August Byrne

EECE 433

5/15/2021

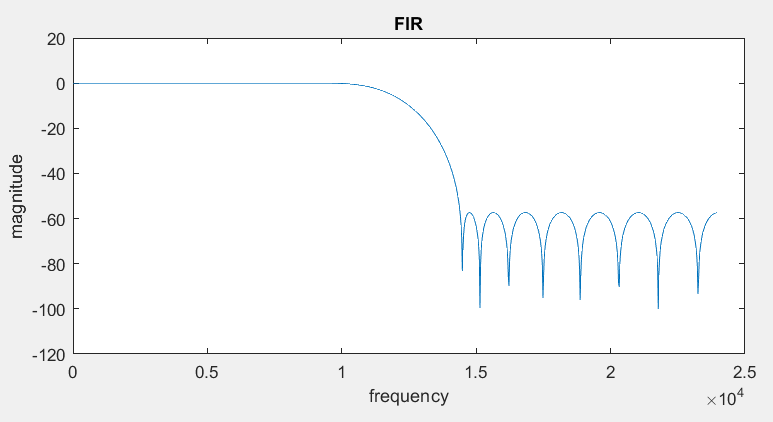
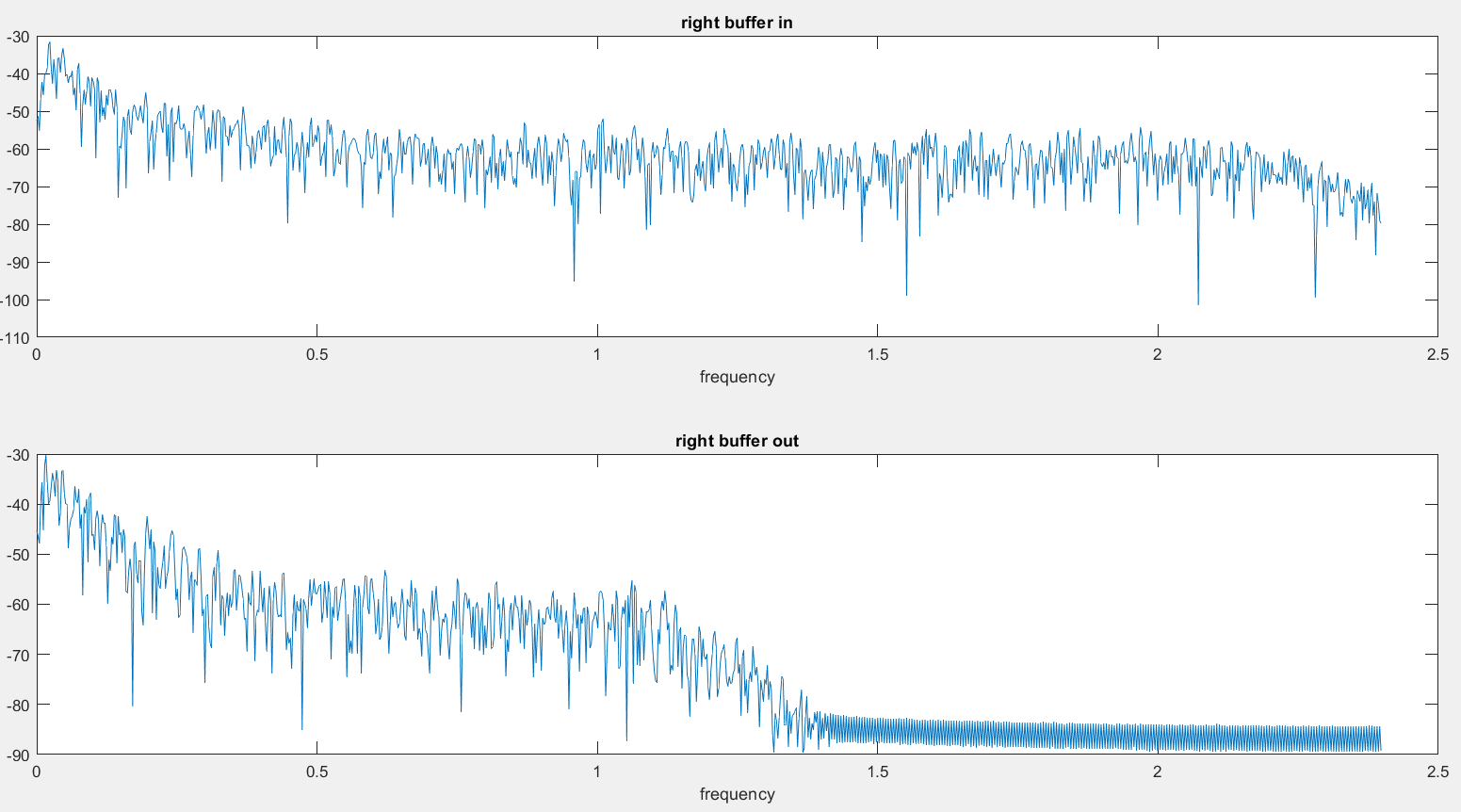
Lab 4

Part 1:

1. NUM\_TAPS
2. static float32\_t firCoeffF32[NUM\_TAPS] which is floating point
3. It creates an instance of the fir in q31 format
4. They relate to the left and right channel fir filters
5. That is the required length of the state buffer, as stated by the arm library.
6. arm\_fir\_init\_q31 has S which is an instance of the Q31 FIR filter structure, numTaps which is the number of taps, pCoeffs which is the filter coefficient buffer, pState which is state buffer, and blockSize is the number of samples processed.
7. arm\_fir\_q31 has S which is an instance of the Q31 FIR filter structure, numTaps which is the number of taps, pCoeffs which is the filter coefficient buffer, pState which is state buffer, and blockSize is the number of samples processed.
8. static q31\_t fir0Stateq31[NUM\_TAPS + AUDIO\_SAMPLES\_PER\_BLOCK - 1];

