

FM

Under guidance of Dr.GVV SHARMA

March 20, 2023

Contents

1	Abstract	1
2	Bandwidth	1
3	Steps to calculate bandwidth	1
4	Modulation	2
5	Demodulation	3

1 Abstract

Finding the bandwidth of an audio signal which is .wav file.

2 Bandwidth

Bandwidth can be calculated using various methods. Few of them are:

1. The bandwidth of an audio signal can be calculated as the difference between the highest and the lowest frequency components present in the signal.
2. **Power Spectral Density:** The bandwidth of a signal can be estimated using its power spectral density by analyzing the frequency range over which a significant portion of the power of the signal is concentrated.
3. **Half Power Bandwidth:** The method involves finding the frequency range over which the PSD of the signal is greater than

or equal to half of the total power of the signal. This frequency range is known as the half-power bandwidth.

3 Steps to calculate bandwidth

1. **Loading Audio file:** We begin by loading the audio file using the `wavfile.read()` function from the `scipy.io.wavfile` module. This returns the sampling frequency and the audio data as a numpy array.
2. **Computing the Fourier Transform:** The Fourier transform of the audio signal using the `numpy.fft.fft()` function. This transforms the audio data from the time domain to the frequency domain.
3. **Calculating Power Spectral Density:** We calculate the power spectral density (PSD) of the signal by taking the absolute value of the FFT and squaring it.
4. **Calculating frequency range:** We also calculate the frequency range using NumPy's `fftfreq` function, with the sampling interval set to `1/sample_rate` to account for the sample rate of the audio file.
5. Finally, we find the frequency range with significant power by applying a mask to the PSD and computing the minimum and maximum frequencies. The bandwidth is simply the difference between these frequencies. we obtain the bandwidth of the audio as 2 khz.

Fig. 1 shows the audio spectrum

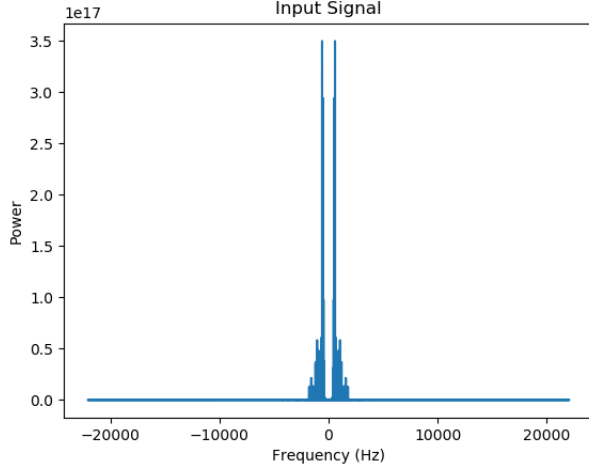


Figure 1: Spectrum of audio signal

The below python code computes bandwidth of the input audio file:

```
/FM/codes/bandwidth_audio.py
```

4 Modulation

$$c(t) = A_c \cos(2\pi F_c t) \quad (1)$$

Above equation shows the mathematical expression for a carrier signal with amplitude A_c , frequency F_c .

The frequency modulation is applied to the sound signal using the formula

$$s(t) = A_c \cos\left(2\pi F_c t + 2\pi K_f \int_0^t m(\tau) d\tau\right) \quad (2)$$

where A_c is the amplitude of the carrier signal, F_c is the frequency of the carrier signal, $m(t)$ is the modulating signal, K_f is the frequency sensitivity constant of the modulating signal.

Fig. 2 shows fm spectrum.

