1 Package Description

The package for the project is broken up into individual files each implementing different functionalities. The functional parts of the project are all gathered under the package “tools”. “tools” is further subdivided into relevant sections. “data generation” handles the generation of the random and bandpass data. “multirate” holds functions that generate the mother filter, the synthesis and analysis filters. “scrambling” hold the code for the single and multi-channel scramblers with supporting classes/functions such as the LFSR. The files labeled “partXXX.m” located in the main section work to utilize the contents under “tools” to generate the outputs needed in the report.

2 Polyphase Decomposition

2.1 Filter Response visualization

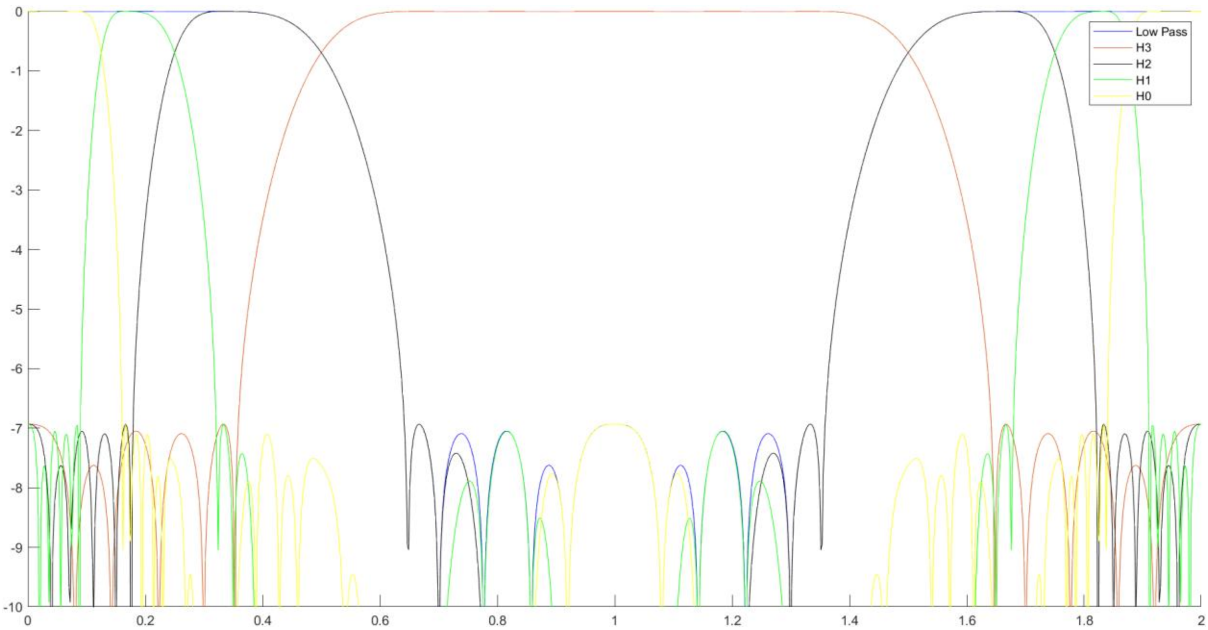

Figure 1: Filter bank response

The magnitude of the frequency response of the filters banks were visualized as given in Figure 1. The responses were generated from the closed form expression for H(z) and H(-z). This was accomplished in the file named “part1_plot.m”. From the plot it can be observed that the entire frequency response is broken up and fully covered by the different legs of the filter bank. As the filters are derived from imperfect low-pass filters, there is a slight loss of gain as one filter’s peak ends and another begins. This will be visible in the next parts.

2.2 Filter implementation

The filter banks are implemented without using polyphase decomposition in the time domain to maintain readability/interpretability/maintainability, at the cost of performance. The methods under “tools/multirate” are used to implement the filter banks.

