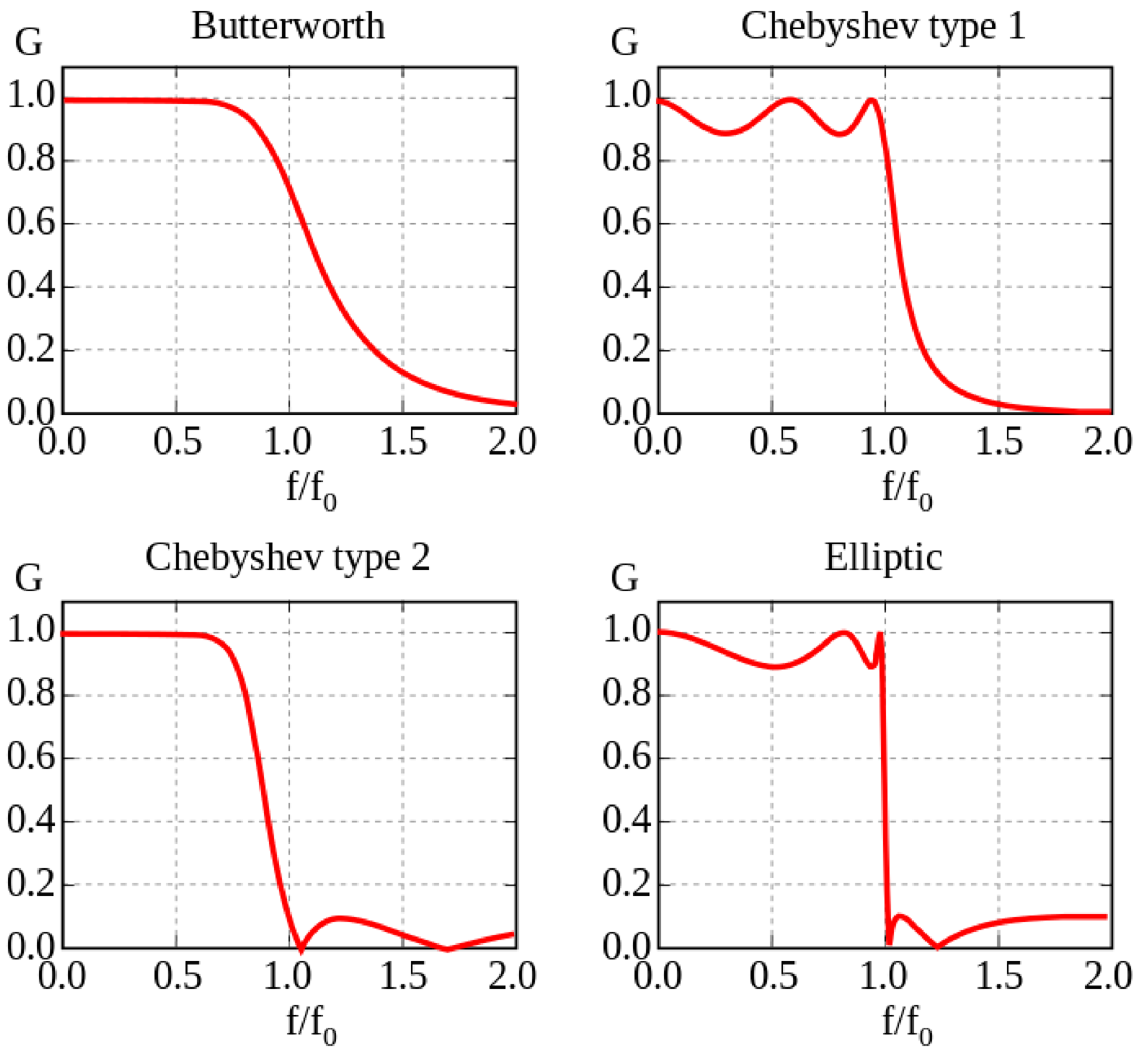
CHAIN: Week of June 3

June 3

- Goals:
  - Finish re-do of parameterizable filter
- What I learned:
  - GUI implementation
  - Filter types ad their use cases
    - Elliptic: fastest transition between passband and stopband
  - Filter parameters and how they affect the signal