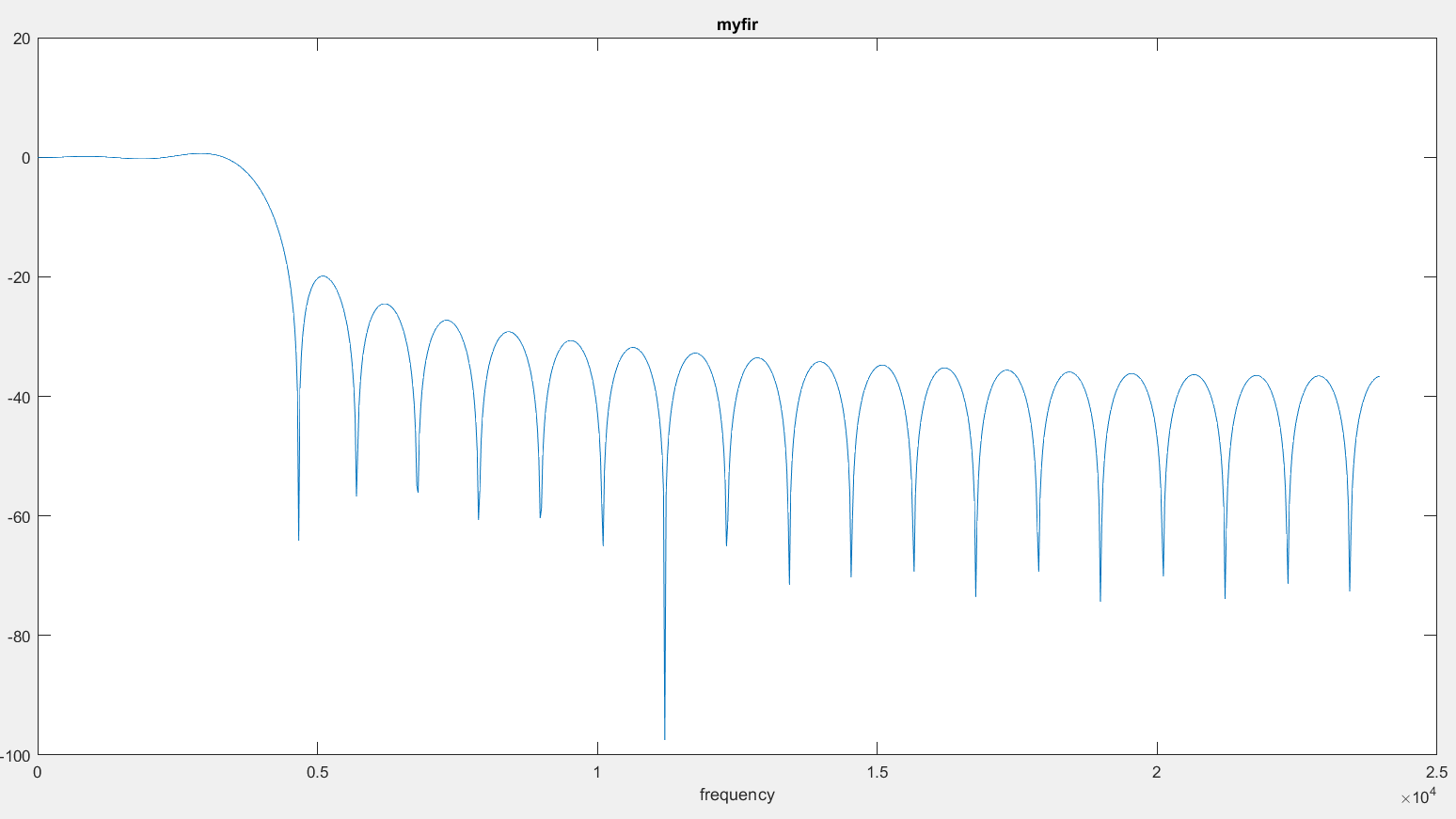
static q31\_t fir1Stateq31[NUM\_TAPS + AUDIO\_SAMPLES\_PER\_BLOCK - 1];

Part 2:

1. To find my filter coefficients, I used the firpm function in MATLAB. I also used Chassaing’s formulas in my c code to implement it inside my DSPInit function, but only used this in my part 2 Demos. The graphs here are all from firpm because it was better filtering.
2. This is the frequency response of the MATLAB generated filter I used.
3. This is the frequency response for white noise at buffer in, and after being filtered by firpm coefficients generated from MATLAB. I did this part using c code generated coefficients as well, but this was a clearer result, so I used the MATLAB generated coefficients here.

Part 3:

1. My AM demodulation algorithm works by multiplying the modulated signal by a sine wave that is at the same frequency of modulation. I then filter the demodulated signal to remove the unwanted high frequency aliased signals that come from the demodulation process.
2. For task 3, to find my filter coefficients I used Chassaing’s formulas in my c code to implement it inside my DSPInit function.
3. Here is the frequency response of my FIR filter



And below is the frequency response of my demodulated signal before, and after filtering it with my FIR filter. You can see that the demodulated frequencies are correct, since it was a 2kHz input signal that was modulated by a 20kz carrier frequency. It was then demodulated to give the frequency response of the upper graph below. We can see that the correct frequency has been demodulated back to 2kHz and that is also has the aliased frequencies that we would expect from this demodulation (frequencies of 38kHz and 42kHz aliased back down to 6kHz and 10kHz because of the sampling frequency of 48kHz). We can also see that the filter is working since we see on the lower graph that frequencies past the cutoff frequency of the filter (4kHz) are being attenuated.

