6.341 RTL-SDR Project Report

Franklin Zhang

November 2020

1 FM Modulation

For our understanding, we will consider a simple example of a FM signal. A transmitted FM signal $y_t(t)$ contains a message, $x_m(t)$ and a carrier, $x_c(t) = A_c \cos(2\pi f_c t)$ where f_c is the carrier frequency, A_c is the carrier amplitude and A_m is the message amplitude. The signal $y_t(t)$ is obtained by frequency modulation of the carrier and the message. The transmitted signal $y_t(t)$ can be written as follows.

$$y_t(t) = A_c \cos(2\pi \int_0^t \mathbf{f}(\tau) d\tau)$$

$$= A_c \cos(2\pi \int_0^t [f_c + f_\Delta x_m(\tau) d\tau])$$

As shown in the above equation, the message is encoded in the frequency changes. $\mathbf{f}(\tau)$ is the instantaneous frequency of the oscillator

We can approximate our message as a CT sinusoid with frequency f_m. The integral of $x_m(t)$ is:

$$\int_0^t x_m(\tau) d\tau = \frac{\sin(2\pi f_m t)}{2\pi f_m} = \frac{\sin(\Omega_m t)}{\Omega_m}$$

and $y_t(t)$ can be written as:

$$y_t(t) = A_c \cos(2\pi f_c t + \frac{f_\Delta}{f_m} \sin(2\pi f_m t))$$

$$= A_c \cos(\Omega_c t + \Omega_\Delta \frac{\sin \Omega_m t}{\Omega_m}) = A_c \cos(\Omega_c t + \Omega_\Delta \int_0^t x_m(\tau) d\tau) = A_c \cos(\Omega_c t + \phi(t))$$

The phase term $\phi(t)$ contains the message data. More specifically, its the scaled integral of the message. $y_t(t)$ is created by multiplying the carrier by an exponential term in the time domain, which is convolution in the frequency domain. The signal's DTFT can be thought of as the convolution of 2 deltas at the carrier frequency with the transform of the exponential. Using this model for the signal and message, we can move forward with demodulation.

2 Dataset

The data set used was (data1.raw) which has a hardware frequency of 100.3 MHz with the station of interest centered at 100.7 MHz and a sampling rate of 2.048 MHz.

3 Discrete Time Channel Selection

3.1 DT Modulation to Baseband

The transmitted signal that was received has a frequency of 100.7 MHz. During data collection, the RTL-SDR modulates the signal to baseband and passes a low passed version through an ADC. Since the hardware modulation frequency was chosen to be 100.3 MHz rather than 100.7 MHz, the DT signal has to go through another round of modulation in order to center the desired signal at $\omega = 0$.

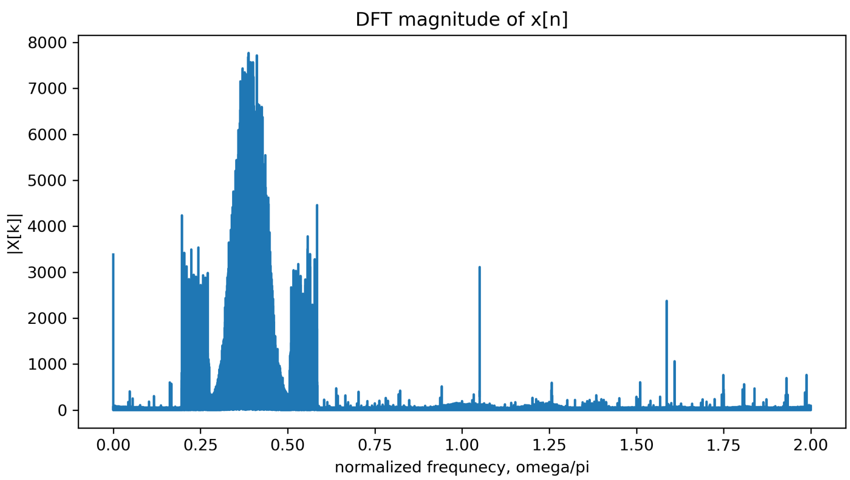

Figure 1: DFT of x[n]

Figure 1 shows that the desired frequency isn't centered at $\omega = 0$. The deviation from zero can be calculated as follows.

$$f_{dev} = (100.7 - 100.3) \text{ MHz} = 400,000 \text{ Hz}$$

$$\omega_{dev} = 2\pi f_{dev}/f_s = 0.390625\pi$$

We can visually see in 1 that 0.39 π is approximately the location of where the peak signal content is located. We can select the appropriate DT channel using the block diagram shown in figure 2.

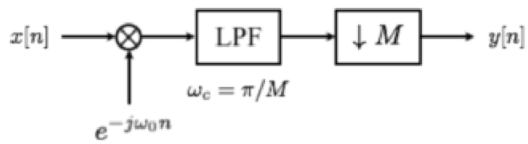

Figure 2: DT Channel Selection

We can modulate $x[n]$ with a complex exponential, $e^{-j\omega_{dev} n}$ to create $x_{bb}[n]$. $x_{bb}[n]$ would be centered at $\omega = 0$. The results of modulation is shown in figure 3.