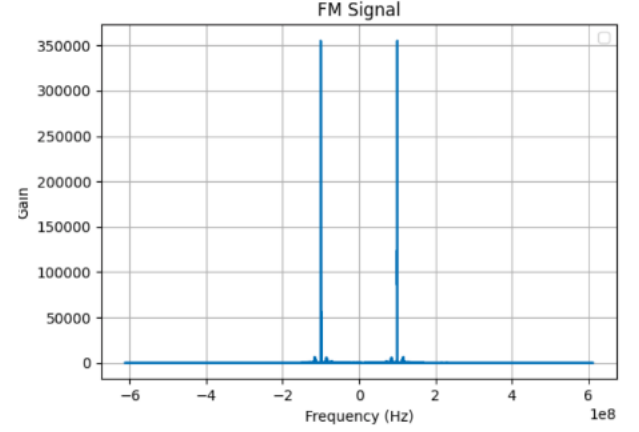


Figure 2: FM spectrum

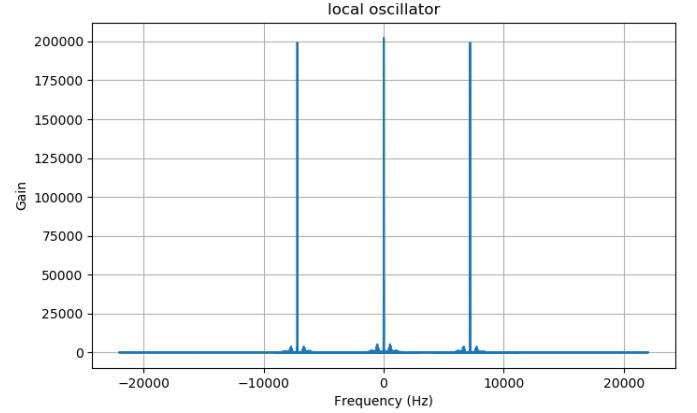


Figure 3: Local Oscillator Output

The bandwidth of the FM signal generated using a carrier frequency of 100 MHz and frequency sensitivity of 25 kHz is approximately 7 kHz. This was calculated by finding the spectral density of the FM signal using the Fourier transform.

The below python code computes bandwidth of the FM signal:

```
/FM/codes/Modulation.py
```

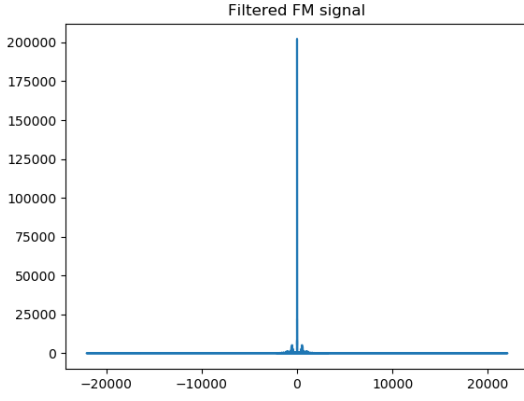


Figure 4: Filtered FM signal

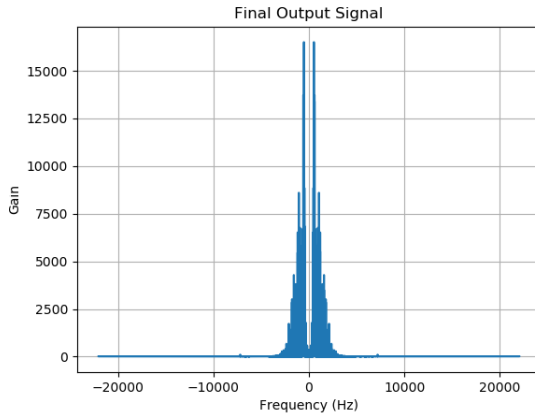


Figure 5: final output signal

5 Demodulation

FM signal is multiplied with local oscillator.
Fig. 3 shows the output of it

$$Y = s(t) \cos(2\pi F_c t) \quad (3)$$

Low Pass Filter:

low-pass filter removes the higher-frequency components of the DSB-SC signal, leaving only the sum frequency component (i.e., the FM signal) in the output.

Butterworth filter with order 6 and cutoff frequency 3khz gives the filtered FM signal.

Fig. 4 shows the Filtered FM signal

Cos inverse is applied and differentiated to extract message signal.

Fig. 5 shows the final output signal.

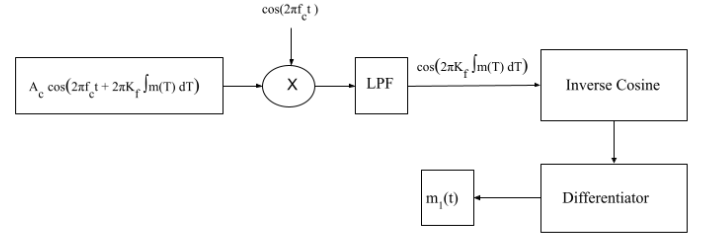


Figure 6: Block diagram

The below python code generates the demodulated signal:

```
/FM/codes/fm.py
```