B.Sc. in Electronic and Communications Engineering

Final Project

Simulation of a communication system

Author: Giordano Andrea

ID: s310679

Contents

1 Abstract

This project focuses on the discussion of two tasks: removing sinusoidal noise from a given set of discrete-time signal and the modulation of the signal so that they can be transmitted over a channel, already occupied, in the base-band, by another signal. Different techniques have been implemented to remove the above said noise, in particular, the usage of a 4-th order Bessel filter and the design of a digital filter by placing manually both zeroes and poles in the complex plane. Double sided amplitude modulation is implemented. At the receiver side, to obtain the demodulated signal, a low-pass filter is deployed. Again both a Bessel and a costum IIR filter are used, and the result is compared, demonstrating the importance of both optimal carrier frequency selection and cut-off frequency.

2 Introduction

The analysis has been carried out using two different songs:

- "Imagine" by John Lennon (from now on Song1)
- "Mamma mia" by ABBA (from now on Song2)

We were given, for both the songs, the reference audio, which is the song without any disturbance. As the vast majority of the discrete-time songs available nowadays, the sampling frequency is $Fs = 24.10kHz$. The reference signal was gives as stereo audio, to simplify the communication system's simulation, it was converted it in a mono signal (selecting all the element of the first column).

The frequency content of "Imagine" and "Mamma Mia" is distributed differently, meaning a single filter design cannot be applied to both signals. Instead, the cut-off frequency or pole placement must be adjusted to account for these variations. Ignoring these differences would result in suboptimal filtering, either removing important musical components or leaving residual noise in the output.

The first part of this report deals with removing a sinusoidal disturbance, by implementing

- a **4-th order Bessel low-pass filter**, designed using the bilinear transform method

- a **custom IIR filter**, designed by placing two complex conjugate poles and zeros

While the first method attenuates all the frequency above the cut-off frequency, by accurate placing the zeroes and the poles it is possible to remove from the song only some specific frequency, thus for the second method, a higher SIR is expected.

The second part of the report focuses on the transmission of a signal into a channel. Different cases are tested, transmitting one song at a time and both the second together. In either case, since the baseband is occupied by some noise, double sideband amplitude modulation is employed. The study explores the impact of carrier modulation on the SIR of the received songs.

3 Recurrent functions

To make the MATLAB script more readable and less repetitive, various function have been written and then used in the main file.

3.1 determine_SIR

SIR (Signal Interference Ratio) is a quantity that shows how much the noise is affecting a signal. It is calculated, by definition, as the quotient between the power of the noisy signal and the power of the equivalent noise, where the equivalent noise $e[n] = x_{noise}[n] - x_{ref}[n]$. It is known that for time-discrete signal the power is defined as $P_x = \frac{1}{N} \cdot E_x$, where N is the extension of the signal. The dimension of the noisy signal has to be identical to the extension of the reference signal (else they won't be the same signal any longer), thus the ratio of the power can be rewritten as the ratio of the energy of the signals. Given the time-discrete signal $y[n]$, with N_y non zero elements, then the energy of the signal is defined as:

$$E\{y[n]\} = \sum_{n=0}^{N_y-1} |y[n]|^2 \tag{1}$$

The SIR can be then written as:

$$SIR = \frac{P_{x_{noise}}}{P_e} = \frac{E\{|x_{noise}[n]|^2\}}{E\{|e[n]|^2\}} \tag{2}$$

The designed function takes two inputs:

- x_{noise}: signal with noise (either original noisy signal or signal after being filtered
- x_{ref}: reference signal

In the function the disturbance signal and the energy of all the relevant signals are determined . The output of the function is the value of the SIR in decibels

$$SIR_{dB} = 10 \cdot \log_{10} SIR_{natural\ unit}$$

3.2 plot_SIR

The aim of this function is to represent graphically SIR values versus frequency. This function takes as inputs:

- `sir`: a vector containing SIR values for each of the tested frequency
- `freqs`: the vector containing the corresponding frequencies
- `maxSIR`: he maximum SIR value achieved
- `optimalFreq`: the frequency at which the maximum SIR occurs

Clearly the dimension of `sir` and of `freqs` must agree, else the function will return an error and the execution of the whole simulation will be interrupted. The output of this function is a plot, that presents on the x-axis the frequencies, while on the y-axis the value of the SIR id decibels. Furthermore, on the plot, the best SIR value for the recovered signal is marked particular attention, highlighting the frequency where this result is obtained (either filter cut-off or poles' frequency).

3.3 plot_signal_spectra

This function visualizes the spectral characteristics of a given signal using two different representations:

- normalized PSD (Power Spectral Density) in dB (logarithmic scale)
- magnitude squared spectrum (linear scale)

The input it requires are

- `f`: frequency axis
- `spectra`: Fourier transform of the signal
- `title_text`: not strictly necessary, is the title of the figure

To normalize the PSD in decibels, instead of dividing by the maximum value, it is sufficient to subtract the logarithm of the maximum spectral value. This ensures that the peak value is set to zero decibels. This approach follows the logarithm property: $10 \cdot log_{10}(\frac{PSD}{PSD_{max}}) = 10 \cdot log_{10}(PSD) - 10 \cdot log_{10}(PSD_{max})$. The output of the function is a figure containing the two subplots described above: on the left the normalized PSD, on the right the magnitude square spectrum.

4 Task 0: perform general analysis of the audio signals

4.1 Analysis of the reference signals

An initial analysis was conducted to collect general information about the song and the noise. Since the disturbance added to both the song is the same, it is possible to use filter with almost identical characteristic to clean both of the songs; this is the reason why it was possible to implement a `case switch` selecting one song at a time, allowing to work with only one song. Once the reference song was loaded onto MATLAB, the number of samples N was defined. With this quantity, both a time and frequency axis were determined: the time vector contains linearly spaced points ($\Delta T = \frac{1}{Fs}$) spacing in the interval $[0, (N-1) \cdot \Delta T]$; similarly the frequency axis f was defined over the interval $[-\frac{Fs}{2}, \frac{Fs}{2} - \Delta F]$ with step equal to $\Delta F = \frac{Fs}{N}$. Needless to be said, the axis of the noisy song and of the reference signal are the same.

As already mentioned above, the sampling frequency is $44.100kHz$. This means, applying Shannon-Nyquist theorem, that the maximum bandwidth that the signal can have is half of the sampling frequency ($BW_{max} = \frac{F_s}{2} =$

$22.050khz$). This, though, would be unlikely, accounting for the fact that humans can only hear frequencies up to $20kHz$. Having an higher frequency that the minimum required, allow the existence of a transition band between consecutive copies of the DTFT, which is important to avoid aliasing. If ideal low-pass filters were realizable, no oversampling would be necessary. However, an ideal low-pass filter would require an infinite and non-causal impulse response, making it impossible to implement. In real signal processing, practical low-pass filters exhibit a gradual transition, not as sharp transition as the ideal one (Heaviside Pi), and hence oversampling must be consider to provide a transition band, where the attenuation on the signal gradually increases. The extra bandwidth ensures that an anti-aliasing filter can sufficiently attenuate unwanted high-frequency components before downsampling occurs. In this case, the choice of a 44.1 kHz sampling rate allows a smooth roll-off of the filter while preserving the full audible range up to 20 kHz, ensuring minimal distortion and optimal audio quality.

Figure 1 presents, for Song1, the spectra of both the original song (Figure 1a) and the noisy song (Figure 1b). The analysis of Figure 1a provides valuable insights into the signal's frequency content. Specifically, by examining the magnitude squared spectrum (MSS) in the right subplot, it becomes evident that most of the signal's informations are concentrated around the centre of the frequency axis (low frequencies). Indeed, a closer inspection of the plot, reveals that frequency components above $4kHz$ are negligible compared to those at lower frequencies (figure 1c). This observation is further confirmed by the normalized Power Spectral Density (PSD), which shows that components above $4kHz$ have power levels at least $50dB$ lower than the most dominant frequency components. That does not mean that the component above this frequency are useless: if those were not presents, it still would be possible to recognize the song, some differences (in tone and timbre) would become noticeable. Higher-frequency components contribute to the finer details of the audio signal, affecting aspects such as clarity, brightness, and the perception of certain instruments or vocal harmonics. Figure 1d presents the signal in the time domain. Although this representation does not provide significant insight into the spectral content of either the signal or the noise, it is still included for completeness. The only appreciable difference visible between the two representation is at the end of the song: in the reference, the audio gradually fades out, in the noisy version,instead, the noise is not attenuated (the noise is independent of the song's dynamic) and oscillates (the noise is a pure waveform).