2.3 Discussion on F1 choice

For Option 1 where F_1 is chosen as -H_1, the system is alias-free. As the mother filter is a linear phase filter, we know that it cannot achieve perfect reconstruction. The group delay of each of the mother filters are 31-1/2 = 15. Since the topmost 2 legs (V0 and V1) passes through 3 of these filters, the group delay is 45. V2 passes through 2 and has group delay 30 and the bottommost has group delay 15.

For Option 2 where F_1 is chosen as H_1, the system is not alias-free. As the mother filter is a linear phase filter, we know that it cannot achieve perfect reconstruction. The group delay of each of the mother filters are 31-1/2 = 15. Since the topmost 2 legs (V0 and V1) passes through 3 of these filters, the group delay is 45. V2 passes through 2 and has group delay 30 and the bottommost has group delay 15.

2.4 Effects on F_1 on MSE

	F_1 = -H_0(-z)	F_1 = H_0(-z)
Random Data	0.071511	0.019233
Bandpass Data	0.062474	0.0014975

3 Scrambling

3.1 Implementation

As the scrambling and descrambling process is functionally equivalent, the scrambling code is reused as the descrambler. The data is scrambled before it enters the transmission channel model and is descrambled after it exits. The LFSR performs n-1 XOR operations per shift, where n is the number of bits used for the feedback. This is equal to 2 for the inphase and 4 for the quadrature component. The bit sequence generated requires 1 multiplication for the normalization and 1 more for multiplying with the corresponding I or Q components, for a total of 4 multiplications per timestep. The normalization involves 1 addition per I and Q component and 1 more to add the two components together for a total of 3 per timestep. For a frame length of 1024, V3 is 512, V2 is 256 and both V1 and V0 are 128 samples long for a total of 1024 samples. Thus the scrambling process involves 1024*4 = 4096 and 3*1024 = 3072 additions. The descrambling process requires equally many operations for a grand total of 6144 additions and 8192 multiplications per frame. The type of the data/choice of F_1 does not impact this value

3.2 Effects of Scrambling on MSE

	F_1 = -H_0(-z)	Number of additions	Number of multiplications
Random Data	0.072443	6144	8192
Bandpass Data	0.062474	6144	8192

As expected, the scrambling process does not impact the MSE.

3.3 MSE vs SNR experiments

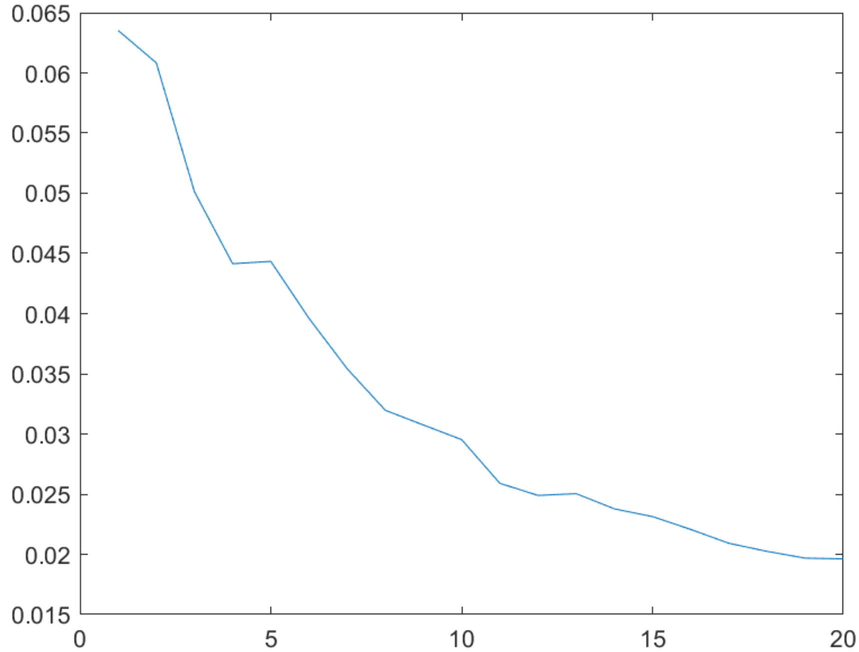

Figure 2: MSE vs SNR with no frequency offset

The MSE of the filter bank given a certain SNR of the noise in the transmission channel model with no frequency offset is given in Figure 2.

