WAVEFORM OPTIMIZATION IN 5G

Advisor : Dr Faisal Shafait
Co Advisor: Dr Naveed Ahmed

Shaheer Mehmood
Hadi Askari

CONTENTS

INTRODUCTION

- Modulated waves used in communication systems get distorted by channel impairments such as AWGN, Frequency Offset, IQ Imbalance etc. Existing methods of identifying and removing these distorted elements are pilot based statistical methods. We propose a neural network based method of identifying these channel impairments in 5G OFDM waves and creating a model that is best capable of removing them based on the channel state information.

PROBLEM STATEMENT

- Create a model to identify the myriad of channel impairments that can arise during data transmission, after identification the next step is to actively remove these waves with state of the art accuracy while preserving actual information sent.

PROJECT OBJECTIVES

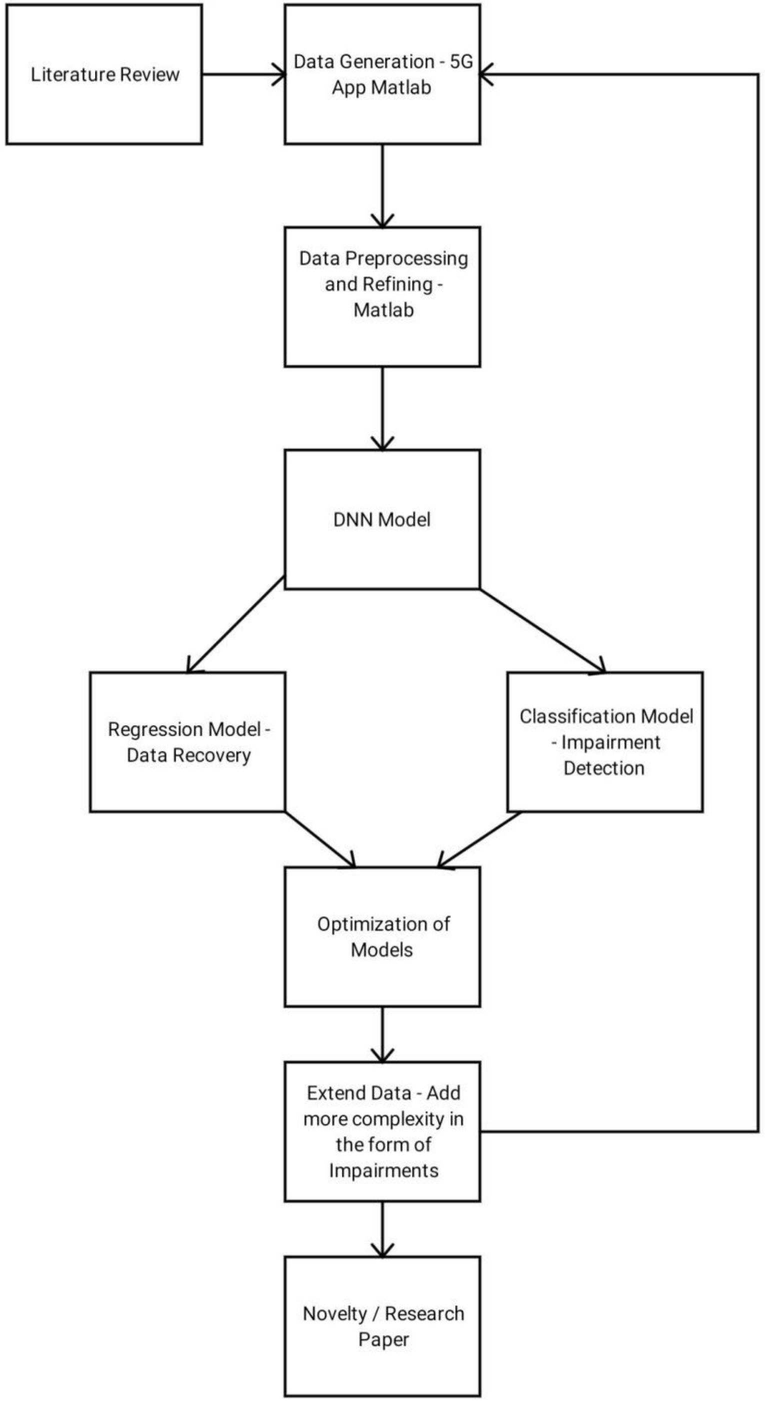
- Our first objective was to understand how to model and simulate the different impairments. After recreation we will create a classification model to identify the impairments in the waves.
- After classification of the impairments we will move on the cleaning our waveforms by removing them and recovering our original transmitted 5G wave.
- Finally we will create a robust model capable of real time detection and removal of these channel impairments.