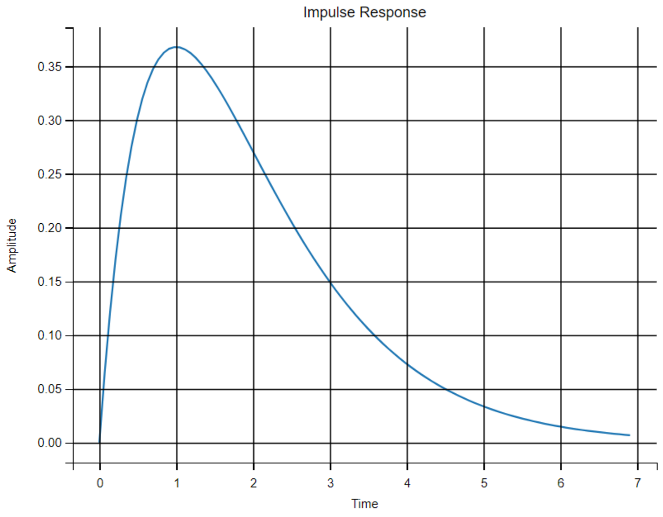
ParameterizableFilter.py (June 3)

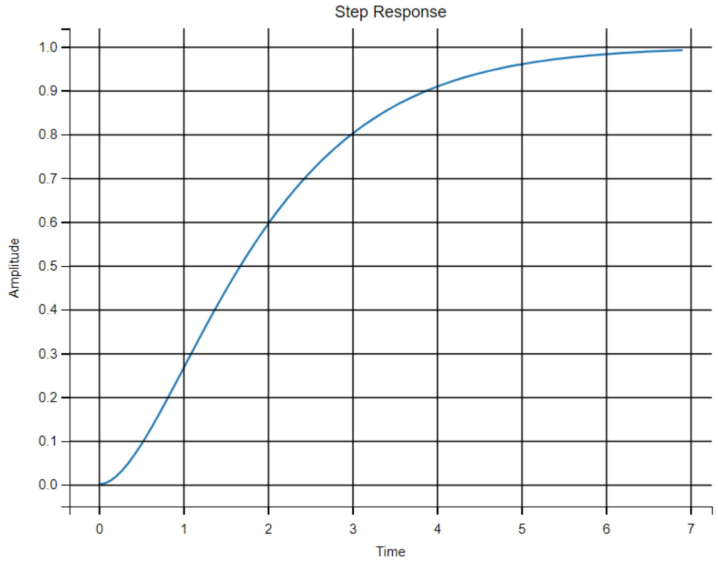
- Reads values from GUI, generates coefficients using elliptic(), generates transfer function using lfilter(), plots output
- Advantages:
    - Parameterizable
    - User-friendly GUI
    - Output in the complex domain
- Disadvantages:
    - Not in the frequency domain
    - Built-in functions have limited customization
    - Not efficient to implement on FPGAs
        - GUI
        - Floating-point arithmetic
    - Complex number generation
- Next steps:
    - Integer-based arithmetic
    - Frequency domain

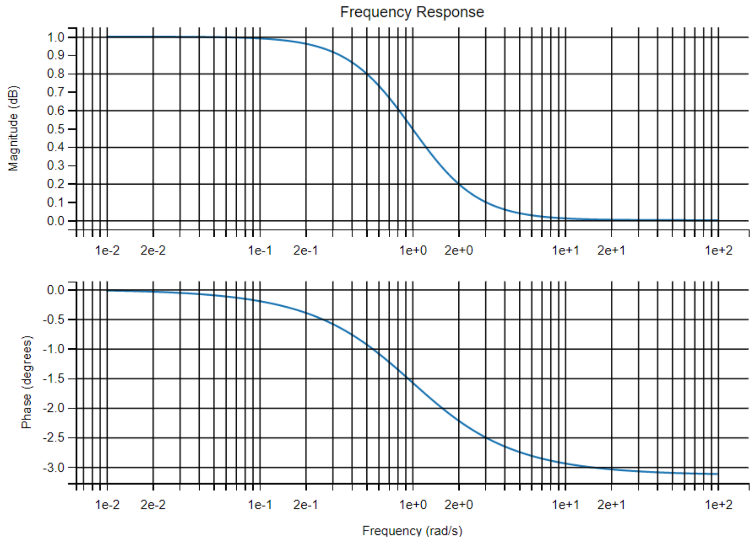
Basics

- Filter: device or process that removes some unwanted components or features from a signal
- Representation, Information:
    - Domains: time, and frequency
        - Frequency domain: information is contained in the relationship between points
            - Optimal characteristics: fast roll-off, flat passband, good stopband attenuation
                - Phase isn't important
        - Time domain:
            - Optimal characteristics: fast step response, no overshoot, linear phase
- Responses: impulse, step, and frequency
    - Impulse response: shows output of system being "fed" an impulse (short pulse)
    - Step response: shows how information represented in the time domain is being changed
    - Frequency response: shows how information represented in the frequency domain is being changed
    - Frequency response depends on coefficients
        - Coefficients determine type of filter (low pass, high pass, band pass, etc.)
        - Frequency response is found by integral of impulse response
- Convolve input with impulse response
    - Each output is calculated by being weighed and then added together
    - IIR filter's output, in contrast, is found through recursion
        - Chosen when linearity is not important and memory is limited