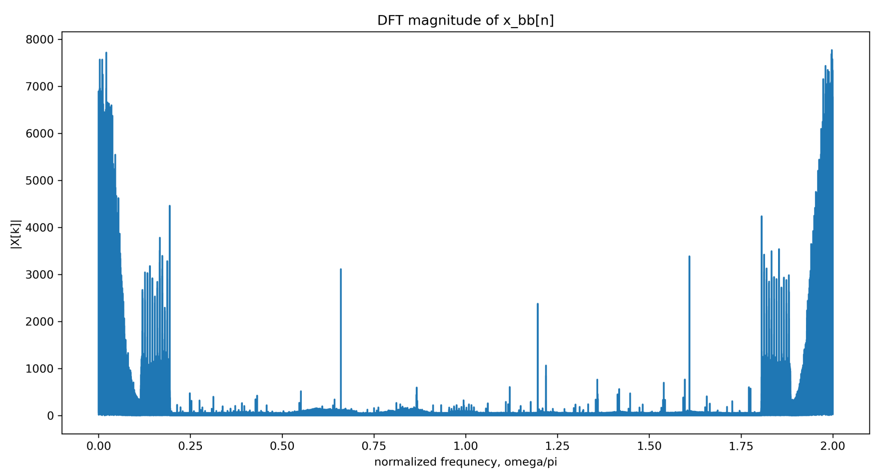

Figure 3: DFT of $x_{bb}[n]$

Now that the desired signal is centered at $\omega = 0$, we can now downsample to 256 kHz. This requires a LPF and a decimator.

3.2 LPF Design

The low pass filter is required to prevent aliasing during the decimation. The LPF is designed given a decimation factor of M=8, which takes the sampling rate of 2.048 MHz down to 256 kHz. Below lists the constraints and explanations for the choices of the LPF parameters.

- Window Type: Parks McCellan
- Cutoff Frequency: $\omega_c = 0.11125\ \pi$
  The cutoff frequency must be less than $\pi/M = 0.125\pi$ to prevent aliasing due to decimation. The signal of interest has is bandlimited to 200 KHz which is equal to 0.195π Rads/sample. To make sure that we don't alias

and to pass the entire frequency bandwidth of the signal through the LPF, we have to choose a cutoff frequency between 0.0975 π and 0.125 π. ω_c was chosen to be the center of that range, with a value of 0.11125π

- Transition width: $\Delta\omega = 0.01375\pi$ This was chosen to be half of the difference between the upper and lower bounds of the cutoff frequency. The transition region chosen to be small enough so that if doesn't effect the signal and doesn't cause aliasing

- Stop band ripple: $\delta_{s_{(dB)}} = -80$ dB The largest DFT magnitude outside the passband range is around $|X_{bb}[k]| = 4500$. We want to make this value negligible compared to the signal. If we scale it linearly by a factor of 10^{-4}, it will be much smaller than the DFT magnitude of the signal, which falls in the range of 50 - 8000. We can use the below equation to get the ripple in dB. $\delta_{s_{(dB)}} = 20log_{10}(\delta_{s_{(linear)}})$

- Pass band ripple: $\delta_{p_{dB}} = -1$ dB We want a LPF with gain equal to 1. Choosing a ripple of -1 dB allows the passband to be scaled by nearly 1 and keep the relative magnitude of the passband much larger compared to the stopband.

The filter that was designed has a impulse response given in figure 5 and magnitude response shown in figure 4. The DFT of the low pass signal $x_{lp}[n]$ is shown in 6.

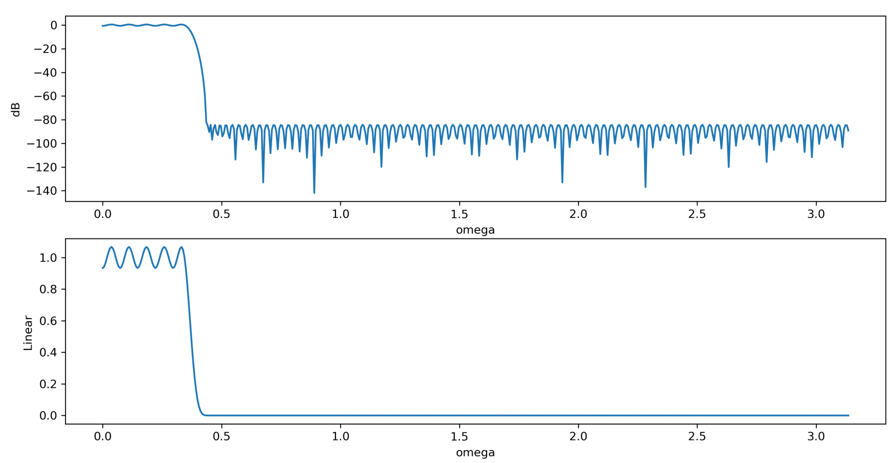

Figure 4: LPF designed using the Parks McClellan method.