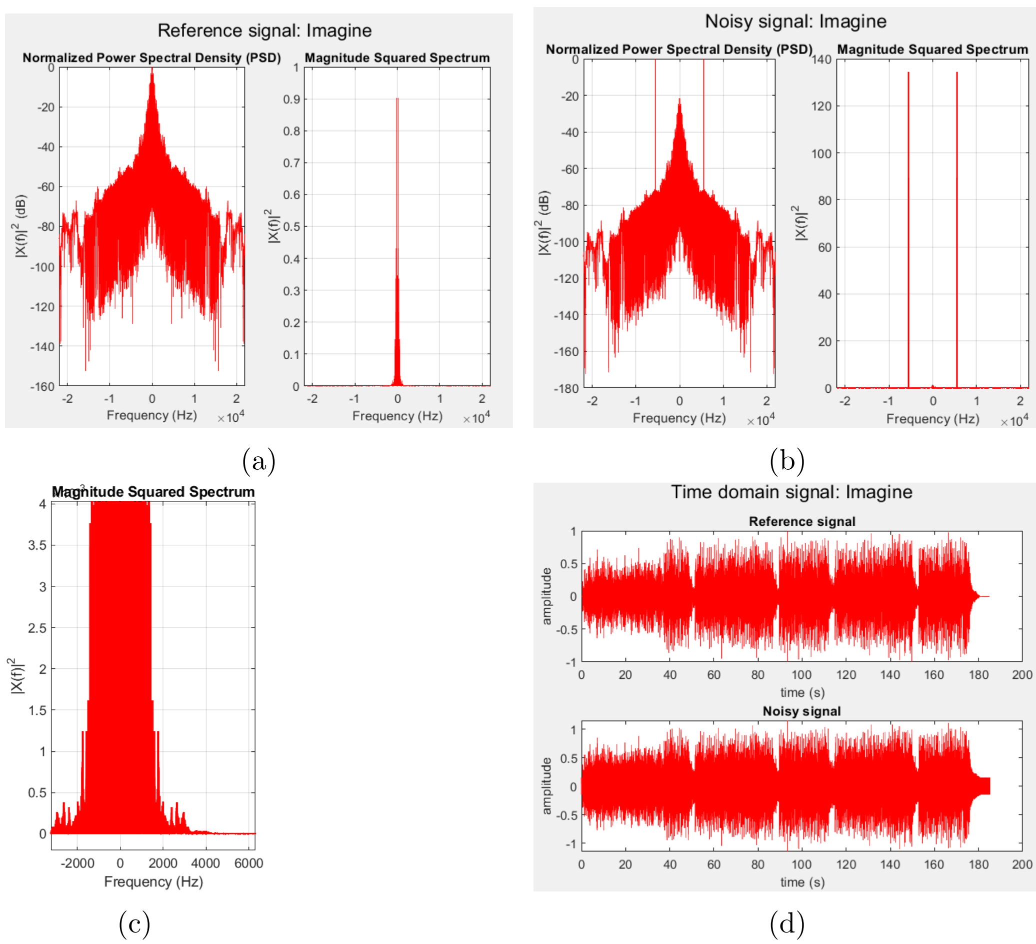
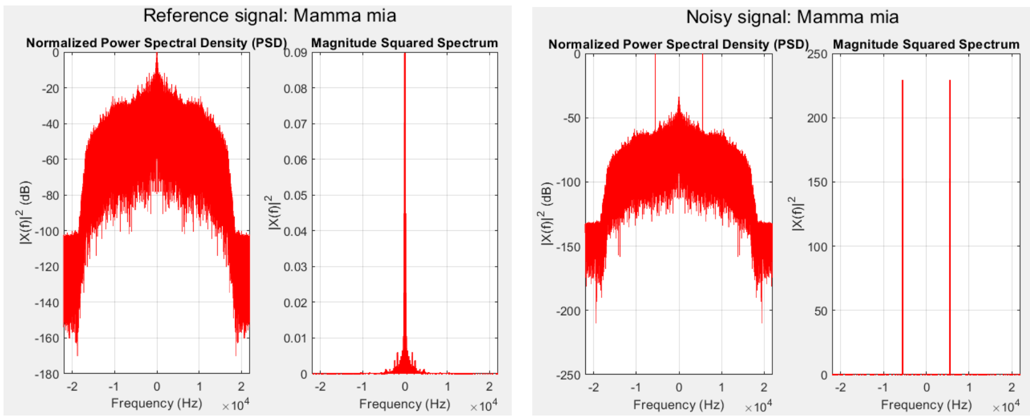


Figure 1: Song1: (a) reference song; (b) song with noise; (c) enlargement of the reference song, highlighting frequency components around $4kHz$; (d) song in the time domain (reference signal above, noisy signal below)

A similar analysis to the one presented above, can be conducted on Song2, shown in figure 2. If in a way the result may be comparable, both the signal have higher components at lower frequency, in case of Song2, components at medium frequency are not as weak as in the case of Song1. In the graph for the normalized PSD of Song2, it is impossible not to notice that at very high frequency (with respect to the bandwidth of the signal), the power components are much attenuated (more than $100dB$ less than the maximum value). Most likely this is due to the usage of an anti-aliasing filter, so that if the song happens to be transmitted, the high-frequency components beyond the Nyquist limit do not introduce aliasing artifacts.



(a) (b)

Figure 2: Song2: (a) reference song; (b) song with noise

Above, the maximum bandwidth imposed by the sampling frequency and by the maximum frequency perceptible by the human ear have been defined. However, one could argue that, since the signals contain very few components above $4kHz$, the actual utilized bandwidth is significantly smaller than the maximum available one. To a first-order approximation, indeed, the effective bandwidth of the signal can be estimated by considering the frequency range where the majority of the power is concentrated. It is hence possible, to consider the bandwidth for both the signals to be $BW \approx 5kHz$.

4.2 Analysis of the noisy signals

The disturbance affecting both songs is identical, however, when comparing figure 1b with figure 2b, some difference become evident. Although the noise remains the same, the impact it has on the songs is different. This variation arises from the difference in power levels between the two songs. Since the noise power is constant, the signal with lower power (in this analysis Song2) experiences a greater degree of degradation. By analysing the MSB for both the songs, one can easily conclude that the noise is much stronger than the song itself. When analysing the signal with the noise, it is easy to understand that the noise is fully represented, in the frequency domain, by two spikes: from there, it is possible to conclude that the disturbance is a sinusoidal signal, represented by two spikes at the fundamental frequency f_0 and its negative counterpart $-f_0$.

$$\mathcal{F}\{Acos(2\pi f_0 t + \theta)\} = \frac{A}{2}\delta(f - f_0)e^{j\omega} + \frac{A}{2}\delta(f + f_0)e^{j\omega}$$

From the analysis of the normalized PSD, it is possible, other than confirming the previous observation, also quantify the by how much the power of the noise is stronger than the power of each song. In the case of Song1, the central frequency components of the signal are attenuated by approximately 20 dB, whereas for Song2, the attenuation reaches about 30 dB. Remembering that the SIR is defined as the ratio of the signal over the equivalent noise, it is evident that the higher the SIR is, the least the signal is affected by the noise. Calculating the SIR for the two signals, reported in table 1, it is evident that, though both signals are higly distorted by the noise, the quality of Song2 is by far worst than the one of Song1.

Signal	**SIR (dB)**
Song1	5.65
Song2	2.54

Table 1: SIR value (in dB) for the two songs

4.3 Analysis of the noise

The frequency of the sinusoidal noise, f_{noise} can be estimated simply by the analysis of one of the graph of a noisy signal in the frequency domain. Comparing the spectrum of the reference signal with that of the noisy signal, it becomes evident that the noise frequency correspond to the dominant peak. Since the signal is real, and hence the Fourier transform symmetric, the maximum value is obtained for both negative and positive frequencies at $\pm f_{noie} = 5553.39Hz$. To obtain a more precise value, it is possible to use some simple MATLAB functions. It is enough to find the index where the Fourier transform is maximum (`[noise, ind_noise] = max(abs(Song_noise))`) and then find the frequency corresponding to that component (since to `Song_noise`, that is the *fft* of the noisy signal, has already been applied the *fftshift* command, the frequency is just `f_noise = abs(f(ind_noise))`, where `f` is the frequency axis).

5 Task 1: remove the disturbance

The aim of the first part of this report is to illustrate how different filtering techniques helps reducing the distortion introduced by the noise. In particular, the result obtained by filtering using a 4-th order digital low-pass Bessel filter are compared with the signal obtained by filtering with a filter designed "ad-hoc" for this particular sinusoidal noise. Additionally, some observations will be made regarding the application of a 4th-order Butterworth filter, instead of the Bessel filter. To compute the filtered signal the function `x_out = filter(B, A, x_noise)` is used throughout all the document. Filtering introduce a delay, hence the reference signal and the filtered one are not perfectly aligned. This can impact the SIR evaluation, so it is important to estimate the delay and realign the two signals. Luckily on MATLAB there are some functions which provide exactly this result. Since the delay is introduced by the filtering process, it is independent of the type of filter used, the usage of the following functions is taken for granted in all the possible implementations. In particular, the functions that are of interest are two:

- `delay = finddelay(x_reference, x_filtered)`, that is responsible of calculating the delay between the two signals
- `x_filtered = circshift(x_filtered, -delay)`, that applies a circular shift to *x_filtered* by *delay* ensuring realignment.

To avoid errors, it is important ensuring that all the signals are considered as column vector before applying those functions.