Responses


Impulse Response
Amplitude
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
1
2
3
4
5
6
7
Time

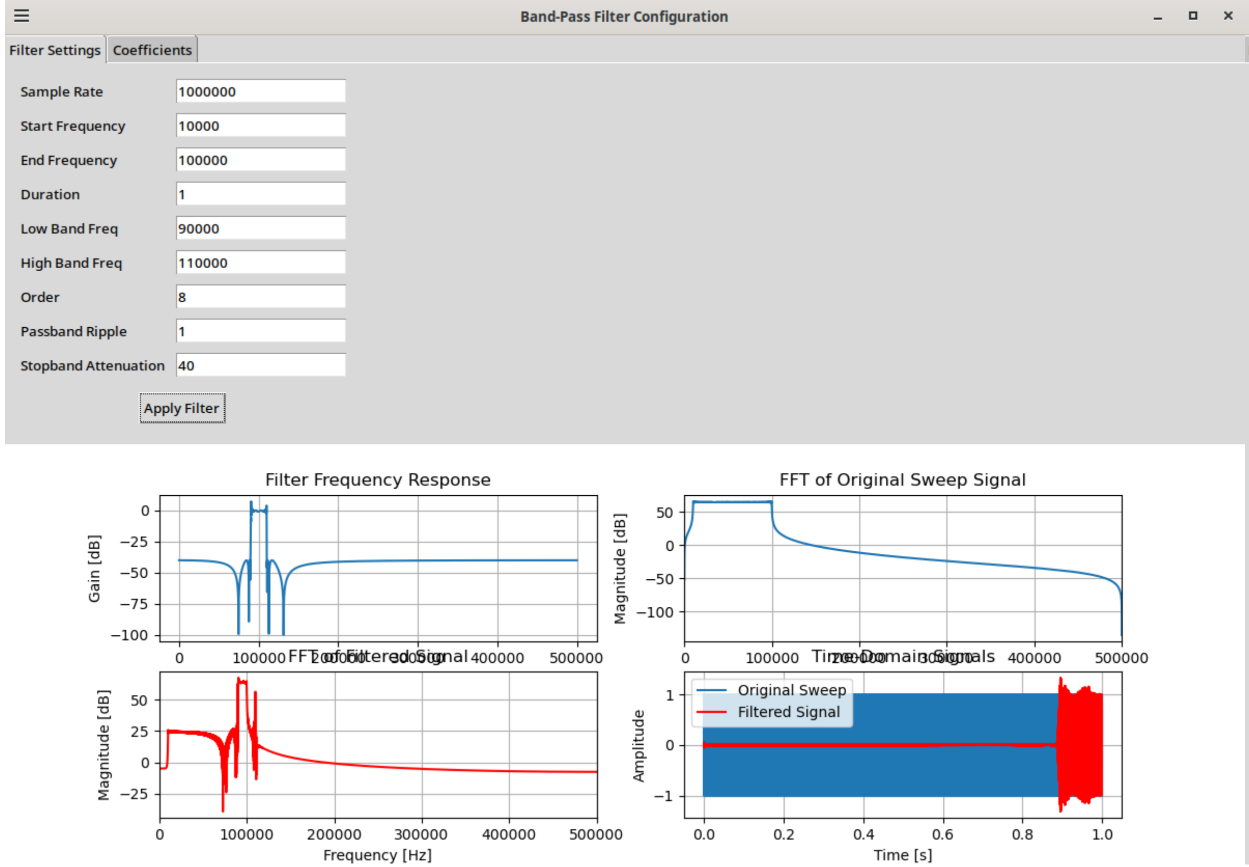

Step Response
Amplitude
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
6
7
Time


Frequency Response
Magnitude (dB)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1e-2
2e-2
1e-1
2e-1
1e+0
2e+0
1e+1
2e+1
1e+2
Phase (degrees)
0.0
-0.5
-1.0
-1.5
-2.0
-2.5
-3.0
1e-2
2e-2
1e-1
2e-1
1e+0
2e+0
1e+1
2e+1
1e+2
Frequency (rad/s)

June 4

- Goals:
  - Finish parameterizable filter:
    - Digitize
    - Transform into the frequency domain (FFT)
    - Do a frequency sweep
    - Test with various inputs
  - Gain knowledge about foundations of DDC, implementation in FPGA
    - Look at data sheets of various DDCs
- What I learned:
  - Performing frequency sweep
  - Components in DDC
    - Local oscillator, mixer, A/D converter, filter/decimator
    - Decimator considered part of filter
      - CIC decimation --> CIC gain correction --> CIC droop compensation decimator --> Halfband decimator --> Final decimator
      - Narrowband vs wideband
  - Integration with FPGA, parameters to consider, how they may affect FPGA functionality