3.3 Decimation

We need to next decimate the signal to get a signal $y[n]$ where $y[n] = x_{lp}[nM]$. The DFT of $y[n]$ is the the DFT of $x_{lp}[n]$ with periodic replicas and a scaled

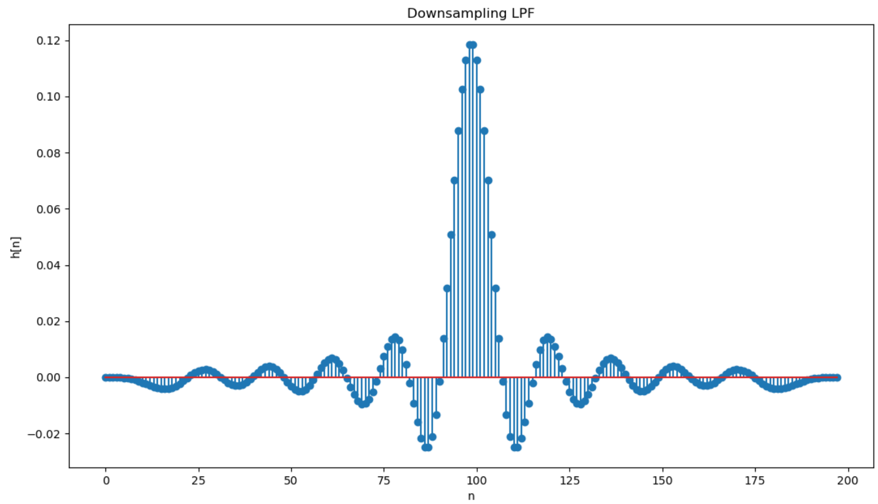

Figure 5: Impulse response of FIR LPF designed using the Parks McClellan Method

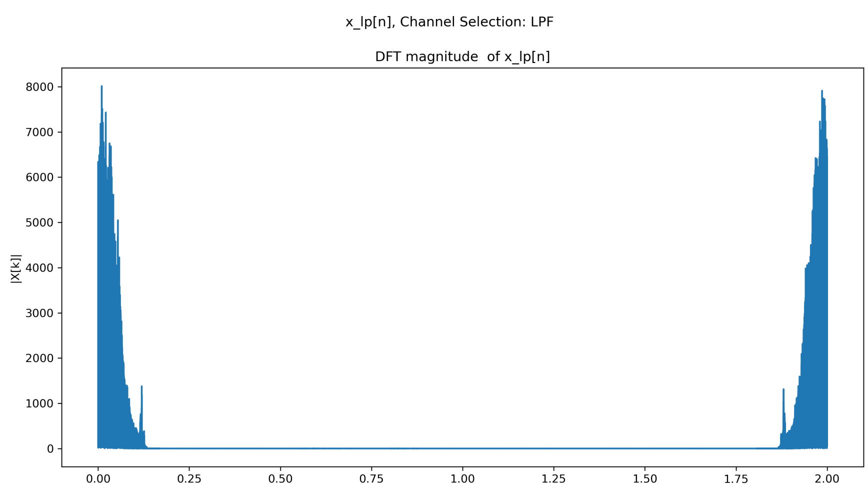

Figure 6: The DFT of $x_{lp}[n]$, where $x_{lp}[n]$ is the result of low pass filtering the $x_{bb}[n]$

frequency axis. The DFT of $y[n]$ is shown in figure 7 With the signal at 256 kHz sampling and the channel appropriately centered, we can move on to the frequency discriminator step.

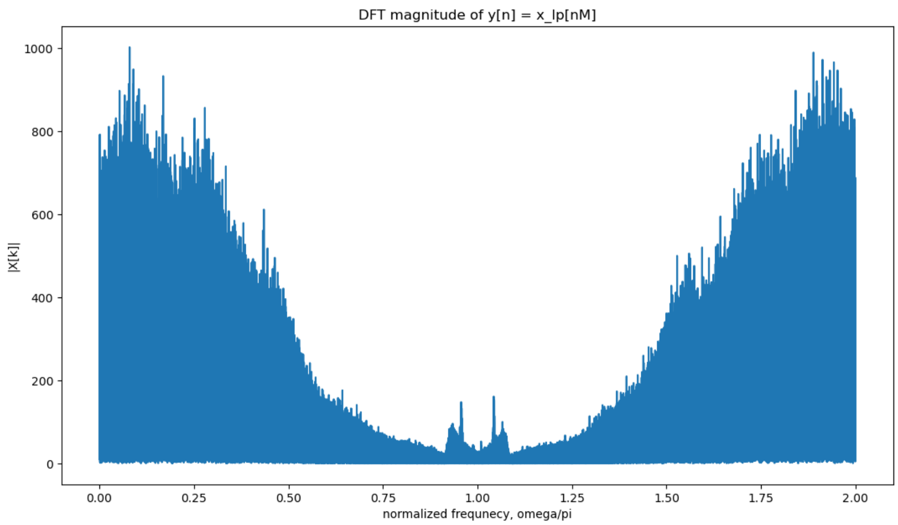

Figure 7: The DFT of $y[n]$ is the the scaled DFT of $x_{lp}[n]$ with periodic replicas and frequency scaling

4 Frequency Discriminator

The frequency discriminator extracts the message signal from the modulated signal. The message is in the integral term part of the phase of the signal. For reference, the original signal is:

$$y_t(t) = A_c \cos(\Omega_c t + \Omega_\Delta \int_0^t x_m(\tau) d\tau)$$

After sampling and channel selection, $y_t(t)$ is converted to discrete time, turned into a complex number and the frequency is centered at $\omega = 0$. So we can write the input to the frequency discriminator as the sampled version of some CT signal:

$$y_c(t) = Ae^{j\Delta \int_0^t x_m(\tau)d\tau}$$

where A and Δ are some constants.