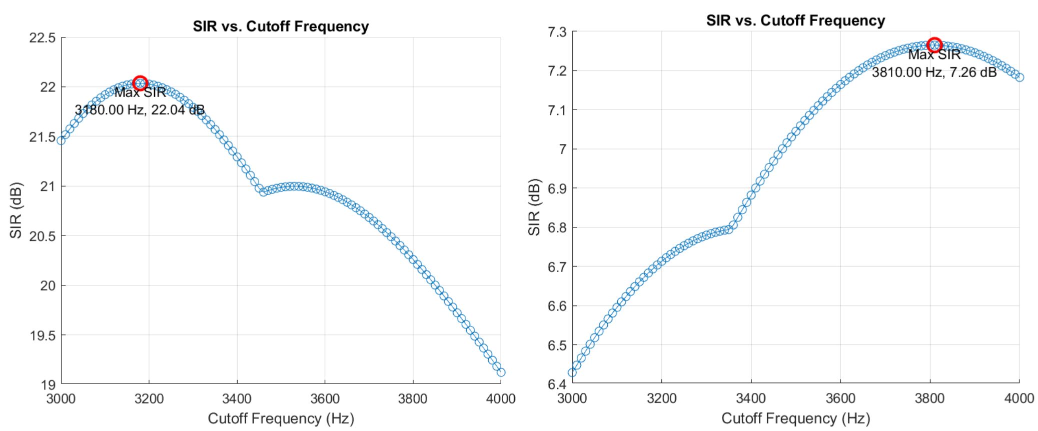
5.1 Filtering with a 4-th order low-pass Bessel filter

In the design of a 4-th order low-pass Bessel filter the only modifiable parameter is the cut-off frequency. To synthetize the filter on MATLAB the following function were used:

- `[B, A] = besself(filterOrder,` $\omega_{cut-off}$`)`: generates numerator and denominator coefficients respectively of an analogue low-pass Bessel filter, where $\omega_{cut-off}$ is the angular cut-off frequency

- `[Bz, Az] = bilinear(B, A, Fs)`: synthetize the discrete-time transfer function that mimics the behaviour of the analogue filter.

The aim was to obtain a filtered signal with quality as high as possible, in order to meet that objective, the cut-off frequency had to be carefully selected. In order to achieve so, various cut-off frequencies were tested (in a `for loop`). After several iteration to narrow the interval, a small range for the cut-off frequency was found. It was determined that the optimal cut-off frequency interval lies within the range $3kHz, 4kHz$. The actual cut-off frequency slightly depends on the song that is being filtered. In figure 3, the SIR as a function of the cut-off frequency is plotted for both songs, highlighting the optimal filtering point. Additionally, table 2 presents the optimal cut-off frequency for each song. Beyond noticing the difference in the optimal cut-off frequencies, it is also evident that the values of the SIR differs significantly. As previously discussed, this is due to the different power contents of each song with respect to the disturbance's power: Song1 is easier to recover than Song2 because it has a higher signal power relative to the noise.



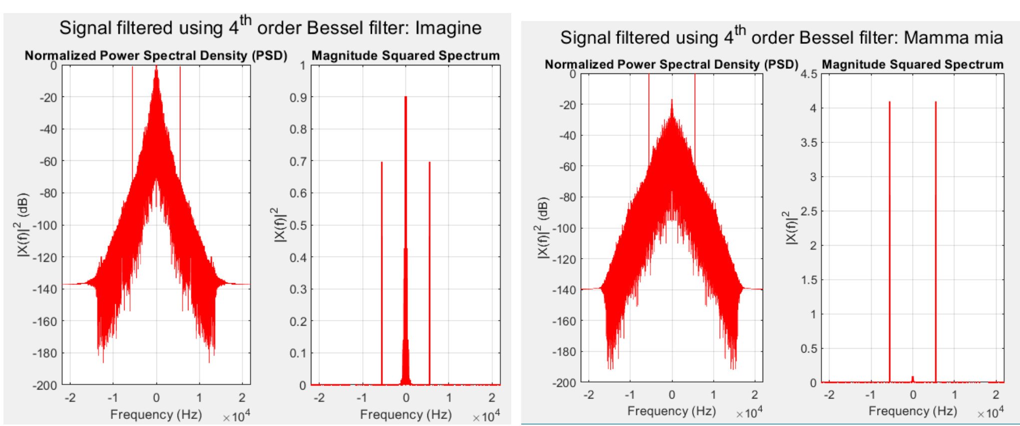
(a) (b)

Figure 3: Optimization process to determine best cut-off frequency, (SIR (dB) versus frequency (Hz)). (a) Song1, (b) Song2

Song	Cut-off frequency (Hz)	SIR (dB)
Song1	3180	22.04
Song2	3800	7.26

Table 2: Best cut-off frequency and best SIR value for each song respectively using a low-pass Bessel filter

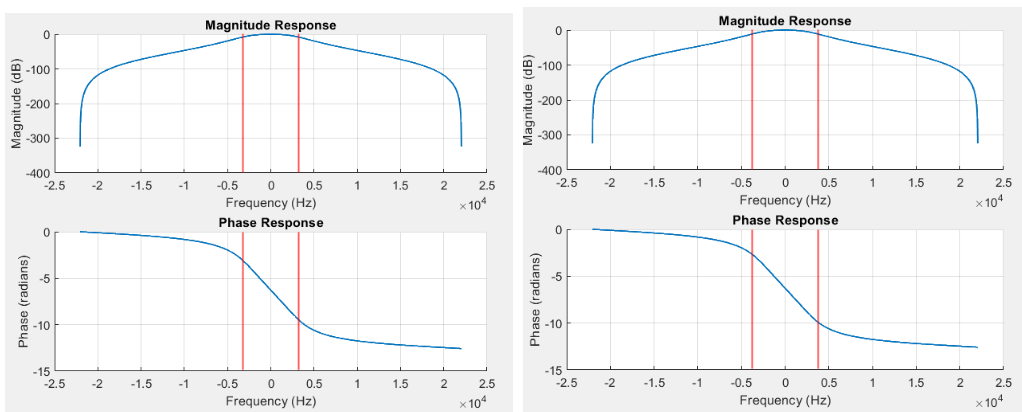
In figure 4 one can appreciate the output of the filer. It is clear that, for both the song, the disturbance is not entirely eliminated. In particular, in figure 4b, the noise remains the dominant component. However, if in the original noisy Song2, the main frequency components of the Song were attenuated by more than $30dB$ with respect to the disturbance's component, in the filtered signal, the attenuation is reduced to less than $20dB$. Filtering the noise in Song1 produce a much cleaner output: the power of the disturbance becomes smaller (but of the same order) than the one of the main component of the song. In both cases, by listening to the obtained signal, a reduction of the noise can be heard, as well as same changes in the tone of the songs. Analysing the normalized PSD it becomes clear that the low-pass filter is attenuating all the frequency above the cut-off frequency, thus the song loses all its high frequency components. Comparing the PSD in figure 4 with the PSD in figures 1 and 2 further highlights this effect.



(a) (b)

Figure 4: Filtered signal (in frequency domain, using Bessel low-pass filter. (a) Song1, (b) Song2

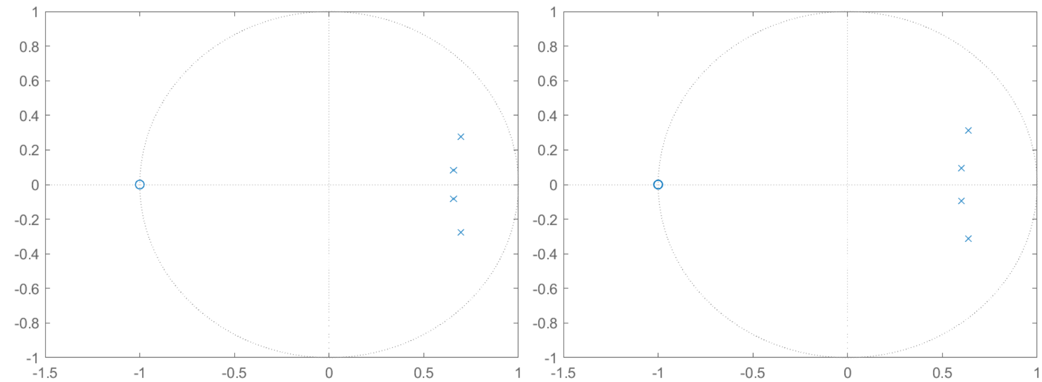
In figure 5 the impulse response of the Bessel filter are shown. One can notice that, even though the filter for each of the song differs (have different cut-off frequency), their shape, in the frequency domain is identical. What it is expected from a low pass filter is to have a passband (from zero to the cut-off frequency) with unity gain and, for frequency above the cut-off, a sharp attenuation.



(a) (b)

Figure 5: Transfer function of the optimum filter, both in magnitude (above) and phase (below). The red line represent the cut-off frequency. Optimum filter (a) for Song1, (b) for S ong2

Analogue Bessel filter are known to have maximally flat group response. The IIR filter obtained by the bilinear transformation do not have, although, constant group delay. In figure 6 is reported the pole-zero map of both the system. It is easy to notice that the two transfer function has identical zeros, for $z = 1$, but different poles (and gain); since the function implemented must be real, the poles must be complex conjugate.



