FIR Octave Band Filtering

THE UNIVERSITY OF UTAH
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Miguel Gomez, Chase Griswold, Tyler Bytendorp, Benjamin Hayes

Digital Signal Processing - Dr. Neda Nategh

References:
- Proakis, J. G. (2007). Digital Signal Processing: Principles, Algorithms, and Applications, 4/E. Pearson Education.
- Yoder, J. H. M. R. W. S. M. A. (n.d.). 9. Z-Transforms. McClellan, Schafer, Yoder, DSP First, ISBN-10: 0136019250 • ISBN-13: 9780136019251. Prentice Hall, Upper Saddle River, NJ 07458. © 2016 Pearson Education, Inc. https://dspfirst.gatech.edu/chapters/07ztrans/overview.html
- J. J. Galvin and Q.-J. Fu, "Effect of bandpass filtering on melodic contour identification by cochlear implant users," The Journal of the Acoustical Society of America, vol. 129, no. 2, 2011. doi:10.1121/1.3531708

Introduction

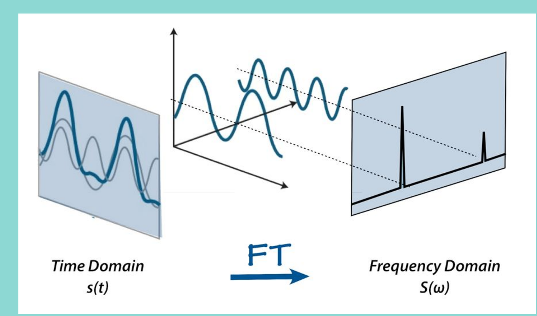
Why Finite Impulse Response (FIR) Filtering?

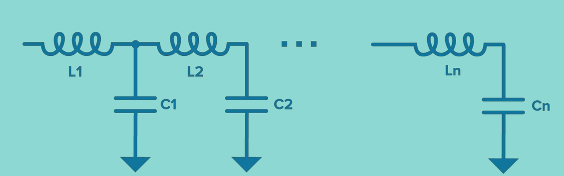
- Taking the Fourier Transform is a simple method of passing and rejecting select frequencies from a signal. However, this requires us to know the entire signal.
- Lumped Element, Waveguide, and Microstrip Filters (Analog) are one way to process an unknown signal as it is transmitted or received by a system.

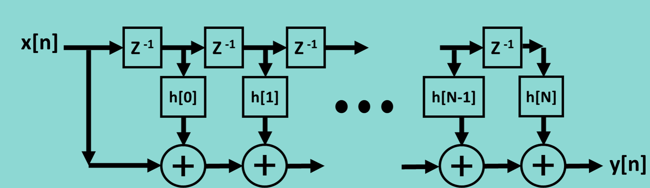
So how can we go about this digitally without knowing the entire signal beforehand? FIR.

- FIR Filtering works through Convolution in the time domain.
- Designed to process a signal in real-time with minimal delay.
- However, this means the FIR length is much shorter than the incoming signal, thus it is not ideal as this will create sidelobes where undesired frequencies still pass through the filter.

In this project, we walk through FIR octave band filter design.

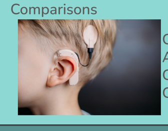






Audio Driver Seperation

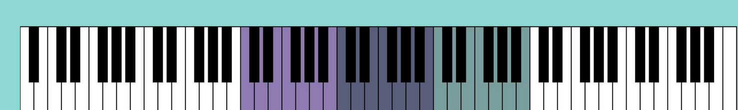
Cochlear Implants/ Audiology For Melodic Contouring Comparisons



Cochlear Implants/ Audiology For Melodic Contouring Comparisons

Based on western music theory, Octave Band Filtering is essentially an algorithm derived from the Piano.

- Piano octaves technically contain 12 notes total. (Based on 440 Hz A)
- Coined an octave because western scales contain 8 notes within the region between any two harmonics.
- i.e. We can define one bandpass FIR filter from C2 to B2 (65 Hz to 123 Hz).