The block diagram of the DT frequency discriminator is shown in figure 8. To understand how the frequency discriminator works, we can more easily think about the process with a CT signal. Essentially the frequency discriminator

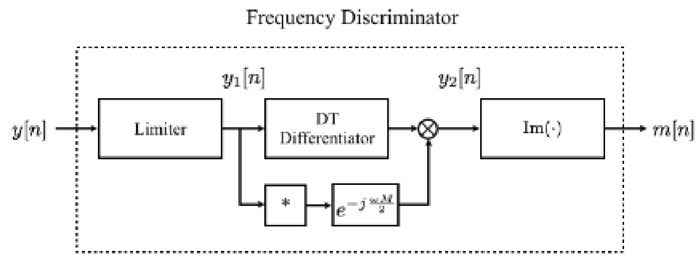

Figure 8: Block diagram of the frequency discriminator

in CT is a limiter followed by taking the derivative, followed by taking the imaginary part.

Given the CT input signal $y_c(t)$, the limiter gets rid of the A term in front of the exponential, giving us $y_1(t)$. Then taking the derivative pulls the message $x_m(t)$ out of the exponent and multiplying that by the conjugate of $y_1(t)$ will get rid of the exponential term, and that result we will be calling $y_2(t)$. We can then take the imaginary part of fully extract the message from the signal, and that result we are calling $m(t)$. The steps are shown below.

$$
\begin{aligned}
y_1(t) &= \frac{1}{A_c} y_c(t) = e^{j\Omega_\Delta \int_0^t x_m(\tau)d\tau} \\
\frac{d}{dt} y_1(t) &= j\Omega_\Delta x_m(t) y_1(t) = j\Omega_\Delta x_m(t) e^{j\Omega_\Delta \int_0^t x_m(\tau)d\tau} \\
y_2(t) &= y_1^*(t) \frac{d}{dt} y_1(t) \\
&= j\Omega_\Delta x_m(t) y_1(t) y_1^*(t) \\
&= j\Omega_\Delta x_m(t) \\
m(t) &= \Omega_\Delta x_m(t)
\end{aligned}
$$

This is the intuition behind the frequency discriminator, however we will be implementing this in discrete time, which will have some differences in the exact implementation. The differentiater has to be a FIR filter, and will introduce a delay. The same delay has to be considered before multiplying the conjugate with the output of the DT differentiator.

4.1 Limiter

The limiter scales the signal $y[n]$ so that it has unit magnitude. $y_1[n] = \frac{y_1[n]}{|y[n]|}$ The DFT of $y_1[n]$ is shown in 9. The DFT is scaled almost uniformly, but some of the higher frequencies are scaled more.

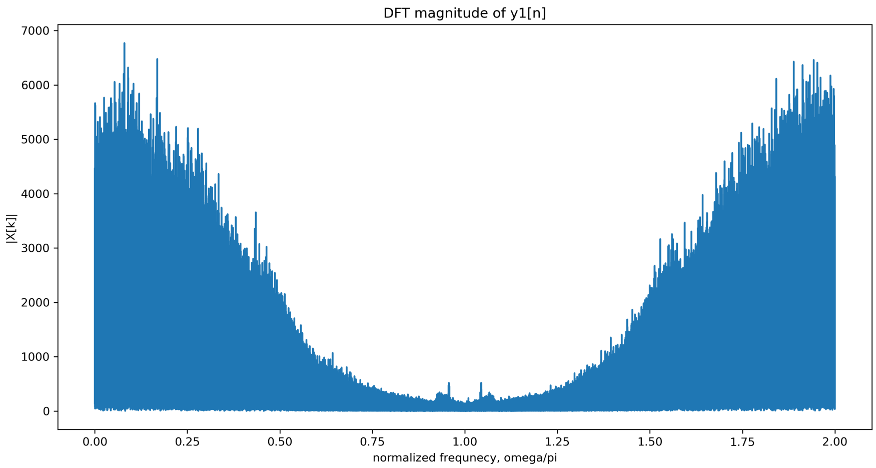

Figure 9: DFT of $y_1[n]$

4.2 DT differentiator

The DT differentiator we are going to use is a ideal linear phase, type IV FIR DT differentiator multiplied by a window. Like the CT differentiator, the DT differentiator involves a $j\omega$ term. In order to phase linear phase, the DT differentiator is given by:

$$H_{diff} = (j\omega)e^{j\omega M/2}, -\pi < \omega < \pi$$

and the ideal impulse response has the form:

$$h_{diff}[n] = \frac{cos(\pi(n - M/2))}{n - M/2} - \frac{sin(\pi(n - M/2))}{\pi(n - M/2)^2}$$

We can multiply $h_{diff}[n]$ by a filter of length (M+1) to get a type III or IV filter. A type IV filter is best suited for designing a differentiator because it doesn't have a pole at π. Ideally we have a filter with a magnitude that grows linearly with ω and has a slope of 1.