(a) (b)

Figure 6: Pole-zero map for the optimum filter. Circle represent the zero, cross represents the pole. Pole-zero map (a) for Song1, (b) for Song2

5.2 Filtering by designing my own filter

Since the disturbance is sinusoidal, it is possible to design a specific filter to eliminate the particular noise's frequency. Instead of using a low-pass filter, that, as seen above, reduces all the frequency components above the cut-off frequency, a filter that eliminates only the noise's frequency (and the surrounding frequencies), may have a better output signal. To eliminate the disturbance, it is important to place the zero (a complex conjugate pair) at a specific frequency equation 3:

$$z_{1,2} = exp(\pm j2\pi \frac{f_{noise}}{Fs}) \tag{3}$$

That ensures that the filter impulse response Fourier transform $H(f)$ is zero at the disturbance frequency, explined by the relationship listed in equation 4

$$H(f_{noise}) = H(z)\Big|_{z=e^{j2\pi \frac{f_{noise}}{F_s}}} = 0 \tag{4}$$

It remains to determine the best frequency to place the poles (again one complex conjugate pair). To define the poles, the formula reported in equation 5 was used.

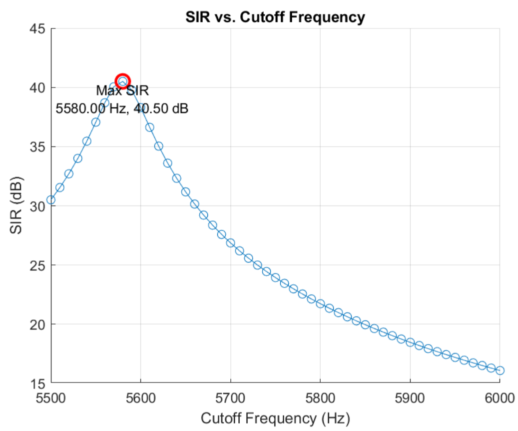
$$p_{1,2} = r \cdot exp(\pm j2\pi \frac{f_{pole}}{F_s}) \tag{5}$$

To ensure stability of the transfer function, the magnitude r of each poles has to be less than one (else BIBO stability is not ensure, and the output may oscillates indefinitely). That means that in our design $r < 1$. It can be easily shown that, the higher r is, the better the value of the SIR is and the frequency of the poles is nearer the the zeroes one. Table 3 reports the value of *SIR*, r and *pole frequency* for both the songs.

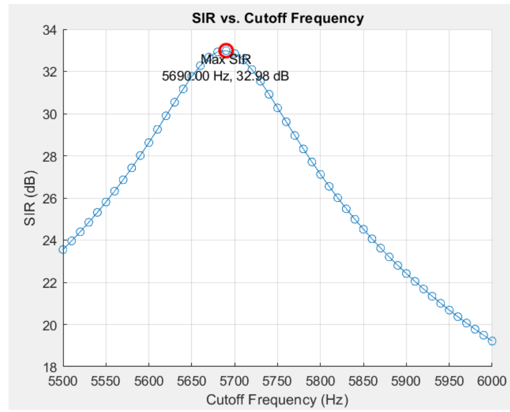
Song	**r**	**SIR (dB)**	**Poles' frequency (Hz)**
Song1	0.99	40.2	5580
Song1	0.95	32.98	5690
Song2	0.99	22.52	5550
Song2	0.95	16.29	5590

Table 3: Comparison of SIR values changing the module of the poles for both the songs

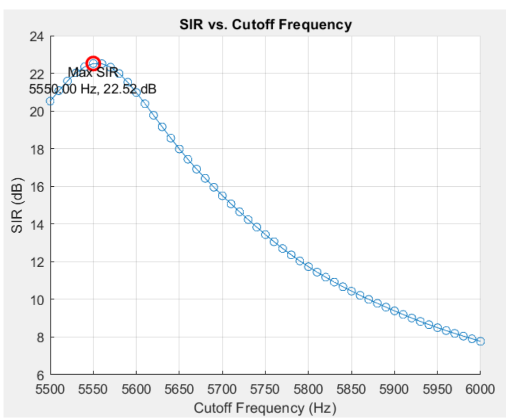
The module of the poles determines the steepness of the zero, in other words, the notch sharpness is directly proportional to the module of the poles. While the choice of the highest possible module for the poles may seem the best one, since we obtain a really high SIR, in practice that may not be the case, mainly because it can still causes some oscillations/to converge it requires a long time, which can cause numerical issues. Furthermore, if the filter is implemented on hardware, hence with limited capacity and/or precision, it can be quite difficult to mimic the exact behaviour, that is why, $r = 0.95$ is a good compromise, typically used in audio-specific applications. As in the design of the Bessel filter, a `for loop` was the best way to find the frequency that optimize the SIR value. The first thing one notices looking at figure 7 is that the SIR values obtained using a filter build with the specific intent of cleaning a particular disturbance are much higher than the one obtained by filtering using a generic low-pass filter (already optimized to have the best SIR possible). As in the previous studied case, also this time the SIR for Song1 is much higher (almost twice as much) than the one for Song2, which is still possible to hear without noticing the disturbance.



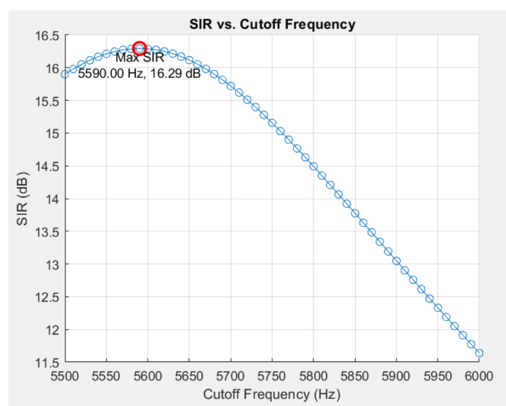
(a)



(b)



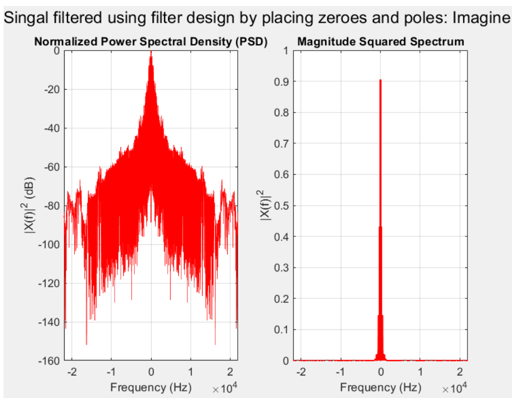
(c)



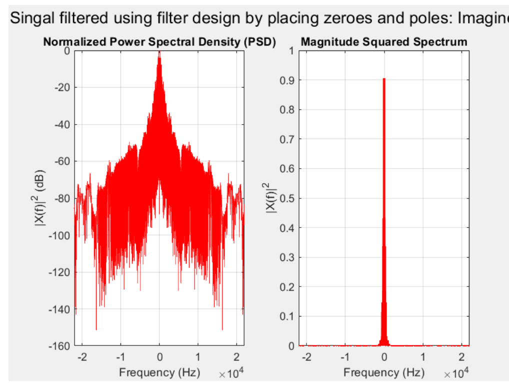
(d)

Figure 7: SIR optimization process by iterating over the poles' frequency for both song, using different pole modules: (a) Song1, $r = 0.99$, (b) Song1, $r = 0.95$, (c) Song2, $r = 0.99$,(d) Song2, $r = 0.95$

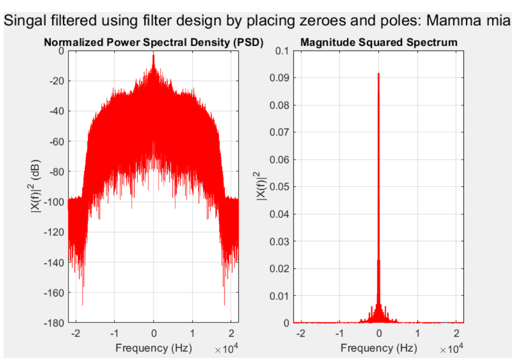
In figure 8 are shown the filtered signals in the frequency domain. In all the plots, the disturbance in no longer visible, which is a great result. By comparing the result obtained for the filter using as module for the poles 0.99 (on the left of figure 8) with the one obtained using $r = 0.95$ (on the right) some differences can be spotted when analysing the normalized PSD (not though by analysing the MSB, since the difference is negligible on the large scale). What is observable, is that, in figure 8b and 8d, some of the frequency in a small neighbourhood of the disturbance frequency are also attenuated, while this effect seems to not been observable in figure 8a and 8c. In figure 8d, this is particularly noticeable, because there are some frequencies (around $5kHz$ indeed), that have a stronger attenuation compared to the ones around them. This attenuation is not present in the reference signal (figure 2a). While in figure 8b this is not observable, one can still see some frequency that are more attenuated than the other.