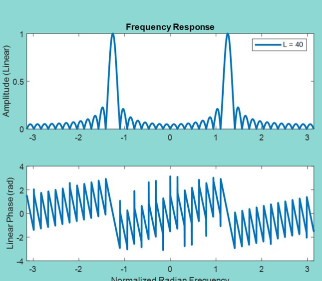
4.1: Simple Bandpass Filter Design

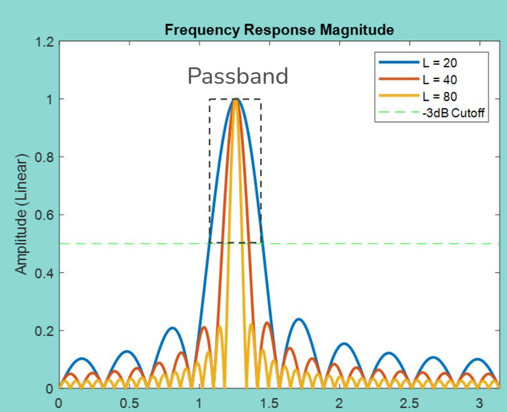
Impulse Response:

$$h[n] = \frac{2\cos(w_c \cdot n)}{L} \quad for\ n = 0, 1, 2, \ldots, L-1$$

where w_c = center freq,
n = index of the filter,
L = length of the filter

Our initial filter used $w_c = 0.4pi$





What are the properties we can identify about a BPF filter with a given impulse response and center frequency?

- Startband
- Stopband
- The passband region.

What point is the first > 0.5 (magnitude)?
What point is the last > 0.5 (magnitude)?
What is the length between these two points?

How do these values change as the length of the filter is increased?

- As the length increases, the passband width decreases.

Filter Length	L = 20	L = 40	L = 80
Startband	0.341pi	0.371pi	0.358pi
Stopband	0.458pi	0.430pi	0.417pi
Passband Width	0.120pi	0.059pi	0.030pi

Applying the Simple BPF to Piano Music

Convolution

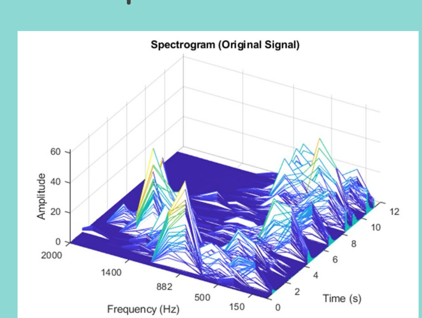
$$y[n] = \sum_{k=0}^{N-1} h[k] \cdot x[n-k]$$

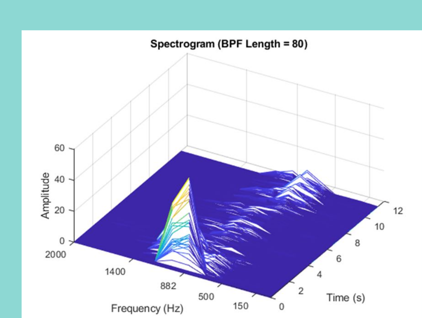
Showcasing the effect of filter length.

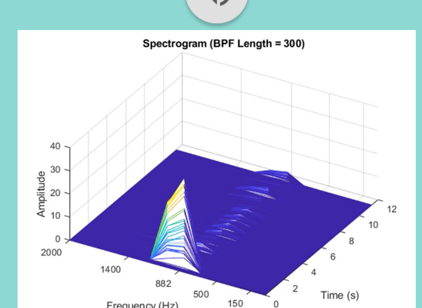
- We use a piano .mp3 recording and linearly convolve it with our given h[n] from part 4.1.
- Sampling rate of .mp3 is Fs = 44.1k, so an wc = 0.04pi results in a center frequency of 882 Hz.
- Increasing the length of the filter showcases the narrowing effect on the bandpass filter that increasing the length causes.

