Università degli Studi di Padova

Department of Information Engineering

Master’s degree in ICT for Internet and Multimedia

Course: Digital Signal Processing

Final project report

Davide Carta

ID: 1210702

Abstract

In this report the procedure and intermediate steps that successfully lead to the development of the final project for the *Digital Signal Processing* course, are explained. The audio file used as input is: signal_124.wav

1. **Input signal spectrum**
   To read the input audio file, the built-in function *audioread( )* was used and relevant information such as number of bits and duration, were extracted through the function *audioinfo( )*.
   To compute the spectrum of the input signal, the Fast Fourier Transform was performed calling *fft( )*. In the figure below the spectrum is shown.

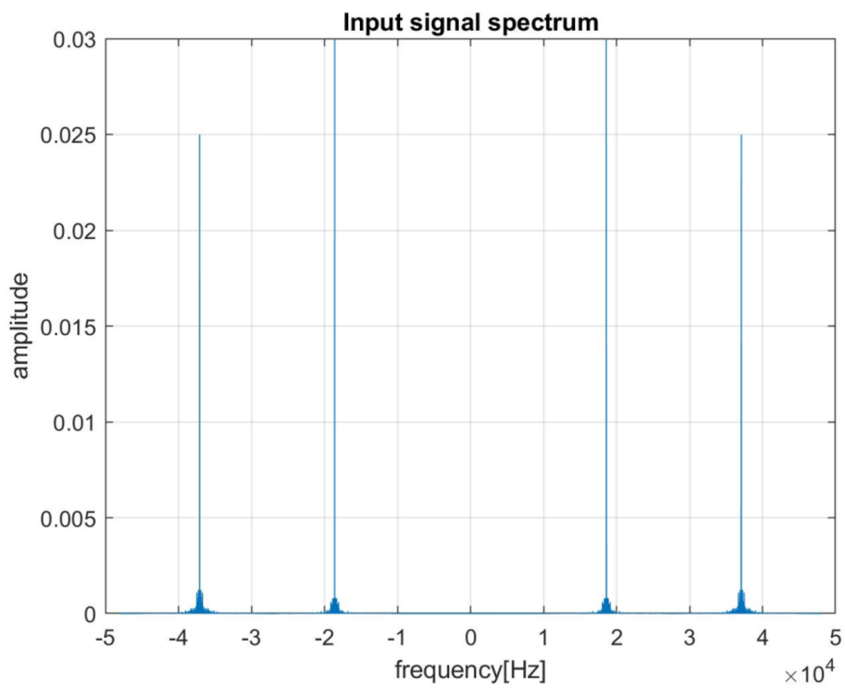

Figure 1. Input signal spectrum

2. **Carrier frequencies detection**
   First, the continuous components were shifted to the center calling *fftshift( )*. Then, the spectrum power was computed and used as threshold to spot the carrier frequencies. Such carriers were found to be f_1 = 18600 Hz and f_2 = 37100 Hz.

3. **Carrier frequencies amplitude estimate computation**
   The amplitude was found, first by normalizing the amplitude spectrum, then getting the *y* value at both carriers. Such amplitudes were found to be A_1 = 0.025 and A_2 = 0.03 as can be seen in the input signal spectrum plot at step 1 above.

4. Carrier frequencies extraction

For this step, a narrowband notch filter was implemented through a 2nd order IIR filter having transfer function: $H(z) = \frac{b_0}{1 - a_1 \cdot z^{-1} - a_2 \cdot z^{-2}}$. An additional constraint was set to make the filter very narrow: $\Delta f_{3db} = 2\ Hz$.

In the figure below the filter's features, for carrier at frequency f_1 and f_2 respectively, are shown.