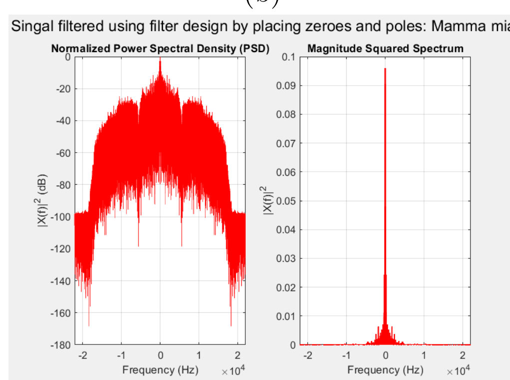
(a)



(b)



(c)



(d)

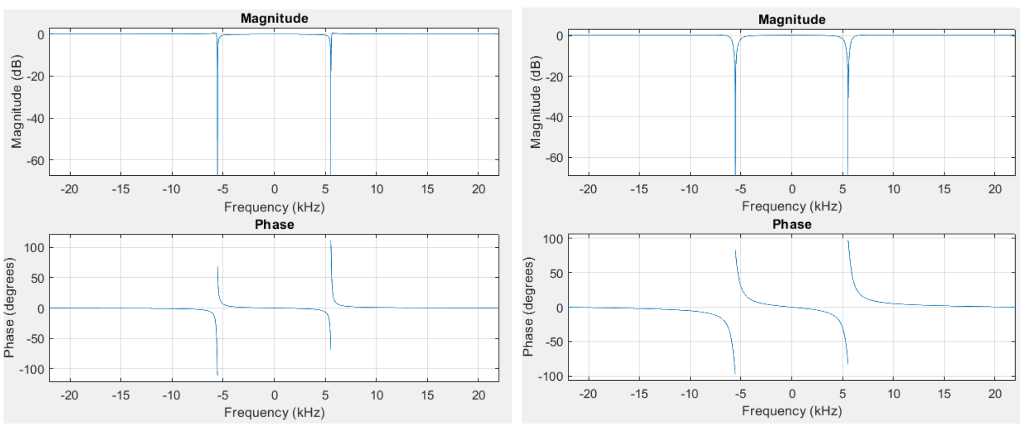
Figure 8: Filtered signals for different pole modules. (a) Song1, $r = 0.99$, (b) Song1, $r = 0.95$, (c) Song2, $r = 0.99$,(d) Song2, $r = 0.95$

The effect of poles and zeroes in the transfer function is mirrored:

- **zeroes**: cause attenuation (frequencies are weakened), in a 3D surface plot are represented as "valleys"

- **pole**: introduce gain (frequencies experience amplification), represented as "mountains" in 3D plots

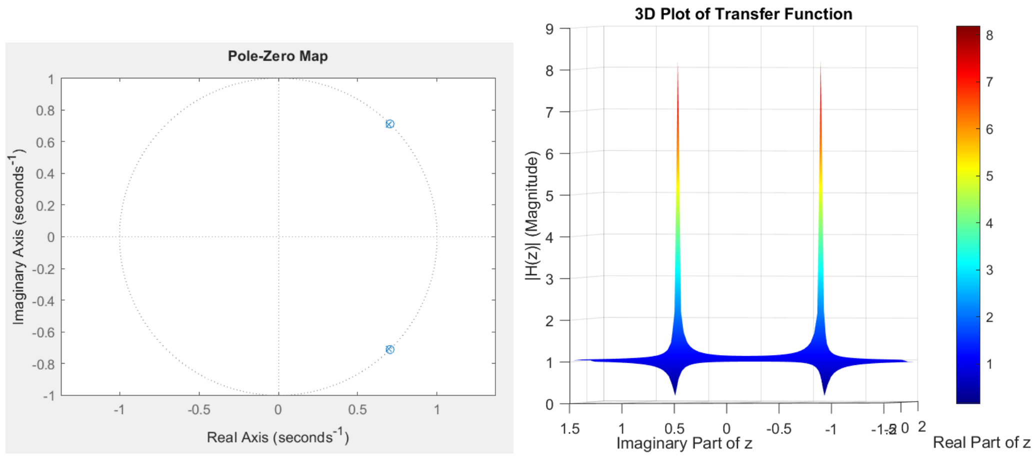
One may thing that placing the pole is irrelevant to filter the disturbance, in a sense this is right, to eliminate the noise, the poles does not have any effect. The poles are though important to limit the frequency that are attenuated. Figure 9 reports the graph for the frequency response of the designed filter, having different values for the module of the poles. While the general shape of the filter is the same, one can easily notice how on figure 9b the notch of the filter is wider than the notch in figure 9a.



(a) (b)

Figure 9: Pole-zero map of the transfer function. Circle represent the zero, cross represents the pole. Only the first quadrant is shown. Pole-zero map (a) for Song1 and $r = 0.99$, (b) for Song2 and $r = 0.95$

Figure 10 reports instead two other interesting plots: figura 10a reports on the unitary circle the poles and the zeroes (since the system is real, the focus is on the first quadrant, to have the complete figure in mind, one should also consider the complex conjugate of the reported poles/zeroes. The zero lies exactly on the unitary circle, thus completing eliminating the zero's frequency; the pole, here with a module of 0.99, is just a little bit translated to the left and to the inner part of the circle. The translation to the left means that the pole occurs at higher (but slightly) frequency that the one of the zero (in the case reported in the figure $f_{pole} = 5550Hz$). To better visualize the effect of placing the zero and the pole, in figure 10b it is reported the 3D surface plot, that highlights valleys and mountains.



(a) (b)

Figure 10: (a) Filter poles and zero for module $r = 0.99$, (b) 3D surface plot for the filter frequency response in the case of $r = 0.95$

5.3 Comparison of filtering done by different filter

Table 4 reports all the SIR values already discussed above plus a test case using a 4-th order Butterworth filter, known to have maximally flat response. In the frequency domain, since also the Butterworth filter would be a low-pass filter, the filter response would be very alike to Bessel filter (figure 5), with flatter passband. Also the output signal of the Butterworth looks like the Bessel output, since all the frequencies above the cut-off would be attenuated. Furthermore, exactly like in the Bessel, the disturbance is not eliminated in its totality (like for the custom IIR filter), but it is only attenuated.

Indeed, the Butterworth filter output a better song than the Bessel filter of the same order, yet not comparable to the specific IIR filter (at least with $r = 0.99$). Interesting, if comparing the Butterworth to the IIR filter with module $r = 0.95$, one would see that the Butterworth filter works better on Song1, while the custom IIR works better, and by far, on Song2. It now appears very clear that the best filter one can design is by directly placing zeroes and poles, to filter out only specific frequency. The problem of this type of filter, compared to the other, is to find an efficient way to create the filter on hardware.

Filter's type	**song**	**SIR (dB)**
Bessel	Song1	22.04
Bessel	Song2	7.26
Mine ($r = 0.99$)	Song1	40.20
Mine ($r = 0.99$)	Song2	22.52
Mine ($r = 0.95$)	Song1	32.98
Mine ($r = 0.95$)	Song2	16.29
Butterworth	Song1	29.94
Butterworth	Song2	8.19

Table 4: Recap of best SIR obtained for each type of filter tested

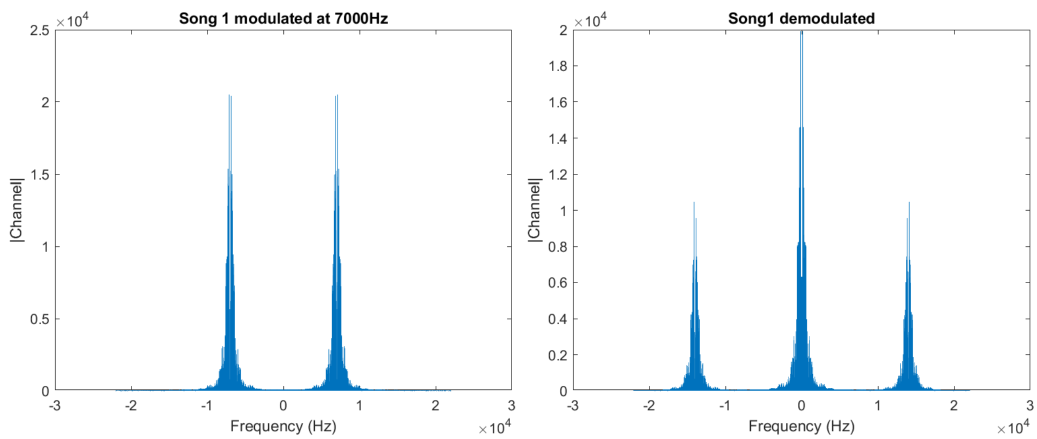
6 Task 2: share a channel and multiplexing

In a real communication system, when multiple signals are transmitted simultaneously, it is crucial to ensure that as little information as possible is lost, ideally none. To do so, modulation is implemented. In this paper amplitude modulation (AM) is employed, in particular, double-sideband amplitude modulation. This type of modulation differs from traditional AM because the carrier signal is not transmitted. If from one side this means that the efficiency of the modulator is increased (no power is lost by transmitting the carrier), the complexity of the receiver demodulator is also increased. To demodulate a signal, it is enough to multiply again by the carrier signal, hence the importance that the signal implemented at the transmitter and at the receiver matches perfectly, as any mismatch in frequency or phase between the transmitted and received carriers introduces interference, degrading the recovered signal. Equation 6 shows the general modulating signal, accounting for zero phase. The modulating signal in the frequency domain consists of two Dirac delta functions with height $\frac{A_c}{2}$.