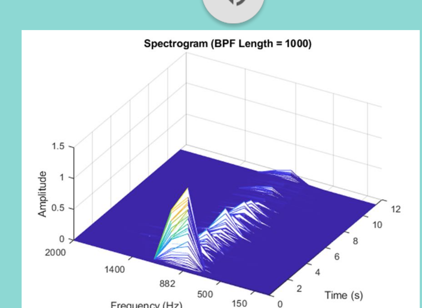
For filtering the piano music, we set
$w_c = 0.04pi$
If Fs = 44.1k for .mp3, then

$$f = \frac{w_c}{2pi} * (sampling\ rate) = \frac{0.04}{2} * 44.1k = 882\ Hz$$









Section 5: Octave Band Filtering

Our objective here is to design FIR filters to decode piano signals to identify the octaves in the signals. First, the octave bandpass frequencies are determined. These are used by an FIR filter design tool we developed to generate our octave band filter bank. The filter design tool generates better filters with flatter passbands. Then, both simulated and real audio signals are filtered and analyzed.

5.1: Determining Piano Octave Frequencies

Basics of piano notes and octaves:

- Piano has 88 notes; 7 full octaves, O1 to O7, +4 more notes.
- A4 is at 440 Hz and all frequencies are relative to A4.
- Notes are geometrically separated; Each note is separated by a factor of $2^{1/12}$ away.
  - Ex) fa#4 = fa4 · $2^{1/12}$ and Ex) fa5 = fa4 · 2
- Lab has sampling frequency, fs = 8000 Hz. Arbitrary value.

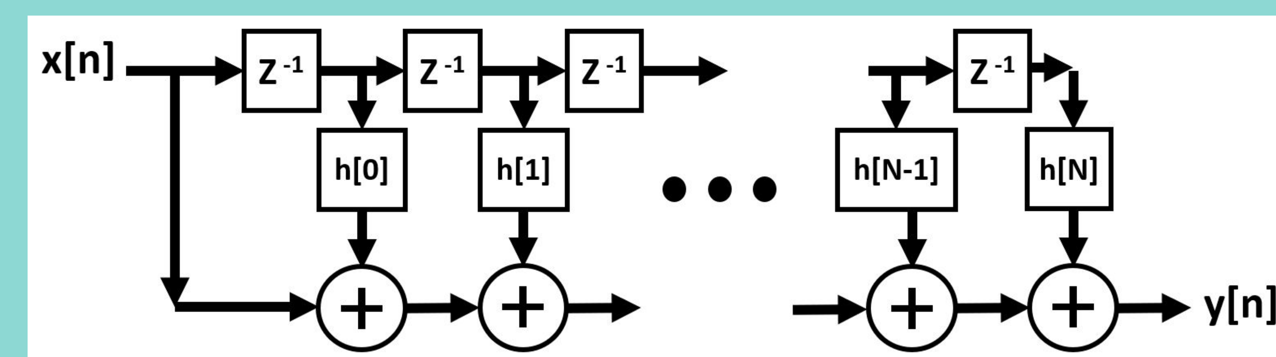
Must calculate upper and lower frequencies for cutoffs for design.

- Here: the bandwidth is defined from Cn to Bn.
- Here: the arithmetic mean is center.
- Octave 8 is beyond half the sampling rate.
- Implementation: Geometric mean of Cn to Cn+1, is F#.
- Implementation: Cutoffs geometrically spaced from F#.

val [units]	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
Lower (Hz)	32.703	65.406	130.81	261.63	523.25	1046.5	2093	4186
Lower (Rad)	0.025685	0.05137	0.10274	0.20548	0.41096	0.82192	1.6438	3.2877
Upper (Hz)	61.735	123.47	246.94	493.88	987.77	1975.5	3951.1	7902.1
Upper (Rad)	0.048487	0.096974	0.19395	0.3879	0.77579	1.5516	3.1032	6.2063
Center (Hz)	47.219	94.439	188.88	377.75	755.51	1511	3022	6044.1
Center (Rad)	0.037086	0.074172	0.14834	0.29669	0.59338	1.1868	2.3735	4.747