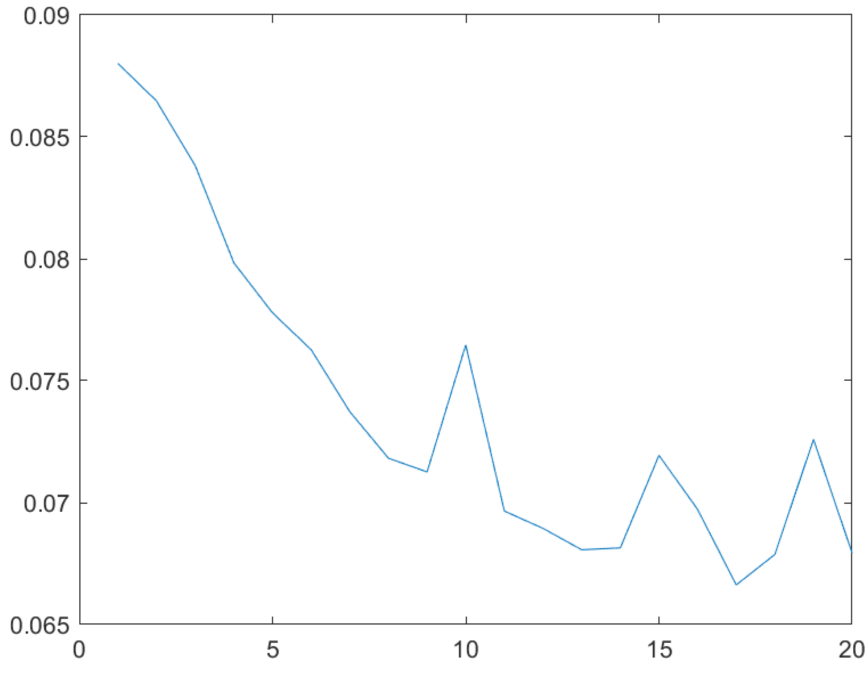

Figure 3: MSE vs SNR with 156.25 Hz frequency offset

The MSE of the filter bank given a certain SNR of the noise in the transmission channel model with a frequency offset of 156.25 Hz is given in Figure 3.