Filter Design Parameters

- Number of taps: 16, M =15
- Filter Type: Type IV FIR filter
- Beta: 1.333
- Transition width: $< \Delta\omega = 0.05\pi$
  Since the ideal differentiator has a magnitude that is linear and without large discontinuities, the transition width can be a little larger. This is unlike the ideal LPF where we have to approximate a sharp discontinuity

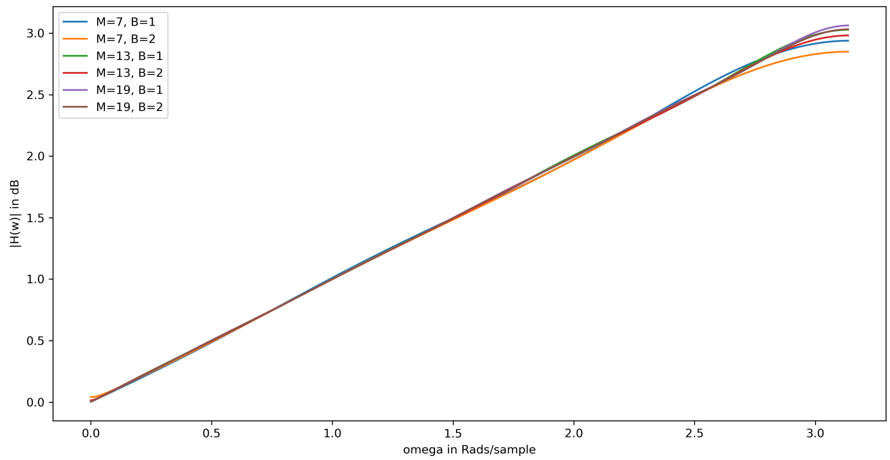

Figure 10: Differentiator with different parameters for kaiser window. Increasing the order of the filter doesn't change too much the frequency response of the fitler. Increasing the order does extend the linear regime of the filter to higher values of ω

happening at the cutoff frequency. With the Kaiser filter parameters above we can achieve this.

- Ripple: $<$ 25 dB
  With the Kaiser filter parameters above we can achieve this constraint

We apply the DT differentiator to $y_1[n]$ to get $y_{1b}[n]$. However, we need to multiply by the conjugate in order to remove the unnecessary exponential term. Because the differentiator also introduced a timeshift, we must time shift the conjugate as well.

4.3 Conjugation and Time Shift

We can first conjugate our signal $y_1[n]$ to get $y_1^*[n]$, now we can move on to time shifting this signal.

We chose a value of M = 15, meaning that our time shift, M/2 is a time shift by a factional delay. This can be interpreted as the samples of a timeshifted by $\frac{Mf_s}{2T}$ version of the bandlimited interpolation of $y_1^*[n]$. An ideal impulse response function that implements the fractional delay is a sinc function centered at $\frac{M}{2}$.

$$h_{delay} = \frac{\sin(\pi(n - \frac{M}{2}))}{n - \frac{M}{2}}$$

To get an FIR window, we multiply $h_{delay}[n]$ with a rectangular window of length M + 1. This allows our FIR filter to be symmetric about M/2. The

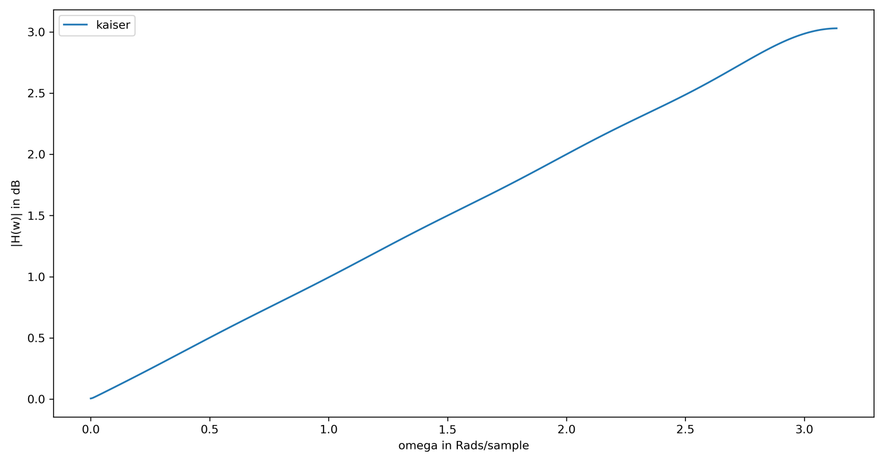

Figure 11: Frequency Response of Differentiator with kaiser window M = 15, Beta = 1.333