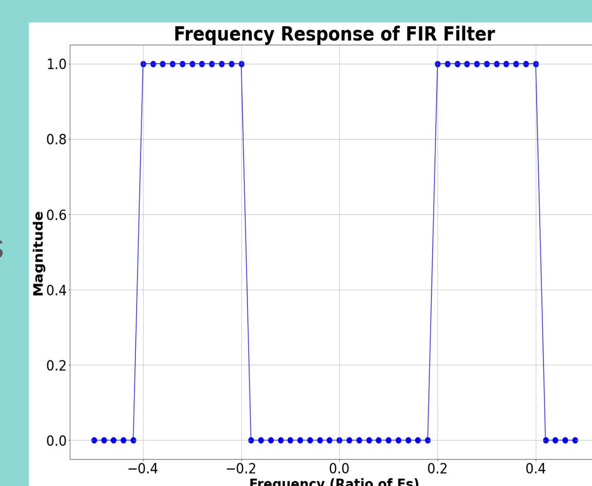
Table 1: Frequency Ranges for Octaves 1 to 8 starting with O_1

5.2: Creating FIR Filters from a Specified Pass-Band



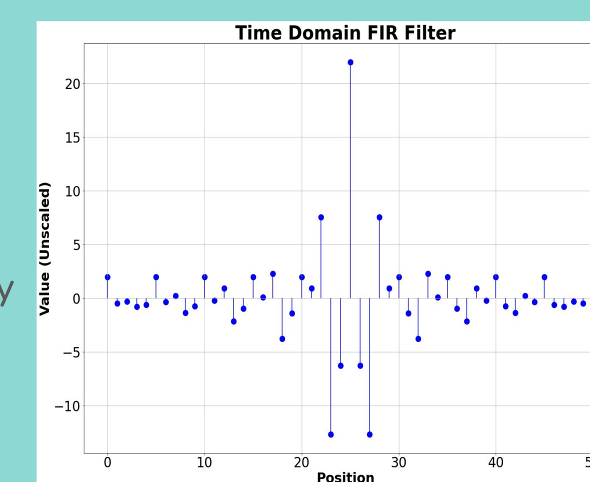
Step 1:

- Specify the ideal frequency response from 0 to Fs/2
- Use L points where L is the length of the FIR filter
- For f<0, use symmetry to avoid complex values



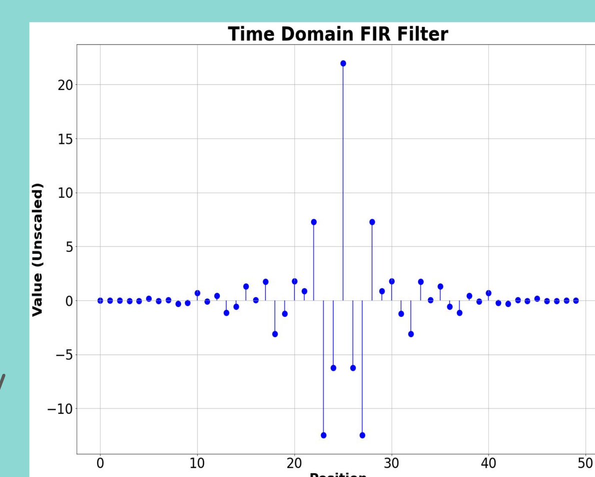
Step 2:

- Take the FFT of the ideal frequency response
- If done correctly, there should be no imaginary component