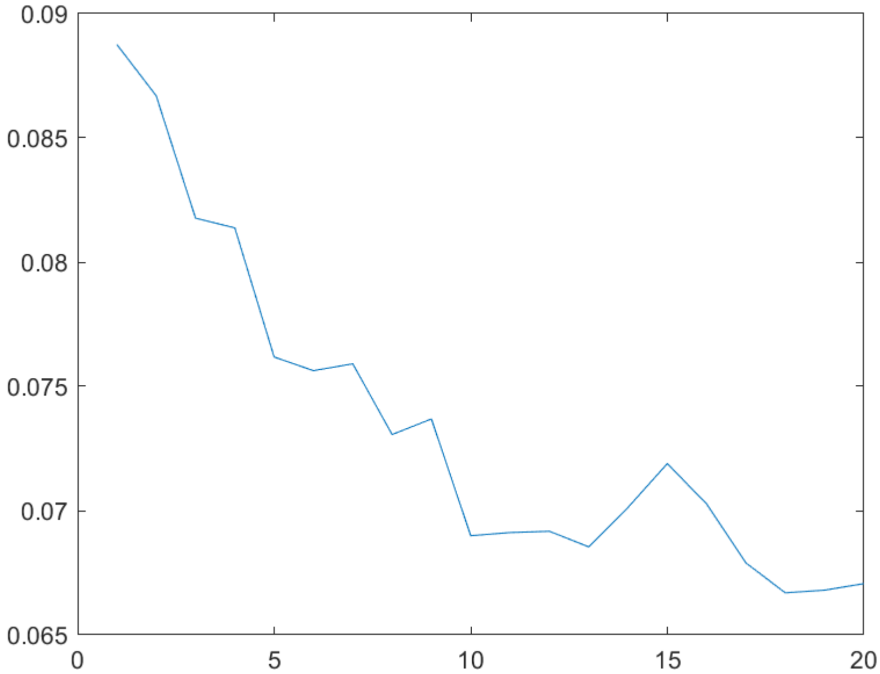

Figure 4: MSE vs SNR with -156.25 Hz frequency offset

The MSE of the filter bank given a certain SNR of the noise in the transmission channel model with a frequency offset of -156.25 Hz is given in Figure 4.

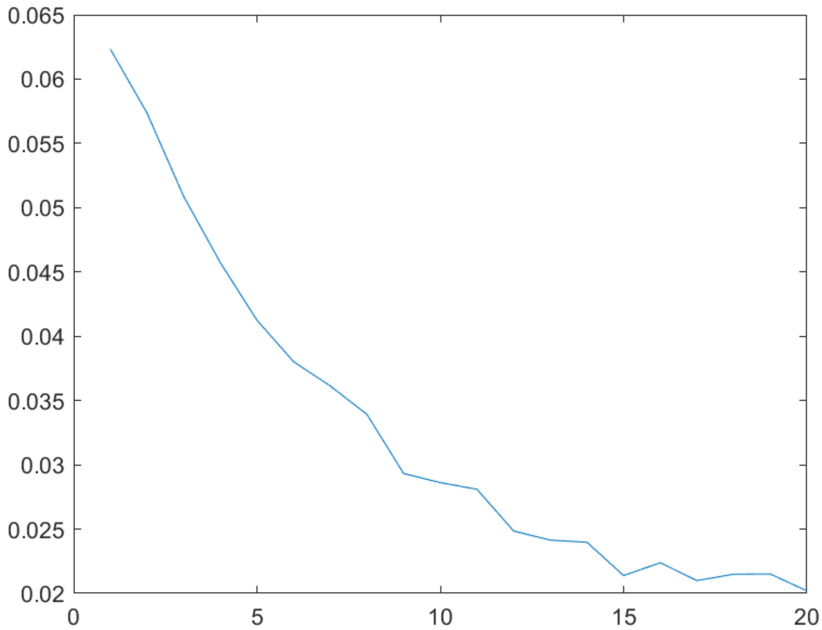

Figure 5: MSE vs SNR with 25 Hz frequency offset

The MSE of the filter bank given a certain SNR of the noise in the transmission channel model with a frequency offset of 25 Hz is given in Figure 5.