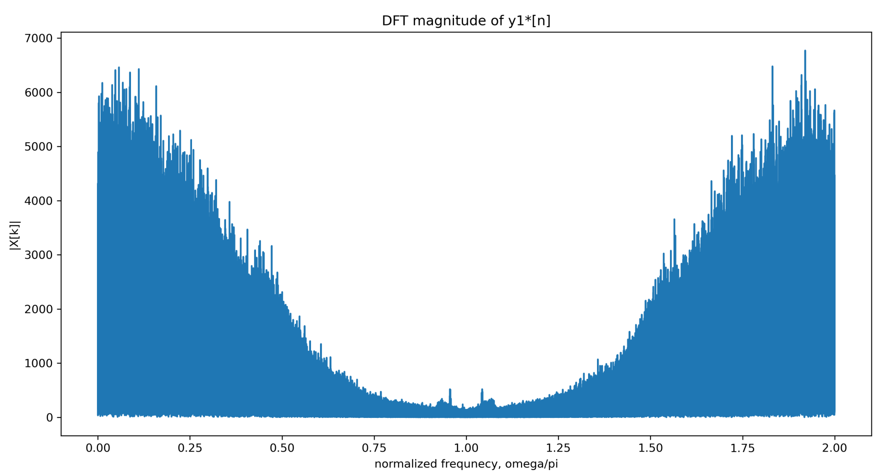

Figure 12: $y_1^*[n]$

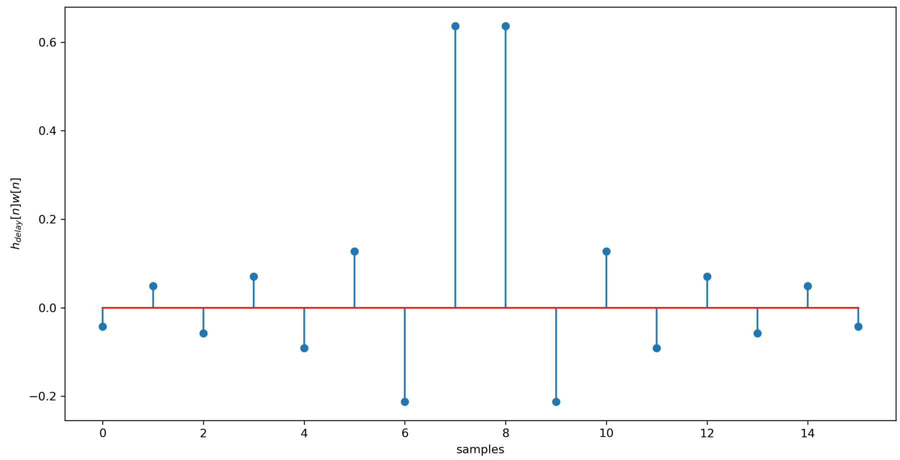

Figure 13: The impulse response of $\hat{h_{delay}}[n]$, the windowed version of a ideal fractional delay filter

impulse response of the windowed filter $\hat{h_{delay}}[n]$ is shown in the figure 13. This filter is applied to $y_1^*[n]$ to get $y_b[n]$, and figure 14 shows the a section of the signal before and after applying the delay.

4.4 Extracting the message

Now that we have $y_{1a}[n]$ Taking the imaginary part of $y_2[n]$ gives us $m[n]$, which contains the message, although it requires further processing to remove the preemphasis and extract the relevant sections.

5 Deemphasis Filter

The deemphasis filter in CT is given by

$$H_d(j\Omega) = \frac{1}{1 + j\Omega\tau_d}$$

We have to approximate it in DT, so we can use either the bilinear transform or the impulse invariance method. The impulse invariance method is straightforward to implement, so we will continue with that.

The CT filter can be written in terms of partial fraction expansion with a single term.

$$H_c(s) = \sum_{k=1}^{N} \frac{A_k}{s - s_k}$$

$$h_c(t) = \sum_{k=1}^{N} A_k e^{s_k t} \; for \; t >= 0$$

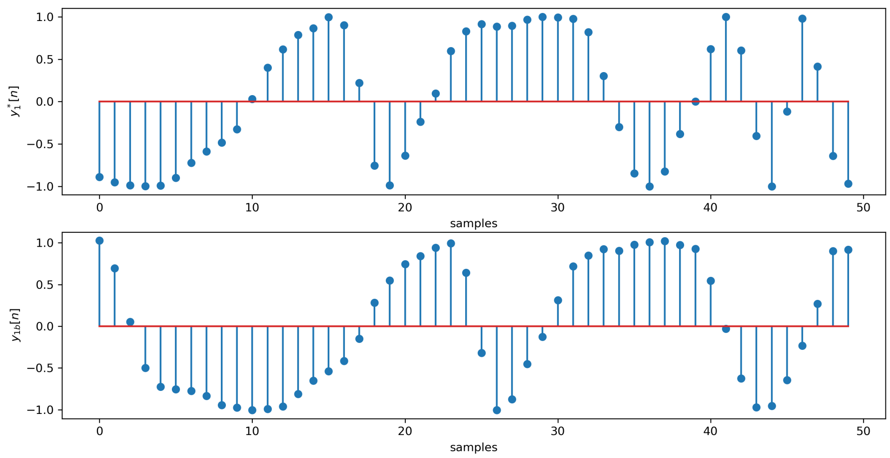

Figure 14: The windowed fractional delay filter $\hat{h_{delay}}[n]$ is applied to $y_1^*[n]$ to get $y_{1b}[n]$. This figure shows the a section of the signal before and after applying the delay. The values plotted on the y axis is the imaginary part.