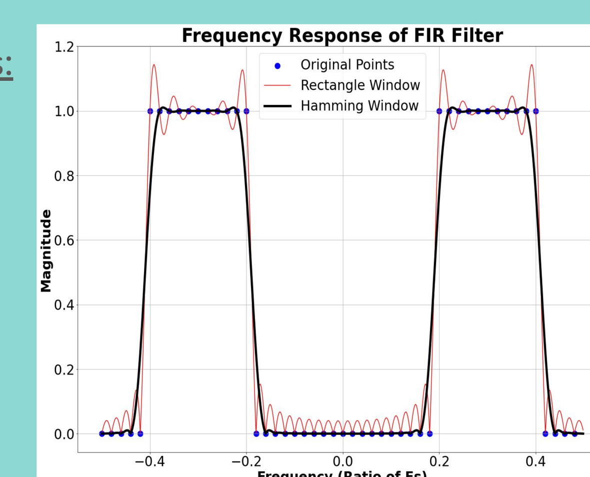
Step 3:

- Apply a windowing function through multiplication in the time domain
- This example uses a Hamming window
- The Hamming window flattens the ends of the FIR filter



Confirm the FIR Filter Works:

- Append many zeros to the FIR filter and take the FFT
- Few zeros are required for convergence.



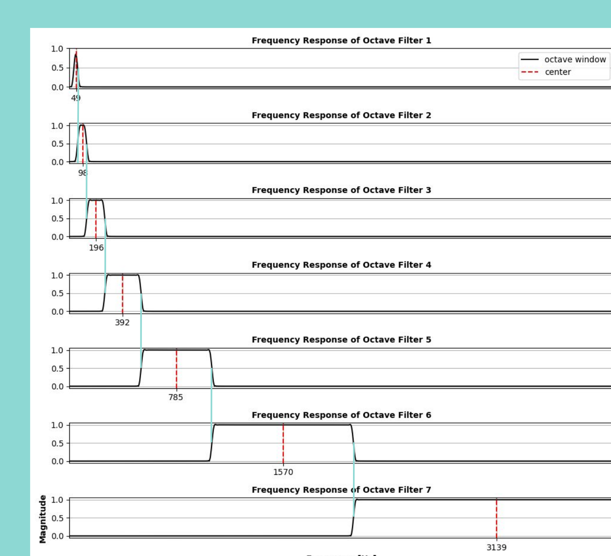
5.3 Octave Decoding

$$x(t) = \begin{cases} \cos(2\pi \cdot 220t) & 0.0 < t \le 0.25, \\ \cos(2\pi \cdot 880t) & 0.3 < t \le 0.55, \\ \cos(2\pi \cdot 440t) + \cos(2\pi \cdot 1760t) & 0.60 < t \le 0.85 \\ 0 & \text{otherwise} \end{cases}$$

Test Signal:
The signal used to test out the filters is a few sinusoids with different frequencies representing the A notes in their octaves, and a superposition of two sinusoids with A's, two octaves apart.

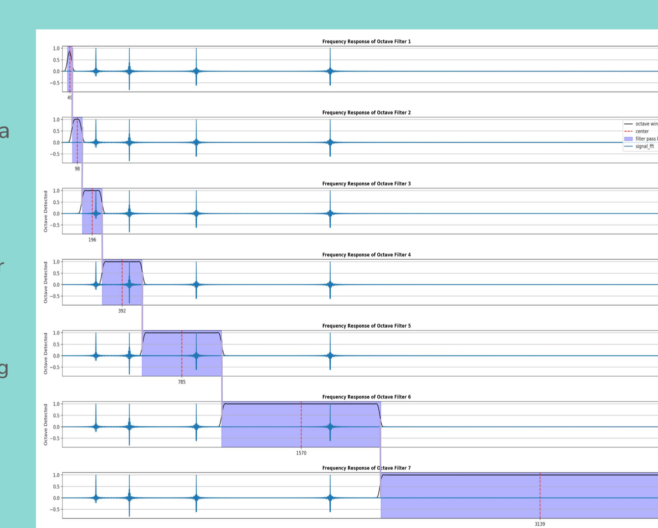
Creating the filter bank:

- Converted the above filter spec into a python class for modularity, increasing development speed.
- Modularity made including the entire range of the piano trivial. Given the sampling rate restriction, we show through octave seven.
- Using a higher order gives sharper transition bands without affecting the bandwidth in our design, modularity helped get sharper results.
- O_1 - O_7 shown on right with centers and lines showing the transitions between octaves.
- Flat pass-band provides unity output throughout the entire filter as expected.