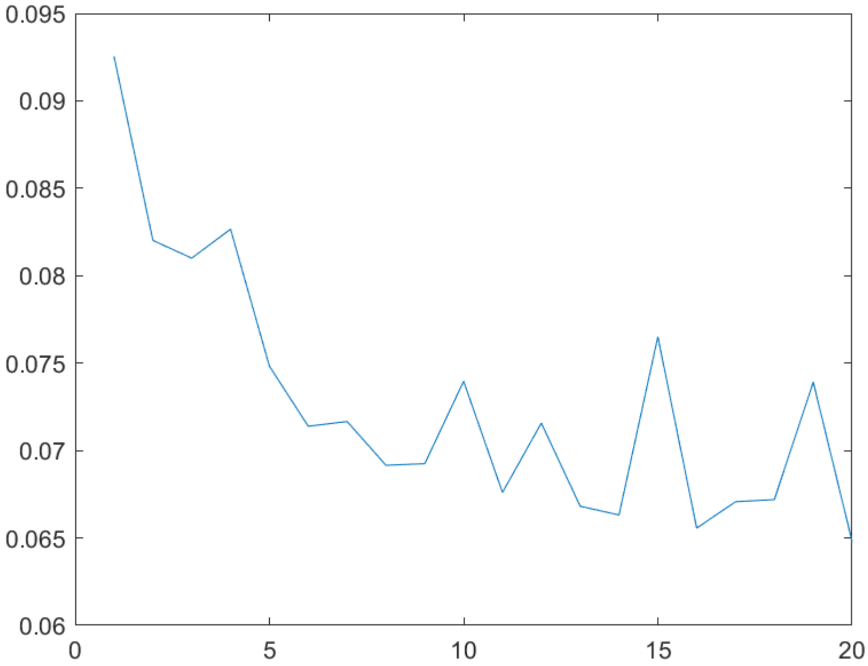

Figure 6: MSE vs SNR with 156.25 Hz offset and scrambling

The MSE vs SNR plot with 156.25 Hz offset and scrambling is given in Figure 6.

As expected, as the SNR is improved the MSE of the filter bank improves as well. The filters are affected by frequency offset quite a bit. This is especially noticeable as SNR improves, and frequency offset becomes the major contributor to error in the system. This shows that our system would be affected by doppler shift if it was implemented as an RF communication link. The scrambling process does not effect the MSE significantly.

Each plot is generated using the "part1_snr.m" file by varying the "scrambling" and, "frequency_offset" variables.

4 Kalman Filter

4.1 Implementation

The Kalman Filter is implemented in the file named "part4.m". The data was taken from Nasdaq's website for $AAPL for a year. No special libraries are needed.

4.2 Data

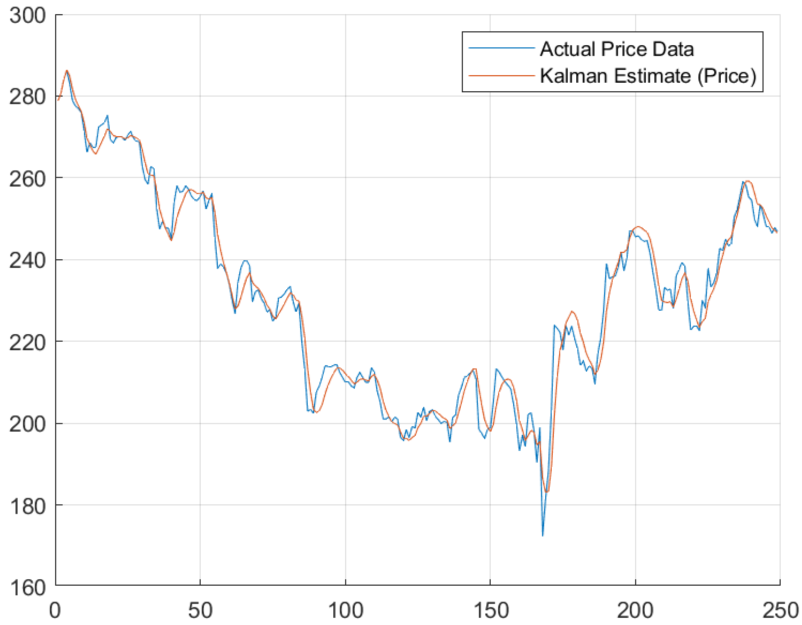

Figure 7: Kalman Filter Estimate vs Actual Data with Q=0.0001 and R=0.01

The estimate of the Kalman Filter and the actual stock price data is given in Figure 7. This data was captured with Q=0.0001 and R=0.01. MSE is 0.0012.

Figure 8: Kalman Filter Estimate vs Actual Data Q=0.0005 and R=0.01

The estimate of the Kalman Filter and the actual stock price data is given in Figure 8. This data was captured with Q=0.0005 and R=0.01. MSE equals 8.1502e-04.

Figure 9: Kalman Filter Estimate vs Actual Data Q=0.0001 and R=0.05

The estimate of the Kalman Filter and the actual stock price data is given in Figure 9. This data was captured with Q=0.0001 and R=0.05. MSE equals 0.0015.

Increasing Q makes the estimate follow the actual price closer with sharper edges. Increasing R has the opposite effect and smooths the estimate.