$$c[n] = A_c \cdot cos(2\pi f_c n \Delta T) \overset{FT}{\longleftrightarrow} C(f) = \frac{A_c}{2}(\delta(f - f_c) + \delta(f + f_c)) \quad (6)$$

It is important to always have in mind what happens in the frequency domain. Modulating is a multiplication in the time domain, thus convolution in the frequency one. By convolution's properties, convolving with a Dirac delta function means shifting the signal by the argument of the delta. Equation 7, as well as figure 11a, shows the result of this operation. During modulation it is possible that some parts of one of the copy of the signal superimposes to the others, causing aliasing, and thus signal deterioration.

$$x_{mod}[n] = x_{in}[n] \cdot c[n] \overset{FT}{\longleftrightarrow} X_{mod}(f) = \frac{A_c}{2}(X_{in}(f - f_c) + X_{in}(f + f_c)) \quad (7)$$



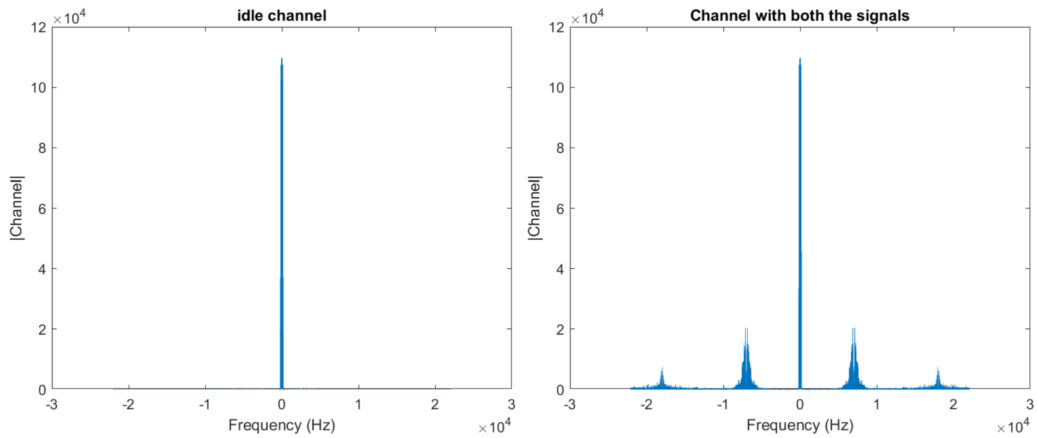
(a) (b)

Figure 11: Example of modulation and demodulation, using as carrier: $c[n] = 1 \cdot cos(2\pi \cdot 7000Hz \cdot n \cdot \Delta T)$. (a) modulation, (b) demodulation

As briefly introduced before, the demodulator implemented in our simulation, is identical to the modulator block, hence it is the time multiplication with a sinusoid carrier. Equation 8 reports the mathematical calculation required to obtain the demodulated version of the signal. Analysing the formula one notices that the demodulated signal, in particular the one at low frequencies that is going to be filtered out to obtain the transmitted signal at the receiver side, has an amplitude which is $\frac{A_c}{2}$ of the original. Hence it is necessary to multiply the recovered signal by a factor $\frac{2}{A_c}$ to obtain the original amplitude of the signal. Figure 11b shows graphically the demodulated signal (with no amplitude correction factor).

$$x_{demod}[n] = x_{mod}[n] \cdot c[n] \overset{FT}{\longleftrightarrow} X_{demod}(f) = \frac{A_c}{4}(X_{in}(f-2f_c)+X_{in}(f+2f_c))+\frac{A_c}{2}X_{in}(f) \quad (8)$$

In figure 12a it is reported the channel as given, which is with its baseband occupied by a signal (noise). The bandwidth of the channel is $\frac{Fs}{2} = 22.05kHz$. One can notice from the figure that the magnitude of the signal that is already transmitted on the channel is very high. Figure 12b shows how the signal looks like when both the signals are transmitted.



(a) Idle channel

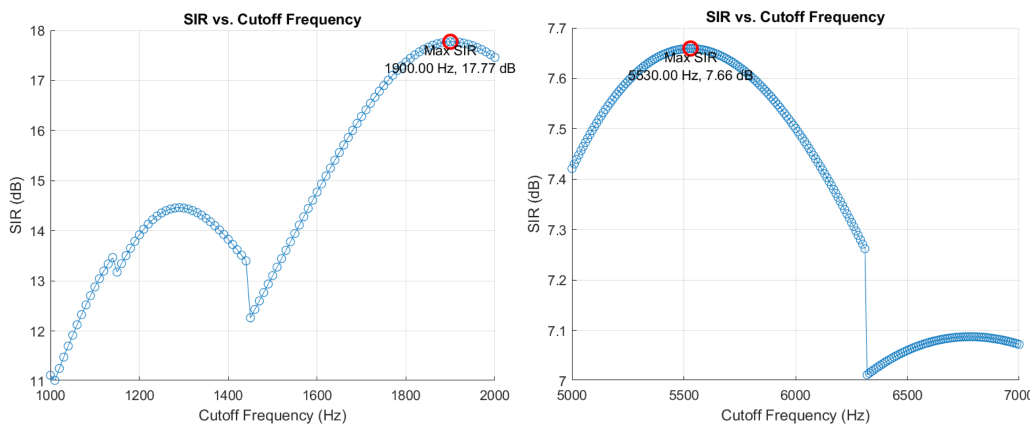
(b) Channel with modulated signals

Figure 12

6.1 Receive with 4-th order Bessel low-pass filter

6.1.1 When transmitting both the signals

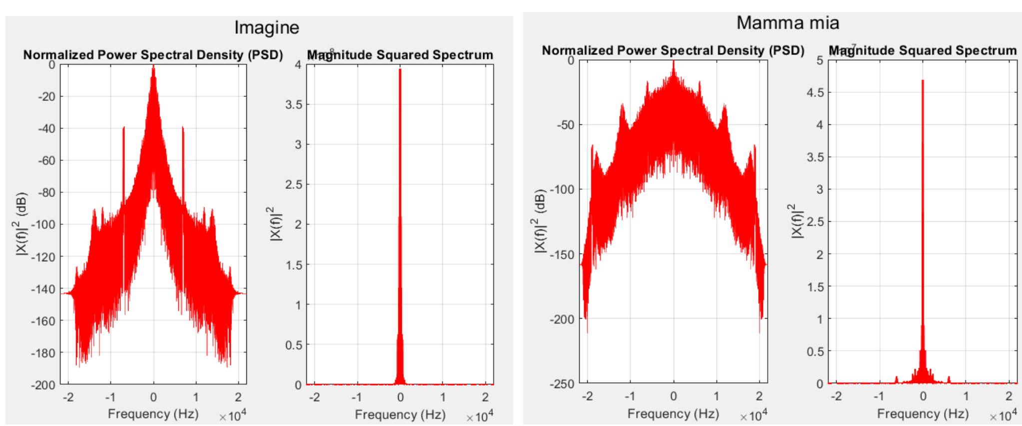
The procedure followed here is much the same as the one described in paragraph 5.1. In figure 13 it is possible to see the trends of the SIR for both songs for different cut-off frequencies. The first thing important to notice is that the range of allowed cut-off frequency is quite different for the two different songs. Comparing the SIR obtained here with what obtained, for example, in figure 3 one immediately notices that, at some frequencies, the SIR value changes a lot with respect to the previous value. Again, as it has always been the case, the SIR for Song1 is higher than the one obtained for Song2.



(a) (b)

Figure 13: Optimization process (best SIR) for filtering demodulated signal using 4-th order Bessel low pass filter. (a) SIR for Song1, (b) SIR for Song2

Figure 14 reports the spectra of both the filtered songs. By a simple analysis it is possible (mostly in the plot using decibels) to see how the other song transmitted on the channel changes the recovered signal. This result in having noises in the final output, Depending on the song, the noise added is at different frequencies (it depends on the carrier it is multiplied).



(a) (b)

Figure 14: Best recovered signal after demodulation. (a) Song1, (b) Song2

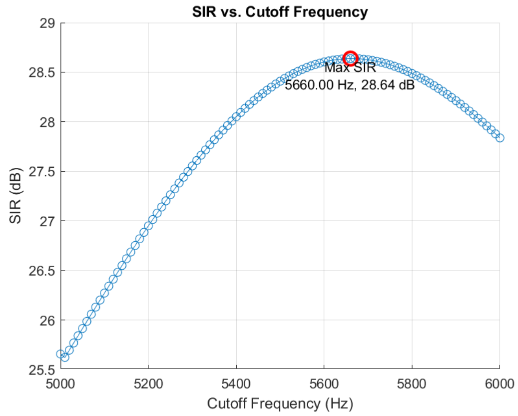
As shown in table 5, changing the carrier frequency also affects the SIR values and the optimal cut-off frequency of the filter. This occurs because if the songs are modulated too close to each other or the existing noise at zero frequency, the interference during demodulation increases. It is possible to notice that the SIR for Song1 changes much faster for Song2 and that when the SIR for one signal increases, the SIR of the other decreases (not necessary by the same amount).

f_{c1} (Hz)	f_{c2} (Hz)	**SIR1** (dB)	**Cut-off1** (Hz)	**SIR2** (dB)	**Cut-off2** (Hz)
7000	19000	17.77	1900	7.66	5350
6000	19000	16.48	1850	7.67	5350
7000	20000	17.97	1970	5.77	5330
8000	20000	19.18	2270	5.73	5310