Filtering Signals:
Using the filter bank, we can now pass in our sinusoidal data with multiple frequencies and detect the presence of some note in an octave.

- Overlay shows the filter passband, the rectangular passband region, and labels on detected notes.
- Sharper cutoffs are great for future work attempting the same with all 88 keys on a piano.

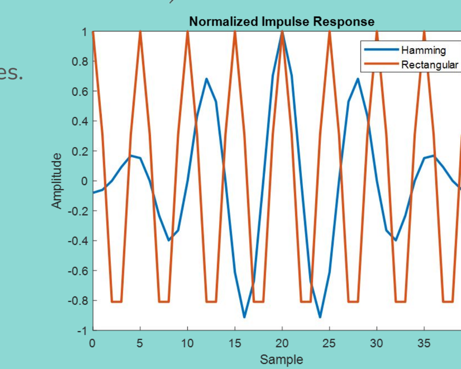


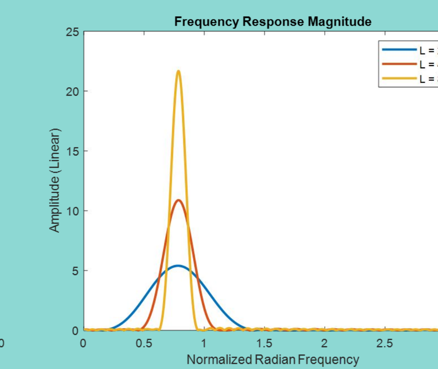
4.2: Applying a Hamming Window to a Cosine Filter

$$h[n] = \left(0.54 - .46\cos\left(\frac{2\pi n}{L-1}\right)\right)\left(\cos\left(n - \frac{L-1}{2}\right)\right) \quad \text{for } n = 0\ 1\ 2\ \ldots\ L-1$$

FIR filter windowed by rect().
Problem: Excessive stopband ripples.
Solution: Replace with Hamming window - a better BFP.

- wc = 0.25·pi
- Hamming window reduces coefficients near edges
- Smaller stopband ripples!
- -3dB cutoff for width
- Width approx. inversely proportional to filter length

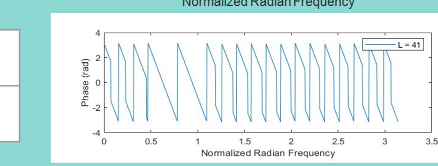




Filtering test signal:

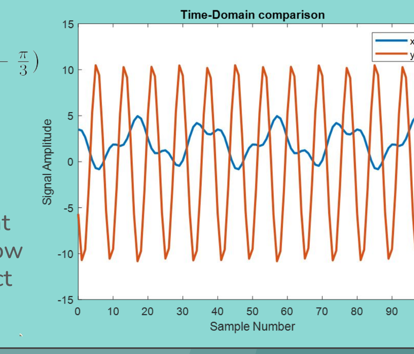
- By hand: For each frequency component, multiply amplitude and add phase of signal and filter.
- Convolution in time domain.

