Project ICARUS

In this repository, I have put in all the python implementation for a receiver. This receiver ‘receiver\_final.py’ will demodulate and decode BPSK signals that are affected by different impairments like timing offsets, SNR inconsistencies, error correction coding and Doppler shifts.

The final version passes the all performance thresholds, as mentioned in the challenge guidelines. The coded, and Doppler-shifted signals were decoded without any errors.

Approach

In my version of the receiver, I have coded in each phase as a different function, and the program calls a certain function to handle the particular impairment for each phase of the dataset.

For phase 1, I have used Matched Filtering. This is to maximize the signal-to-noise ratio (SNR) at the sampling instant. I do this by convolving the incoming signal with a time-reversed version of the pulse shape. This was a good experience for me, as we just learnt matched filtering in the subject, Signals & Systems, and I was able to put that theory into practice here.

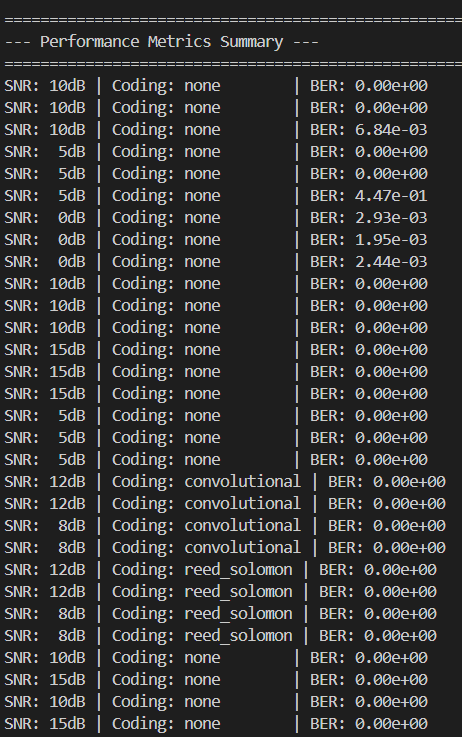
A problem that occurred was, there was a significant and unknown timing offset at the start of each transmission. To tackle it, I used something called Coarse Packet Detection, and Fine Peak Detection.

For phase 2, the given SNR was incorrect. I then calculated the estimated SNR by individually calculating the Signal power and the noise power. To extract the noise, I took the decoded bits and converted them back to the ideal symbols, and then ‘noise = sample – ideal’. Lastly I converted the estimated SNR in dB.

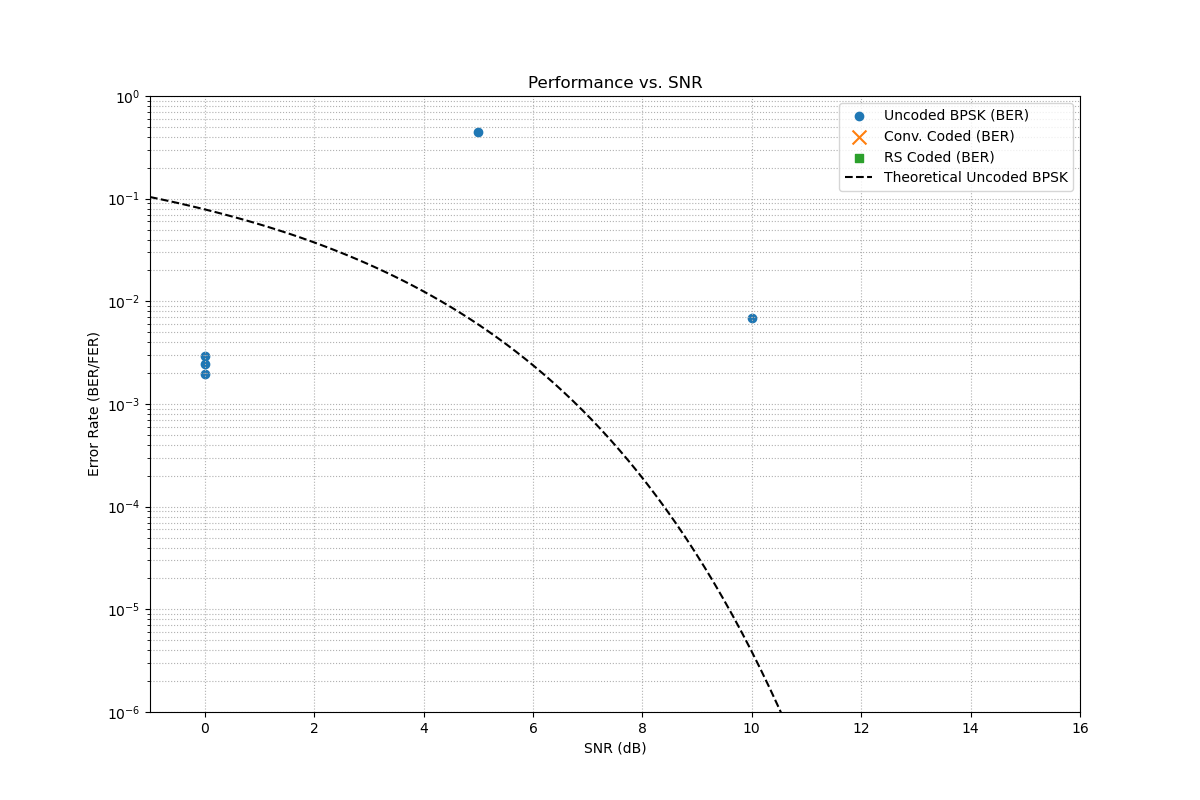
For phase 3, there were high BERs, which required error correction. I used a soft-decision Viterbi Decoder. I used 133, and 171 as the generator codes, and a constraint size of 7. The decoder runs on LLRs (Log-Likelihood ratios) instead of hard bits, as it gave better error correction. I also used Reed-Solomon (15, 11) Decoder. It was built over the Galois Field GF(2⁴). It implements Berlekamp-Massey, Chien search and Forney algorithms. These algorithms locate and then correct two errors per block.

