EE/CprE/SE 491 - sddec22-13
Simultaneous Call Transmission (SCT)
Week 8

March 28th, 2022 - April 3rd, 2022
Client: Collins Aerospace
Faculty Advisor: Dr. Andrew Bolstad

Team Members

- Hani El-Zein - *Digital Signal Processing Lead and Research*
- Sullivan Jahnke - *Project Manager, Lead Communicator, and Machine Learning Lead*
- Tyler Mork - *Reports, Communicator, Co-Webmaster, and Communication Systems Co-Lead*
- Json Rangel - Reports, *Communicator, Co-Webmaster, and Communication Systems Co-Lead*
- Austin Rognes - *Research, MATLAB Lead, and Co-Webmaster*

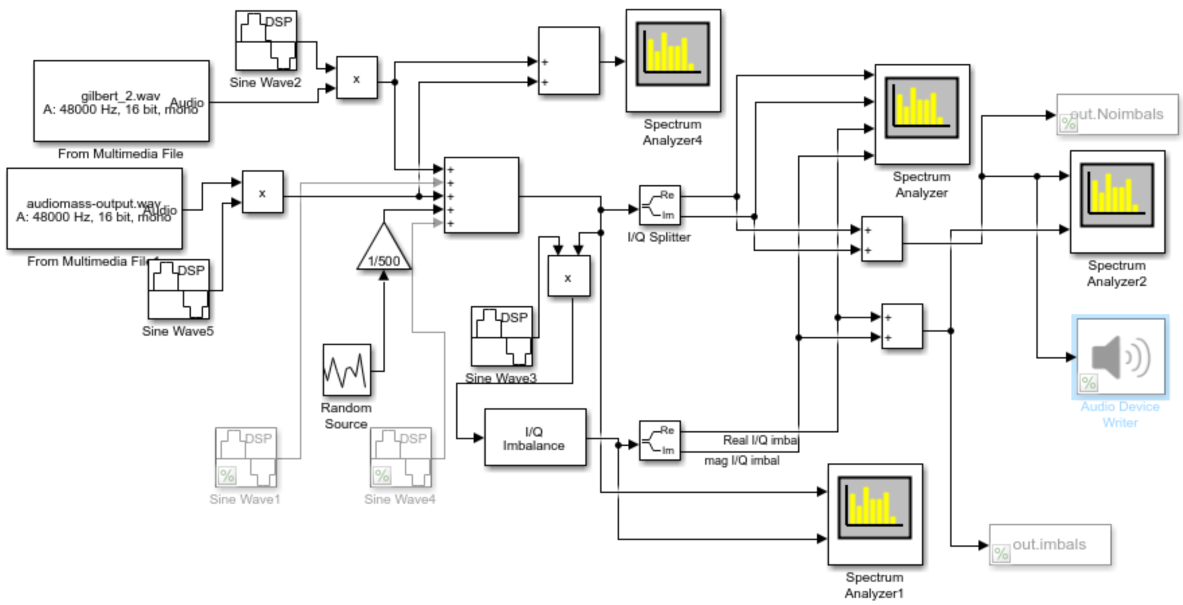
Week 8 Summary

Week 8 consisted of further implementing signal features into our simulation for data extraction later. An equivalent complex baseband model shown in Figure 1 was constructed with foreseen real world variances such as frequency and phase shifts, noise and possibly DC offset. DC offset is commonly removed in modern receivers today and won't be a needed feature to this simulation.

Further improvements to the simulation will include doppler effect and reflection, and also the ability to create instance variables for the different features to be randomly generated over time to create multiple sets of training data for the ML algorithm to learn off of. Besides the complex baseband model, we have also constructed a RF signal modulation and demodulation simulation for reference to our complex baseband. For errors seen in the demodulation process, we attempt to model them in the complex baseband equivalent to simplify as much as possible.

Attempts have been made to determine ways to model phase variations between simulations. An advised method is to view the magnitude and phase components of the I and Q data to distinguish a 2-Dimensional relationship with phase. From there, we may be able to accurately distinguish multiple signals present.