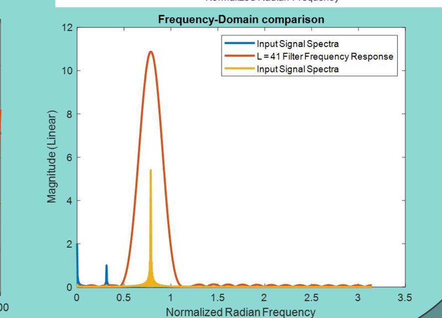
Filter Length	L = 21	L = 41	L = 81
Passband Width	0.180pi	0.088pi	0.044pi



$$x[n] = 2 + 2\cos(.1\pi n + \tfrac{\pi}{3}) + \cos(.25\pi n - \tfrac{\pi}{3})$$

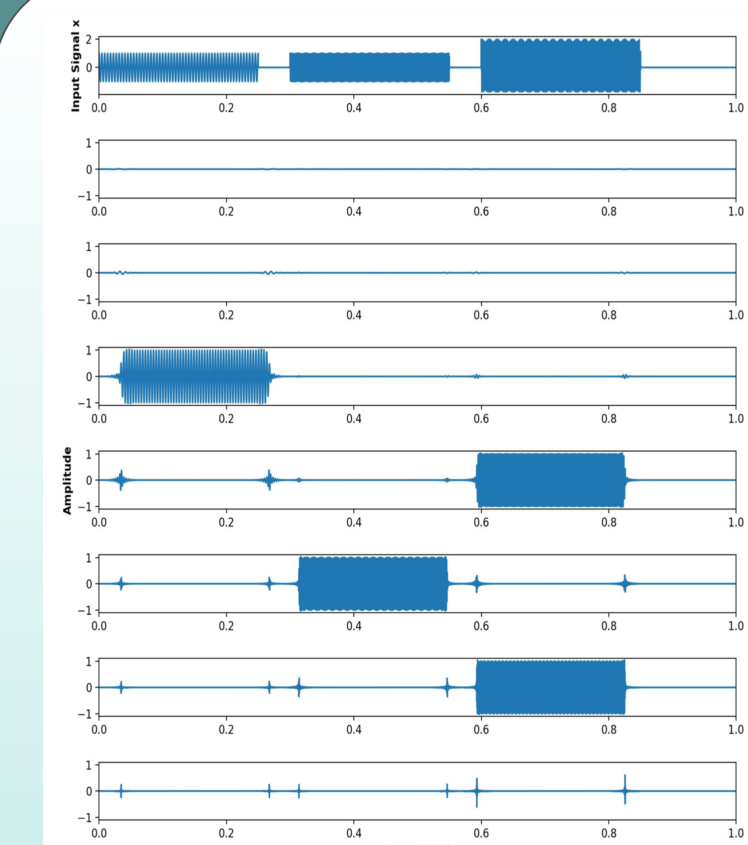
	0	0.1pi	0.25pi
Mag	0.00	0.00	10.88
Phase	3.1416	-3.1416	3.1416





- Input signal 0.25·pi component amplified in passband, while low stopband ripples strongly reject DC and 0.1·pi.

5.3 Results/Conclusions



Given a particular signal, we would expect to have the sampling frequency ahead of time to create a system like this; to correctly pass only the sinusoidal signals with the corresponding frequency. A clear indication of this working as intended is the separation of two octaves between the 440Hz and the 1760Hz sinusoids in the third time window. Additionally, the filters correctly pass the signal without amplification, proving the flatband works as intended and minimizes ripple effects.

Transient effects can be seen in the time data. The sharp transition in the signal from a sinusoid to 0 in the spaces between them creates discontinuities. As we have learned, these impulses contain an infinite number of frequencies, and thus, all filters pick these up to some extent.

There is a specific time needed in order for the filter to begin capturing a tone. Similar to a rise time in a transistor gate, we can measure the time taken to fully rise, and fall. ~.035[s] or 35[ms] rise and fall time. The time is approximately the same for all filters, but is quicker to rise with higher frequencies as it takes less time for it to accumulate data as the signal propagates through it.

FIR Filter Pros and Cons

Pros:

- Real-time processing
- Rapidly reprogrammable

Cons:

- Sidelobes
- Transition bands

Conclusion:

- An FIR filter can successfully isolate desired octaves
- Windowing functions like the Hamming can reduce ripple at the cost of transition band steepness
- Increasing the length of an FIR filter improves the accuracy at the obvious trade off of more hardware and a longer delay

Extras: