**Tin Vo and Nathan Ngo**

**Final Project Report**

***Introduction***

For this final project, we used Matlab to implement a modulator and demodulator for NASA’s APT data. The modulator accepted a JPEG image with a fixed width of 909 pixels and produced a .wav file that is amplitude modulated using PAM. The demodulator accepted a .wav file and wrote a grayscale image to JPEG file, where the image presented “A” and “B” subimages next to each other.

We also implemented an impairment which accepted a .wav file and added AWGN based on the specified signal-to-noise power ratio in dB.

***Procedure***

When implementing the modulator, we first read in the JPEG and converted it to grayscale. We then created the sync A and sync B frames using the provided APT synchronization details. Next, we concatenated all the frames together into one vector and replicated the values 4 times to create 4 samples per word. Finally, we modulated the signal by multiplying it with a cosine wave and resampled the results to 16 kHz before writing it to a .wav file.

For the demodulation, we performed similar actions as in modulation, but in the reverse order. After reading in the .wav file, we resampled it to 16.64 kHz and performed the hilbert transform. We squared the imaginary component added it to the square of the input signal. Finally, we took the square root to get back the demodulated signal. Since this demodulated signal still had a higher frequency component, we ran it through a low-pass filter with a cut-off frequency of 2.4kHz.

We then recreated the sync A vector and used convolution to find the sync A frame in the input signal. Once this was found, we simply looped through and extracted the data A and data B frame and placed it on a new line of a resulting matrix that stores the data for the image.

For the impairment, we simply read in a .wav file and calculated the signal’s power. We then used the function awgn() and inputted the signal’s power and desired SNR in decibel to add AWGN to our signal.

***Results***

We were able to successfully implement the demodulator, modulator, and impairment. When we inputted the given .wav files, we were able to demodulate the signals and write the data to a JPEG. When inputted the JPEG file, we were able to modulate it to a .wav file and then also demodulate it back into a JPEG file. When demodulating the signals after running it through the impairment, we saw noise was introduced into the image.