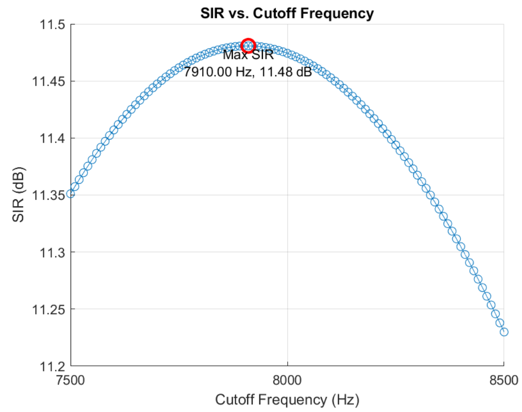
Table 5: SIR and cut-off frequency (for Bessel filter) changing the carrier frequency of both the song on the same channel

6.1.2 When transmitting a signal at a time

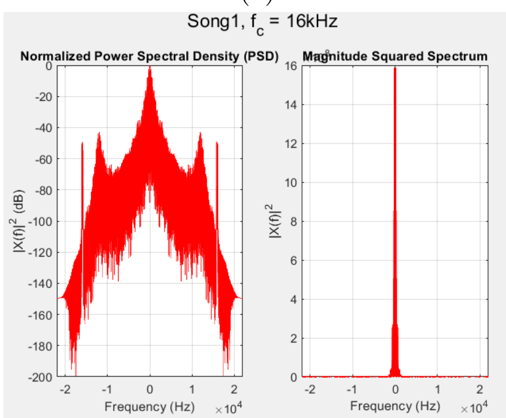
If instead of sharing the channel, the signals are transmitted one at a time, it is possible to have a higher overall SIR. As expected, the best carrier frequency is the same for both songs: after some iteration, it was found that the best SIR is obtained when $f_c = 16kHz$. At a first fast analysis one may think that the best carrier frequency would be in the middle of the bandwidth, around $11.025kHz$. This is not case because at low frequencies there is another signal, while the channel is empty for all other frequencies. Thus, the best carrier frequency is shifted towards higher frequency; on the other hand, if it too high, some information may get lost since the channel has limited bandwidth. Comparing the SIR obtained in this simulation (reported in table 6) with the analysed above (table 5), immediately it is clear that in this case the SIR are much higher, hence a signal with a better quality is received. Optimization process and best recovered signal are shown in figure 15.



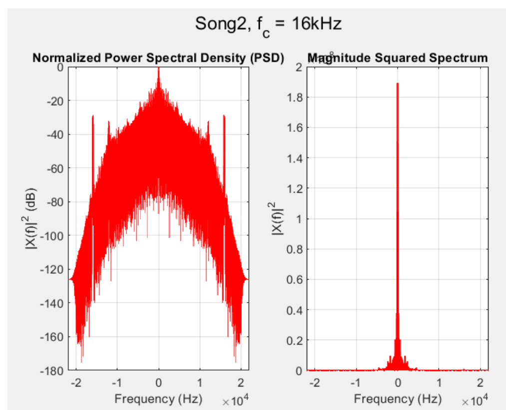
(a)



(b)



(c)



(d)

Figure 15: In sub-figure (a), (b) optimization process, based on SIR values to find the best cut-off frequency for a 4-th order Bessel LPF, respectively for Song1 and Song2. Filtered signal in the frequency domain, (c) Song1, (d) Song2

song	**cut-off frequency (Hz)**	**SIR (dB)**
Song1	5650	28.64
Song2	7910	11.48

Table 6: SIR and cut-off frequency when transmitting only one song on the channel, having a carrier with frequency $f_c = 16kHz$

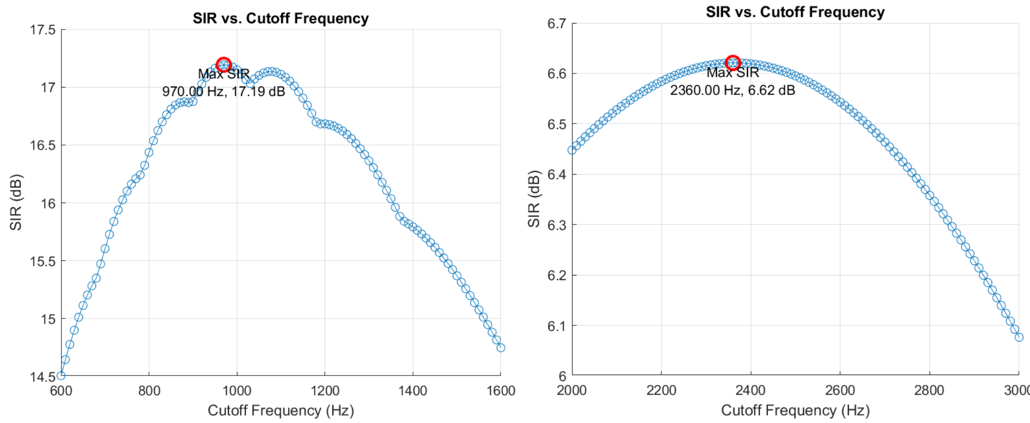
6.2 Receive with my own filter

As seen before, using a filter designed with the specific purpose of removing a specific disturbance instead of Bessel filters results in better outcomes. Here an analysis on what happens when instead of a sinusoidal disturbance a noise, which is spread on multiple frequencies, is to be filtered. To define a low-pass filter by placing only a pair of poles and zeros (complex conjugate), the best way is to iterate on both the zeroes and the poles frequency. The most important factor is the frequency of the zeroes, since this is the factor that determines the attenuation, while the module and the frequency of the poles are useful to define the sharpness of the transition band.

6.2.1 When transmitting both the signals

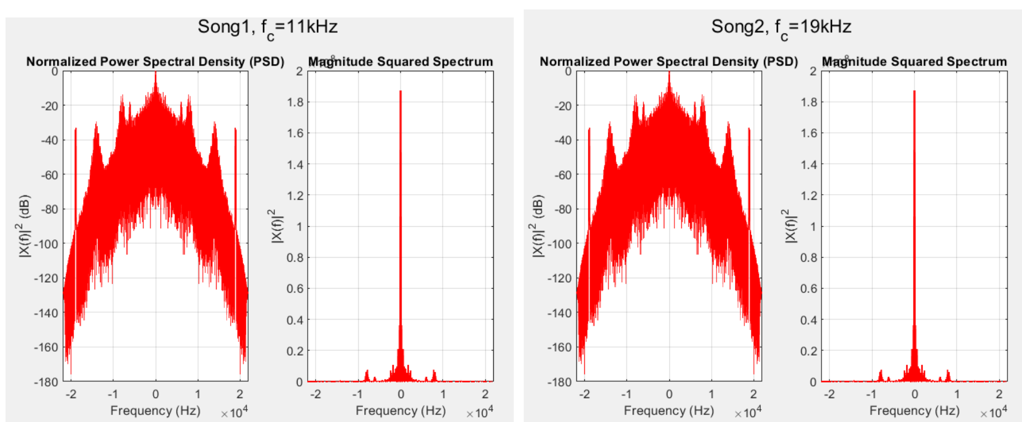
To obtain a low pass filter by working with zeroes and poles, the best technique is to place the zeroes at -1, which indeed represents the high frequency on the unitary circle. For this purpose, the choice of the pole module becomes really important: a too high module may create a ripple (with a non negligible amplitude), thus amplifying non desirable frequencies, a too low module may result in a not at all sharp transition, hence resulting in non attenuating enough the copies.

Opposed to what find out when filtering using the Bessel filter, in this case the maximum SIR (for both the songs) is obtained for higher carrier frequencies: if earlier $f_{c1} = 7kHz$ and $f_{c2} = 19kHz$ were the frequencies for which the maximum SIR was obtained, in this case $f_{c1} = 11kHz$ and $f_{c2} = 19kHz$ work the best. That is because by employing carrier at higher frequency the copies that are created when demultiplexing are at higher frequency, so it is possible to attenuate them better. Following the usual optimization process, SIR values (figure 16), filtered songs (figure 17). Analysing the last figure, it is clear that the signal is affected by multiple disturbance: if the song is listened, background noises may be heard.