ALGORITHMS / TECHNOLOGY

- MATLAB 5G NR Toolbox for Dataset generation, simulation and modeling.
- Keras/Tensorflow and Python for Deep Learning.
- Algorithms: Deep Convolutional Neural Networks and Radio Transformer Networks.

PROJECT REQUIREMENTS

- GNU Radio Software for further modeling
- GPU on Google Colab for Training.
- Excessive storage space required for data.



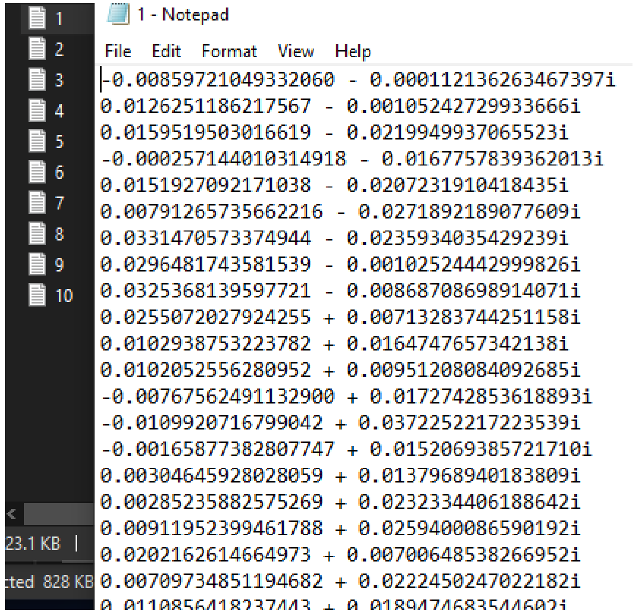
FLOW DIAGRAM

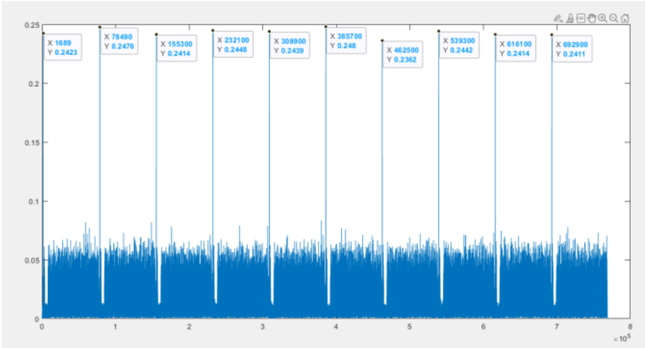
We are following this methodology in order to proceeded with our problem statement.

PROGRESS (START TILL 15.10.20)

- Literature review
- Remade 4G – OFDM model and studied results
- Redefined the problem statement
- Data Generation
- Used 5G toolbox from MATLAB
- Generated 5kHz waves
- Added white noise (10-30dBs)
- Extracted PSS1 (instead of entire wave) to start with
- Made multiple samples

SAMPLE DATA GENERATION





PROGRESS (15.10.20 TILL 31.12.20)

- Read several papers of Deepsig.io and understood how classification models can be applied to modulation networks and how Radio Transformer Networks are made.
- Increased number of samples from 50 to 300. From 0-30dBs.
- Constructed a regression CNN model but was discarded due to a change in approach.
- Constructed and implemented a classification CNN model with 62.6% accuracy. Currently working on trying to optimize results.