For phase 4, the carrier frequency was offset. This is the Doppler shit, which arises due to relative motion between the transmitter and receiver. There was also a 180-degree phase ambiguity. I solved this by using Doppler Compensation and then Carrier Phase Recovery. For the doppler compensation, the complex signal is squared so as to create a tone at double the offset. I then used an FFT to locate the tone. The signal is then mixed back into the baseband. After the doppler correction, I use another squaring technique to find the remaining average phase offset. Lastly I rotated the constellation to align with the real axis. This resolves the phase ambiguity.

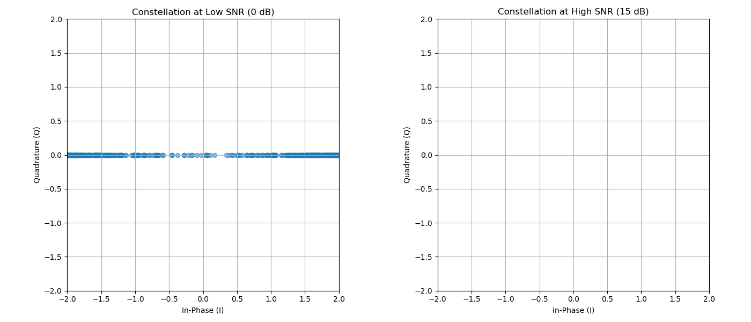
Performance Results



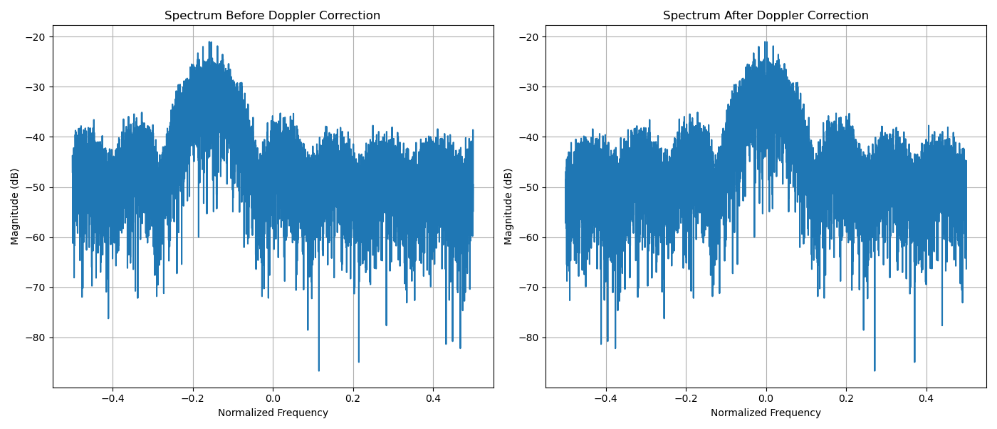
BER vs. SNR Performance



Constellation Diagrams



Doppler Compensation



A special shoutout to Dr. Ajin S Nair, who brought me up to speed on the basics of modulation and demodulation of BPSK signals. He is also my Signals & Systems professor. He gave a crash course on all these topics as I am new to them, and told me how I should approach the issue. I am indebted to him.