined radios and practical applications*. 2017. URL: https://nptel.ac.in/courses/108/107/108107107/.

[8] Travis F. Collins. *Software-defined radio for engineers*. Artech House, 2018.

[9] Aditya K Jagannatham. *Electronics & Communication Engineering - Advanced 3G and 4G Wireless Mobile Communications*. 2012. URL: https://nptel.ac.in/courses/117/104/117104099/.

[10] Vincent Savaux and Yves Louët. "LMMSE channel estimation in OFDM context: a review". In: *IET Signal Processing* 11.2 (2017). DOI: 10.1049/iet-spr.2016.0185.

[11] Aditya K Jagannatham. *NOC: Bayesian/ MMSE Estimation for Wireless Communications -MIMO/ OFDM*. 2016. URL: https://nptel.ac.in/noc/courses/noc16/SEM2/noc16-ec15/.

[12] Aditya K Jagannatham. *NOC: Estimation for Wireless Communications:MIMO-OFDM Cellular and Sensor Networks*. 2017. URL: https://nptel.ac.in/noc/courses/noc17/SEM2/noc17-ee19/.

[13] MATLAB. *Introduction to MIMO Systems*. URL: https://in.mathworks.com/help/comm/ug/introduction-to-mimo-systems.html.

[14] mhanney. *Stream webcam and audio source from PC to remote IP address using mpegts*. URL: https://gist.github.com/mhanney/dc329b1d9bff032b3600a1f7b44099e5

[15] Ali A. Nasir et al. "Timing and carrier synchronization in wireless communication systems: a survey and classification of research in the last 5 years". In: *EURASIP Journal on Wireless Communications and Networking* 2016.1 (2016). DOI: 10.1186/s13638-016-0670-9.

[16] MATLAB. *DVB-S.2 System Simulation Using a GPU-Based LDPC Decoder System Object*. URL: https://in.mathworks.com/help/comm/ug/dvb-s-2-system-simulation-using-a-gpu-based-ldpc-decoder-system-object.html.

[17] MATLAB. *Simulation Acceleration Using GPUs*. URL: https://in.mathworks.com/help/comm/ug/simulation-acceleration-using-gpus.html.

[18] Yong Cho. *MIMO-OFDM Wireless Communications with MATLAB*. Wiley, 2010.

[19] Andrea Goldsmith. *Wireless Communications*. USA: Cambridge University Press, 2005. ISBN: 0521837162.

[20] Krishna M Sankar. *spectrum of an OFDM transmit*. 2021. URL: https://in.mathworks.com/matlabcentral/fileexchange/19189-spectrum-of-an-ofdm-transmit.

[21] *Video Compression*. URL: https://en.wikipedia.org/wiki/Data_compression#Video.

[22] slhck. *Video quality metrics*. URL: https://superuser.com/a/338734.

[23] *BCH encoder in MATLAB*. URL: https://in.mathworks.com/help/comm/ref/bchenc.html.

[24] Heidi Steendam. "How to select the pilot carrier positions in CP-OFDM?" In: *2013 IEEE International Conference on Communications (ICC)*. 2013, pp. 3148–3153. DOI: 10.1109/ICC.2013.6655027.

[25] Erdal Arikan. "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels". In: *IEEE Transactions on Information Theory* 55.7 (2009), pp. 3051–3073. DOI: 10.1109/TIT.2009.2021379.

[26] Marwan Dhuheir and Sıtkı Öztürk. *Implementation of Polar Codes in 5G Systems with Different Waveform Modulations by Using USRP*. 2019. DOI: 10.16984/saufenbilder.536540.

[27] EventHelix (2019 April 29). *Polar codes*. URL: https://medium.com/5g-nr/polar-codes-703336e9f26b.

[28] *Code Generation in MATLAB*. URL: https://in.mathworks.com/help/coder/?s_tid=srchbrcm.

[29] Jingyi Zhao et al. "Research and Implementation of 2x2 MIMO-OFDM System with BLAST Using USRP-RIO". In: *Proceedings of the International Conference on Promotion of Information Technology (ICPIT 2016)*. Atlantis Press, 2016/08, pp. 89–94. ISBN: 978-94-6252-219-0. DOI: https://doi.org/10.2991/icpit-16.2016.14. URL: https://doi.org/10.2991/icpit-16.2016.14.

[30] Jaime Luque Quispe, Jackelyn Tume Ruiz, and Luis G. Meloni. *SDR implementation of an OFDM transmitter and receiver*. URL: https://www.ccp-br.fee.unicamp.br/JP3I_website/anteriores/02jp3i/papers/Paper015.pdf.

[31] Giuseppe Baruffa, Luca Rugini, and Paolo Banelli. "Design and Validation of a Software Defined Radio Testbed for DVB-T Transmission". In: *Radioengineering* 23 (Apr. 2014), pp. 387–398.

[32] Asha Mariam Iype and Shashanka C D.

[33] T. M. Kavyashree, L. Jyothi, and C. UmashankarShetty. "Design and Implementation of 4 G LTE Components-eNodeB and UE on SDR platform using srsLTE". In: 2016.

[34] Ufuk Tamer. "TESTBED IMPLEMENTATION OF A MIMO-OFDM DATA LINK AT SUB 6 GHz FREQUENCY BANDS USING AN SDR PLATFORM". PhD thesis. Aug. 2018.

[35] Xiao-Wei Tang and Xin-Lin Huang. "A Design of SDR-Based Pseudo-analog Wireless Video Transmission System". In: *Mobile Networks and Applications* 25.6 (2020), pp. 2495–2505. DOI: 10.1007/s11036-020-01592-6. URL: https://doi.org/10.1007/s11036-020-01592-6.

[36] Prasidh Ramabadran. *OFDM Transmitter and Receiver with BPSK baseband, RF up-down conversion*. 2021. URL: https://www.mathworks.com/matlabcentral/fileexchange/57494-ofdm-transmitter-and-receiver-with-bpsk-baseband-rf-up-down-conversion.

[37] Amod Anandkumar. *Developing and Prototyping Next-Generation Communications Systems Video*. Apr. 2017. URL: https://in.mathworks.com/videos/developing-and-prototyping-next-generation-communications-systems-1502972744022.html.

[38] Kai Niu et al. "Polar codes: Primary concepts and practical decoding algorithms". In: *IEEE Communications Magazine* 52.7 (2014), pp. 192–203. DOI: 10.1109/MCOM.2014.6852102.

[39] Marwan DHUHEIR and Sıtkı ÖZTÜRK. "Polar Codes Applications for 5G Systems". In: *Journal of Institue Of Science and Technology*. 2018th ser. 34.3 (Dec. 2018).

10 Appendix

The repository linked to below contains all the code written and detailed documentation for the code referenced throughout this report.

GitHub Repository

For in case of access issues, please use the google drive linked to below :

Google Drive

Note that the google drive may not be up to date with the GitHub repository.