MODEL AND RESULTS

```
Epoch 117/200
8/8 [==============================] - 0s 18ms/step - loss: 0.0358 - accuracy: 0.9708 - val_loss: 4.4393 - val_accuracy: 0.6000
Epoch 118/200
8/8 [==============================] - 0s 18ms/step - loss: 0.0361 - accuracy: 0.9833 - val_loss: 4.4616 - val_accuracy: 0.6000
Epoch 119/200
8/8 [==============================] - 0s 18ms/step - loss: 0.0353 - accuracy: 0.9708 - val_loss: 4.5114 - val_accuracy: 0.6167
Epoch 120/200
8/8 [==============================] - 0s 20ms/step - loss: 0.0348 - accuracy: 0.9792 - val_loss: 4.5390 - val_accuracy: 0.6167
Epoch 121/200
8/8 [==============================] - 0s 23ms/step - loss: 0.0353 - accuracy: 0.9750 - val_loss: 4.5865 - val_accuracy: 0.6000
Epoch 122/200
8/8 [==============================] - 0s 20ms/step - loss: 0.0351 - accuracy: 0.9750 - val_loss: 4.6040 - val_accuracy: 0.6167
Epoch 123/200
8/8 [==============================] - 0s 18ms/step - loss: 0.0340 - accuracy: 0.9792 - val_loss: 4.5930 - val_accuracy: 0.6167
Epoch 124/200
8/8 [==============================] - 0s 18ms/step - loss: 0.0352 - accuracy: 0.9792 - val_loss: 4.5631 - val_accuracy: 0.6167
Epoch 125/200
8/8 [==============================] - 0s 19ms/step - loss: 0.0353 - accuracy: 0.9708 - val_loss: 4.5641 - val_accuracy: 0.6167
Epoch 126/200
```

```
dr = 0.5
model = Sequential()
model.add(Reshape(out_shape, input_shape = in_shape))
model.add(ZeroPadding2D((0, 2), data_format = 'channels_first'))
model.add(Conv2D(256, (1, 3), padding = 'valid', activation = "relu", name="conv1", kernel_initializer='glorot_uniform', data_format="channels_first"))
model.add(Dropout(dr))
model.add(ZeroPadding2D((0,2), data_format = 'channels_first'))
model.add(Conv2D(80, (2, 3), activation="relu", name="conv3", padding="valid", kernel_initializer="glorot_uniform", data_format="channels_first"))
model.add(Dropout(dr))
model.add(Flatten())
model.add(Dense(256, activation="relu", name="dense1", kernel_initializer="he_normal"))
model.add(Dropout(dr))
model.add(Dense(6, name="dense3", kernel_initializer="he_normal", activation = 'softmax'))
model.add(Reshape([len(list_labels)]))
```

CHALLENGES

- A lack of datasets and online material for Deep Learning in Wireless Communication has been a huge thorn in our side. A lot of approaches lead to dead ends and reference material isn't readily available to help in problem solving.
- Lot's of literature review needs to be done to figure out what works and what doesn't due to the novelty of the field and lack of set procedures. Approximately 50 papers reviewed and 4,5 models constructed before the latest one was agreed upon.

TIMELINE

MILESTONES	TIMELINE
Literature Review	Complete
Problem Identification	Complete
Data Generation (awgn)	Complete
CNN Model (awgn)	Complete
Data Generation (all impairments)	Feb – March 2021
CNN Model (all impairments)	March 2021
LSTM/RTN Model	March – April 2021
Real Time Deployment	April 2021
Results / Publication	April – May 2021

CONCLUSION

- We hope to improve our results on our model and then move towards creating a larger dataset comprising of all the known channel impairments.
- After implementing that we will move towards a Radio Transformer Network architecture or an auto-encoder based architecture for further optimization
- Once we have all the results and optimized models, we will work on publication by structuring our findings, comparing our models and then publishing respective results.

REFERENCES

[i] Convolutional Radio Modulation Recognition Networks: https://arxiv.org/pdf/1602.04105

[ii] Radio Transformer Networks: https://arxiv.org/abs/1605.00716