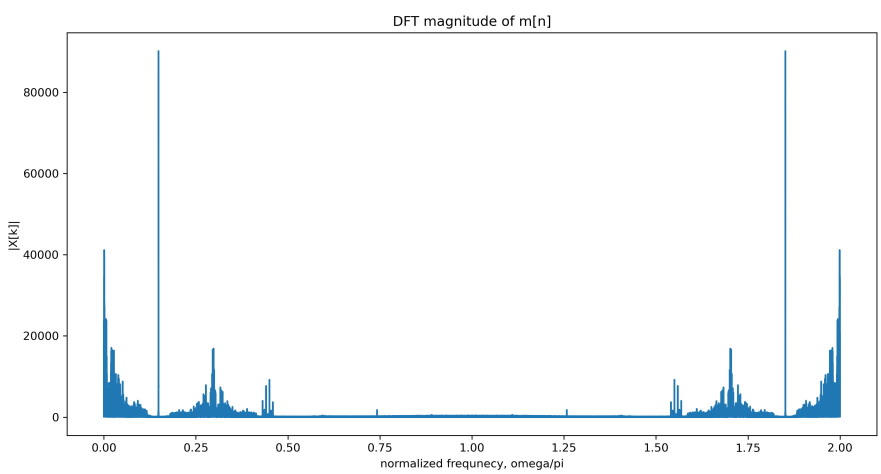

Figure 15: $m[n]$, the output from the frequency discriminator

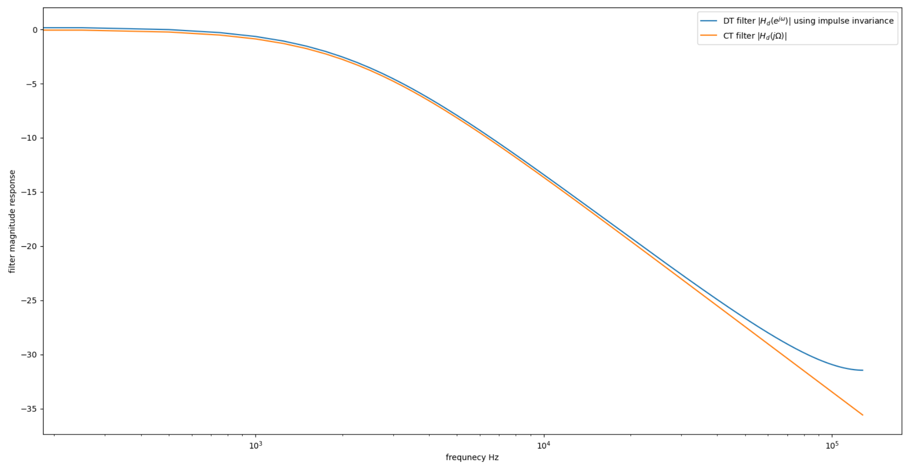

Figure 16: Comparison of the DT deemphasis filter with the CT filter. Y axis is in log scale.

Then the DT filter can be written as

$$h[n] = \sum_{k=1}^{N} T_d A_k e^{s_k n T_d} u[n]$$

and

$$H(z) = \sum_{k=1}^{N} \frac{T_d A_k}{1 - e^{s_k T_d} z^{-1}}$$

for the CT deemphasis filter given above, we can write the DT version as

$$H_d(z) = \frac{T}{T_d - T_d e^{-\frac{T}{T_d}} z^{-1}}$$

This can also be written in a LCCDE:

$$y[n] = e^{\frac{-T}{T_d}} y[n-1] + \frac{T}{T_d} x[n]$$

we can apply this IIR filter on our signal to get an output $m_d[n]$, where T is chosen to be the sampling rate.

6 Low Pass Filter for message band selection

The message that we care about is mono audio located in the frequency range $|f| \leq 15$ kHz. We can design a LPF to extract the mono audio signal.