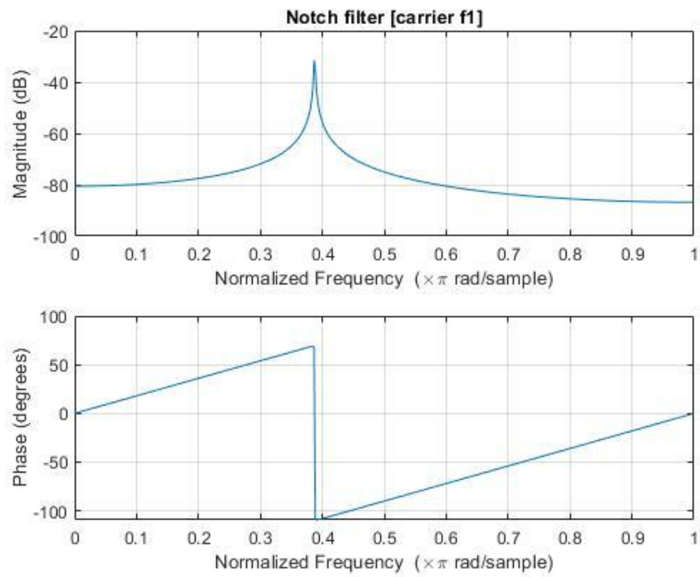

Figure 2. Filter's features, for carrier at frequency f_1

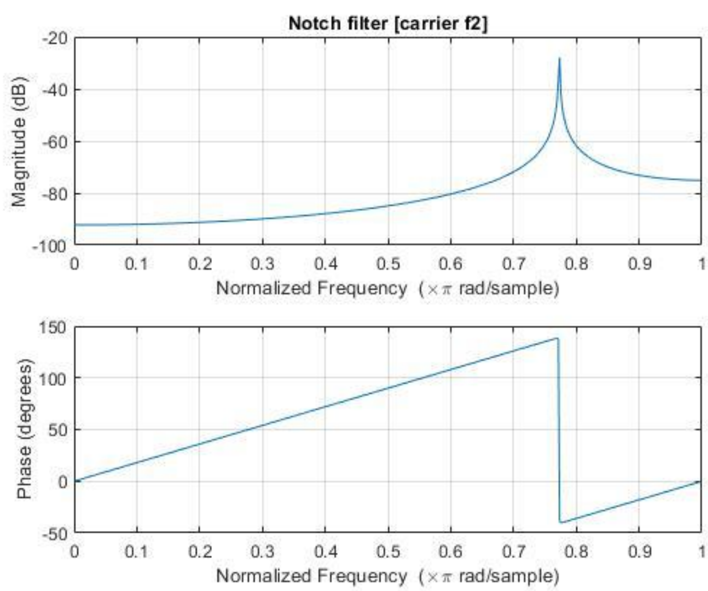

Figure 3. Filter's features, for carrier at frequency f_2

5. **Signals demodulation through the extracted carriers**
The signal is a double sideband with reduced carrier. The demodulation operation was performed multiplying the information signal by a *cos( )* function with suitable parameters related to both carriers. The first information signal was then obtained multiplying the original information signal by the carrier related to f_1. The second information signal was obtained multiplying the original information signal by the carrier related to f_2.

6. **Filters design**
The purpose of the filter is to remove any distortion in the demodulated signal. It was implemented using a cascade of a low-pass FIR filter and a high-pass notch IIR filter.

6.1 Low-pass FIR filter

The table below report the low-pass FIR filter's features.

Passband frequency	*Stopband frequency*	*Attenuation*	*Peak to Peak ripple*
8000 kHz	8500 kHz	80dB	0.001

Table 1. Low-pass FIR filter features.

The figure below shows the low-pass FIR filter's frequency response.

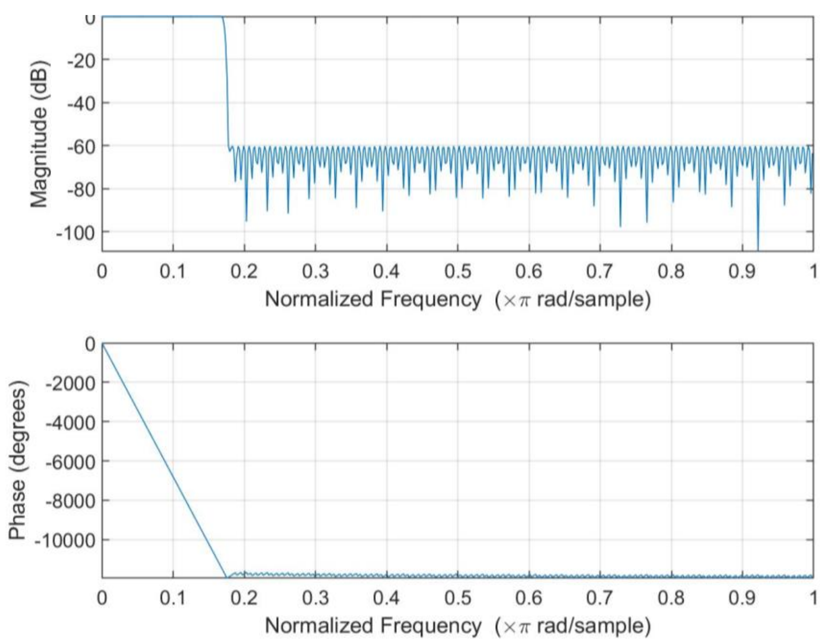

Figure 4. Low-pass FIR filter frequency response

6.2 High-pass IIR filter

It is a high-pass second order IIR filter with transfer function:

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 - a_1 \cdot z^{-1} - a_2 \cdot z^{-2}}$$

and additional constraint:

$$\Delta f_{3db} = 2\ Hz.$$

It has a very narrow bandwidth to remove the continuous component.
The figure below shows the high-pass IIR filter's frequency response.

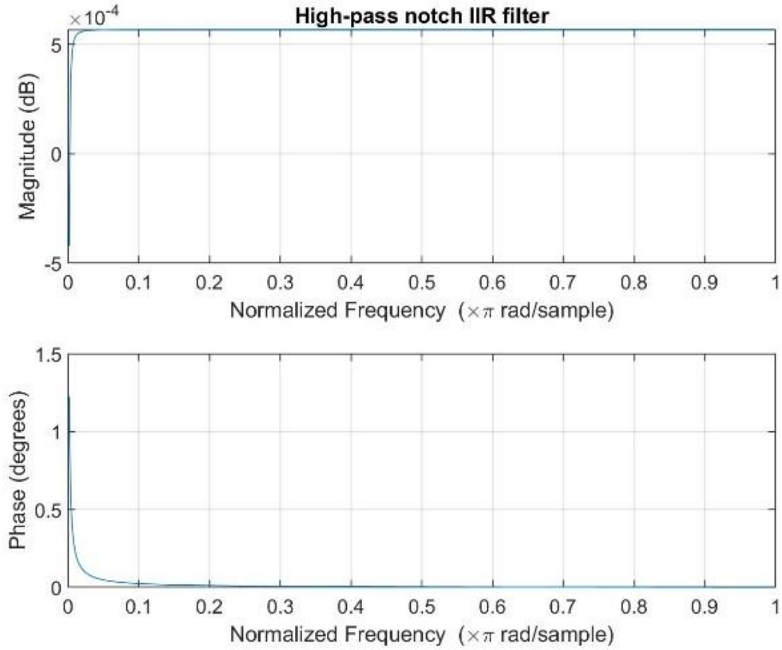

Figure 5. High-pass IIR filter frequency response

7. Demodulated signals spectrum

To compute the spectrum, the Fast Fourier Transform was performed calling *fft( ),* then shifting to the zero frequency and normalization by the number of samples were performed.
Below the spectrum of the signal demodulated with f_1 and f_2 respectively are shown.

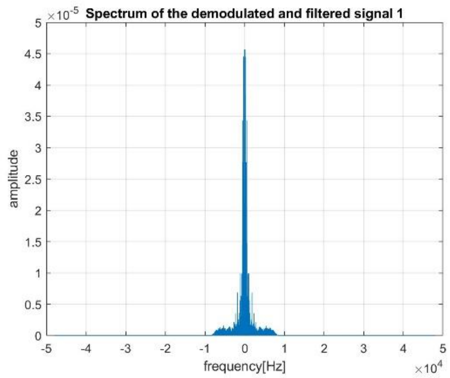

Figure 6. Demodulated signal spectrum at carrier f_1

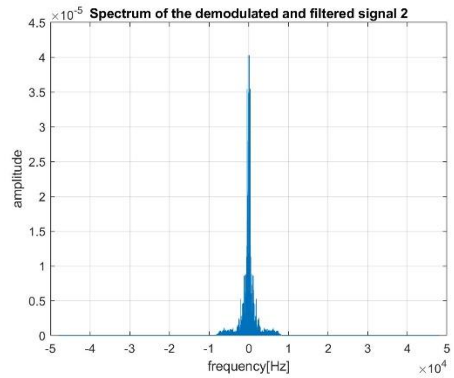

Figure 7. Demodulated signal spectrum at carrier f_2

8. **Clean signal saving**

   The output file is a two-channel stereo file thus, a vector containing the two different signals, extracted during the above steps, must be provided. Before saving the final result, the signal was amplified. To save the final clean audio signal the function *audiowrite( )* was called.

   The clean signal is a piece of the song *“Father and Son”* by Cat Stevens from the album “Tea for the Tillerman”, 1970.