BinaryParamFilter



Sample Rate 1000000

Start Frequency 10000

End Frequency 100000

Duration 1

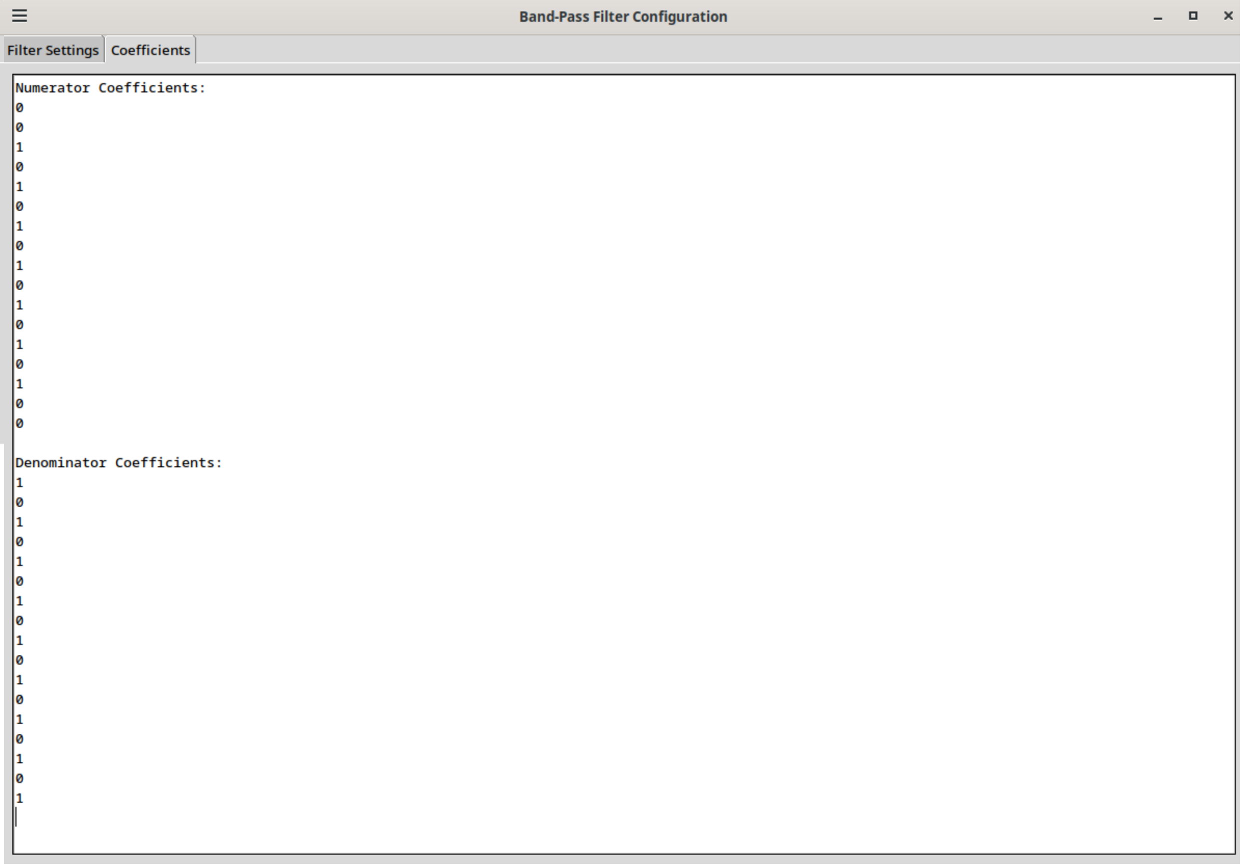
Low Band Freq 90000

High Band Freq 110000

Order 8

Passband Ripple 1

Stopband Attenuation 40



```
Numerator Coefficients:
0
0
1
0
1
0
1
0
1
0
1
0
1
0
1
0
0

Denominator Coefficients:
1
0
1
0
1
0
1
0
1
0
1
0
1
0
1
0
1
```

June 5

- Goals:
  - Implement a decimator
  - Implement a numerically-controlled oscillator (NCO)
- What I learned:

Research Article

- https://ieeexplore.ieee.org/document/7533688
- Receiver: receive the transmitted signal from multiple satellites and use these signals to decode navigation data and to measure the travelling time between the point of transmission (satellite) and the point of reception (receiver).
- The purpose of signal acquisition process is to obtain coarse Doppler frequency, code phase, and visible satellites.

Resources Consulted

- DDC FPGA: http://hunteng.co.uk/support/ddc.htm