Figure 1



On the software side, we have decided the initial layout of our neural network. We are going to begin with 3 hidden layers, with the activation function being ReLU. For the output layer, we will use a sigmoid activation function to classify whether an SCT event occurred. As for optimization and loss, we will begin with the Adam optimization function and the binary cross entropy loss function (typically used for classification problems like this). We are currently working on determining the input layer size and how we will handle the complex (or other various types of) data as opposed to real data. Once we can get this initial network to work for real data, we can start optimizing for better accuracy by tweaking things such as the layout, the functions we use and the formatting of the data.

Past Week Accomplishments

The group has successfully created a complex baseband model for input data the Machine Learning Algorithm. This model includes as of to date, multiple signals with what would be considered different carrier signals at complex baseband, the summation of random source noise, frequency shifts due to receiver demodulation, and phase shifts due to receiver I/Q imbalance or other errors. The adjusted signal with added noise errors is then broken down into its real and imaginary components to be extracted as the Q and I phase of the demodulated signal. Another unadjusted signal is then referenced to the adjusted signal to observe desired alterations. We also began to experiment with sending the I and Q phase data extraction to the Matlab Workspace for other uses and sending it to a file writer for import to the Machine Learning Algorithm.

As stated in the summary, we chose the loss, activation, and optimization formulas for the neural network. It is now implemented in python with the code shown below.

```
#Create the neural network
def create_nn_model():
    model = keras.Sequential([
        keras.layers.Dense(32, activation=tf.nn.relu,
                           input_shape=train_data.shape[1]),
        keras.layers.Dense(8, activation=tf.nn.relu),
        keras.layers.Dense(4, activation=tf.nn.relu),
        keras.layers.Dense(1, activation ="sigmoid")
    ])
    optimizer = tf.keras.optimizers.Adam()
    model.compile(loss='binary_crossentropy',
                  optimizer=optimizer,
                  metrics=['accuracy'])
    model.summary()
    return model
```

Now that we have the initial layout for the NN complete, we attempted a few different input formats to see if we could get this implementation to work, but were unable to run it without getting errors. The work we will be doing this week (said below) will be addressing this issue.

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Hani El-Zein		2	22
Sullivan Jahnke	Applied initial loss, activation, and optimization functions to the neural network. Created a base neural network format. Worked on determining input layer size. Worked on formatting the input so that it can be used by the neural network.	6	28.5
Tyler Mork	Further built an equivalent complex baseband model. Introduced frequency offsets, phase offsets, and noise into the input signal.	4	30.5
Json Rangel	Added more meetings with Dr. Bolstad to schedule and coordinated respective room reservations. Continued to work on an I-Q Demodulation system within Simulink. Also modified previous simulations to include I-Q demodulation. Finally, did additional research on quadrature amplitude modulation and baseband modulation.	4.5	30
Austin Rognes	Worked on formatting machine learning input data and taking in an array of arrays to slice each radio wave into fragments.	2	22

Plans for Upcoming Week

The team plans to begin trial runs of the interface between simulation data and the Machine Learning Algorithm. We plan to introduce items slowly at first such as a single signal with noise, and observe whether the algorithm functions properly. From there, we will continuously make the input data more and more complex.

Furthermore, the equivalent complex baseband model will require additional signal error features such as Doppler Effect and Reflection introduced into the model. Research will be required on how to effectively model such phenomena. Further research will be required to accurately model the ranges in which these real world errors can be observed. For example, phase offset would be ranges around the unit circle whereas the frequency offset would deviate from the 0 Hz baseband center. As to how far each of these deviations reach will be vital information to accurate simulation data. Additional signal processing can be observed through the simulation to obtain ideal criteria such as low pass filtering and downsampling, but we are uncertain as to whether hardware “after” the I and Q phase decimation is accurate and should be considering the Machine Learning Algorithm to contain further pre-processing capability. More will need to be looked into as to if the pre-processing capability enhances efficiency of the algorithm overall.

The communications portion of the team intends to have a first draft of our IQ baseband demodulation system ready for testing. This system will include realistic components, such as white noise, doppler effect, and reflection.

We plan to be able to have the neural network fully trainable with any data that we want. We need to come up with a consistent format to preprocess the data with. We also want to be able to implement the labels to go along with the formatted data. We hope to start getting some output this week!