LPF design parameters

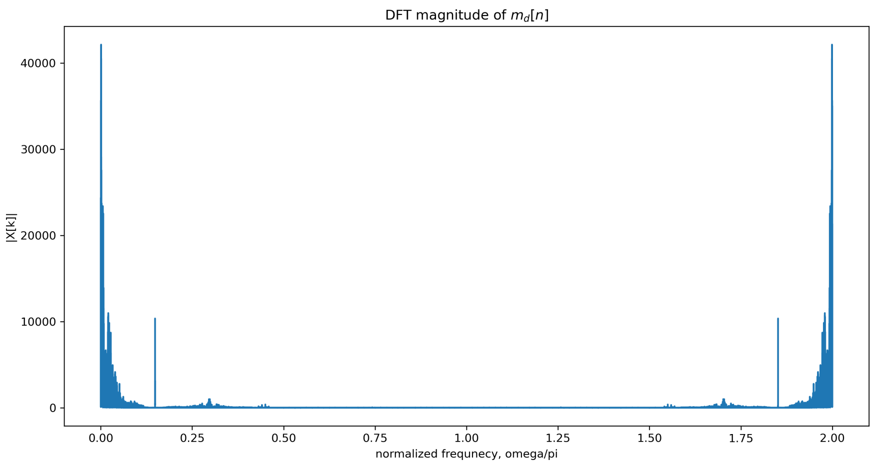

Figure 17: DFT of $m_d[n]$

- Window Type: Parks McClellan LPF
- Stop band ripple: $\delta_{s_{(dB)}} = -60$ dB The magnitude of the pilot tone that is just out past the stopband has a magnitude of around 2000. We don't want the pilot tone to show up in our signal, so we must choose a stopband ripple that minimizes this tone. If we scale by a factor of 10^{-3} or -60 dB then that will be much smaller than the DFT magnitude of the desired signal, which is in the range of 100 to 14,000. $\delta_{s_{(dB)}} = 20log_{10}(\delta_{s_{(linear)}})$
- Pass band ripple = $\delta_{s_{(linear)}}$ = (-10**(-1/20) + 1) = 0.10879
  This ripple is small enough that the relative magnitude of the DFT of the band of interest is still much larger than the DFT magnitude of the stop band signal after passing through the filter.

The result of applying this LPF is the mono audio signal $a_m[n]$, with DFT shown in figure 18.

7 Results

After decimating by a factor of M = 4, the final signal was written to a wav file with a sampling rate of 64000 Hz. The DFT of the audio signal is given by $a[n]$ in the figure 19. The audio was a clip from the song "Don't Stop Believing" by Journey. I was not expecting the audio signal to sound so clear and had expected there to be some artifacts or noise to remain in the signal that may require more advanced FM demodulation techniques to address.

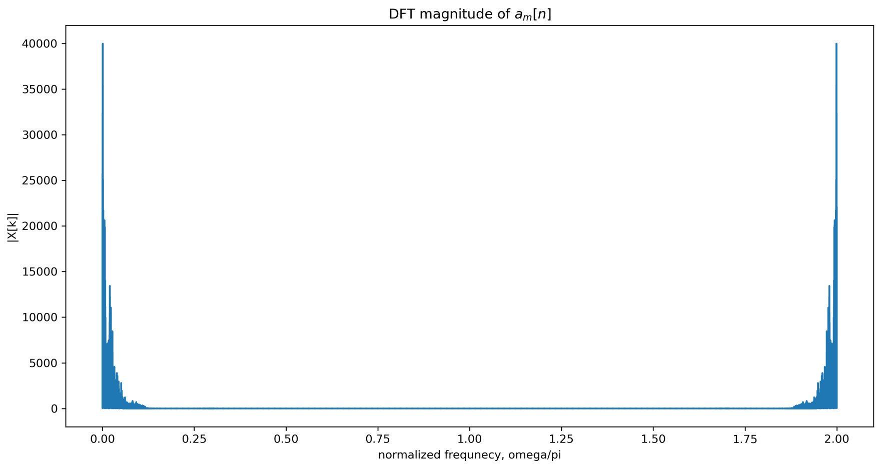

Figure 18: mono audio signal $a_m[n]$ after passing $m_d[n]$ through a LPF

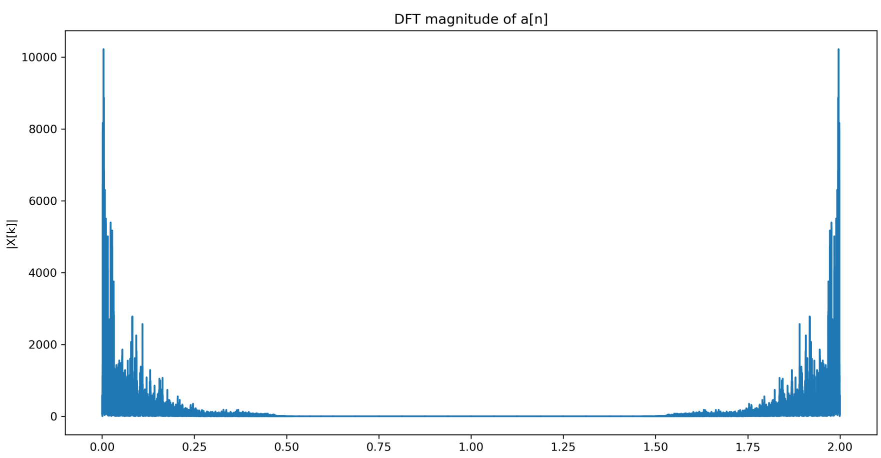

Figure 19: DFT of the mono audio signal.