(a) (b)

Figure 16: SIR values for different poles' frequency: (a) Song1, $r = 0.9$, (b) Song2, $r = 0.75$

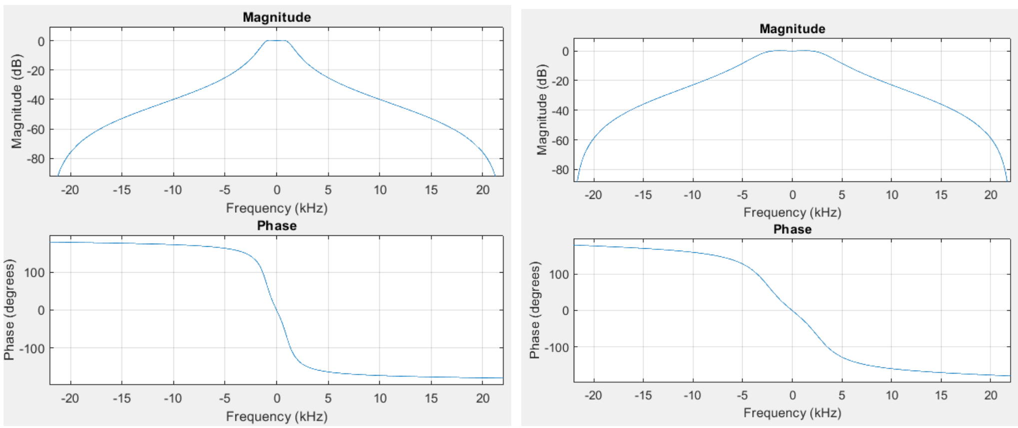


(a) (b)

Figure 17: Filtered songs: (a) Song1 and (b) Song2

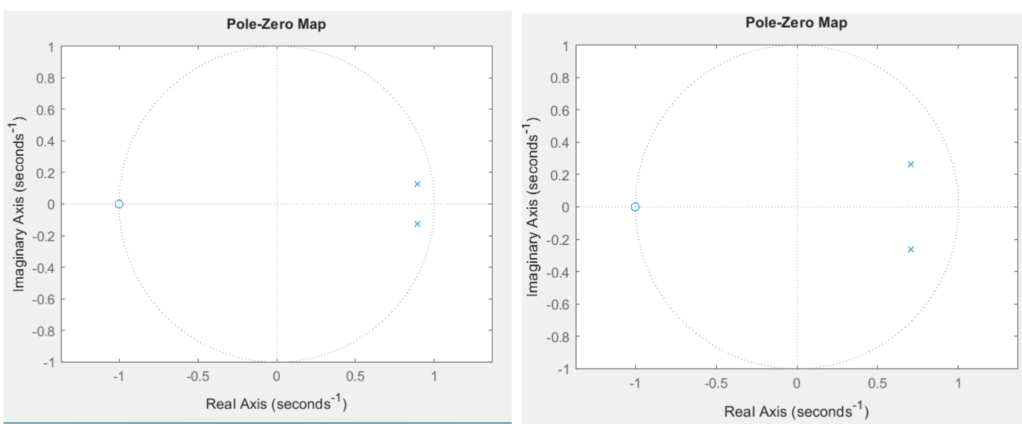
Of particular interest are the frequency response of the filter, reported in figure 18. The overall shape is the same for both the filter (almost flat passband at low frequencies and attenuation at higher frequency, but the similarities ends there: for Song1 the passband is very narrow, for Song2 it is wider; transition in Song's filter are sharper than in Song2's filter. This last fact is caused by the different pole module, 0.90 and 0.75 respectively.

In figure 19 it is possible to appreciate on the unitary circle the position of poles and zeros for both the filters.



(a) (b)

Figure 18: Frequency response of the filter, for (a) Song1 and (b) Song2



(a) (b)

Figure 19: Frequency response of the filter, for (a) Song1 and (b) Song2

Table 7 reports some of the SIR values found when testing for the best carrier frequencies, assuming the module of the pole fixed: $r_1 = 0.9$, $r_2 = 0.75$.

f_{c1} (kHz)	f_{c2} (Hz)	**SIR1** (dB)	f_{pole1} (kHz)	**SIR2** (dB)	f_{pole2} (Hz)
10	19	13.7	600	7.13	2600
12	19	17.15	1000	6.62	2360
12	18	16.82	1000	6.42	2200
12	20	16.89	970	7.03	2510

Table 7: Comparison between SIR obtained for different carrier frequencies

6.2.2 When transmitting a signal at a time

The best SIR values were found, this time, for a carrier frequency of $15kHz$. The reason for this becomes pretty evident when analysing figure 20: by selecting this specific carrier frequency, when demodulating, the noise, occupying the baseband in the original channel and the copies of the actual transmitted signal are almost at the same frequency (around $15kHz$). Once that has been determine, it is possible to set the zero to a specific location and then iterate to find the best frequency for the pole.

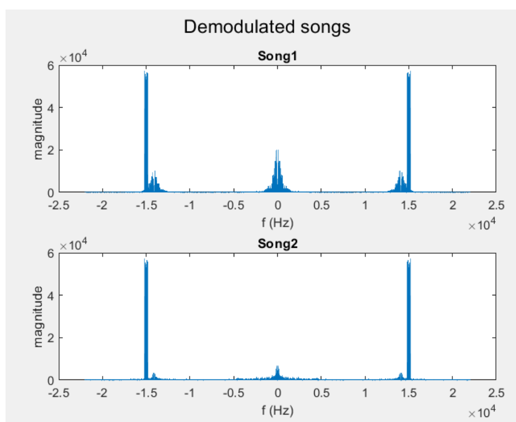
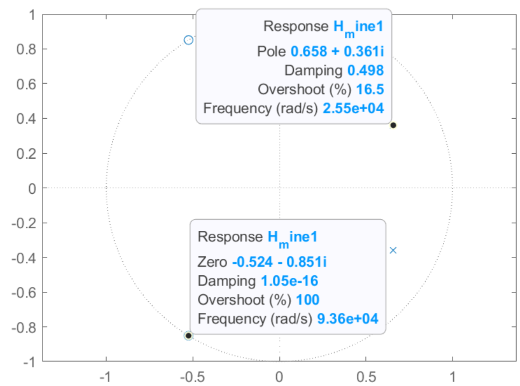
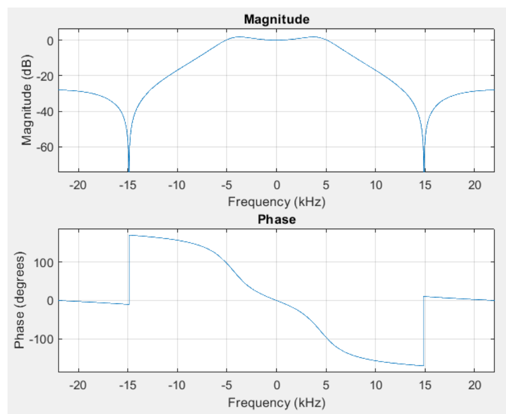


Figure 20: Demodulated songs: $f_c = 15kHz$

As opposed to the case when it was required to eliminate a specific frequency, here the aim was to construct a low pass filter. In my implementation, the frequency response (figure 21b) of the system has a different shape than the traditional low-pass filter. In the passband the filter is almost completely flat, with a minimal ripple due to the poles magnitude ($r = 0.75$ is the best trade-off between sharpness of the notch and amplitude of the ripple). At very high frequency the signal is attenuated, independently from the actual frequency, by $30dB$. The maximum attenuation, is at $15kHz$, indeed the frequency where the disturbances are located once the signal is demodulated. Figure 21a shows the location of the zeroes and of the poles inside the unitary circle: the zero, again, lies exactly on the circle, while the poles are inside.

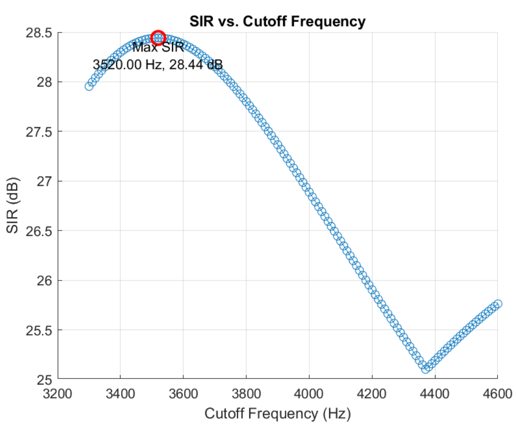


(a) Position of poles and zeros in the unitary circle

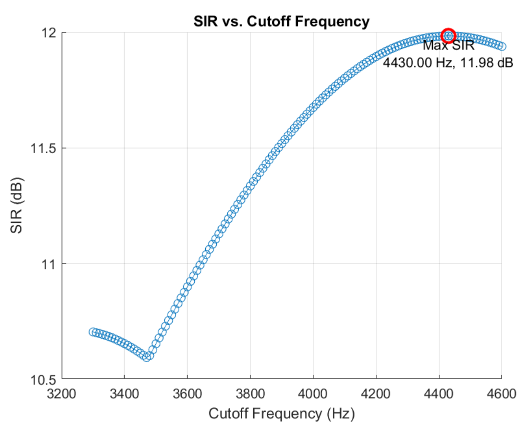


(b) Frequency response of the filter

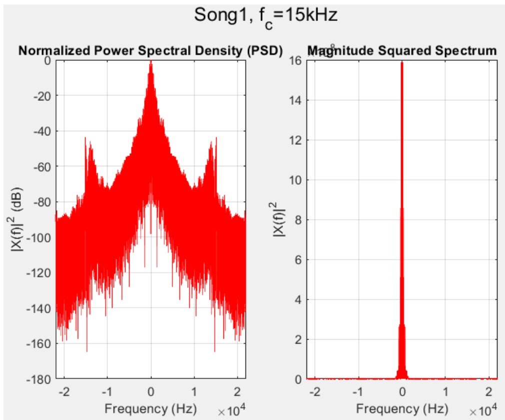
Figure 21



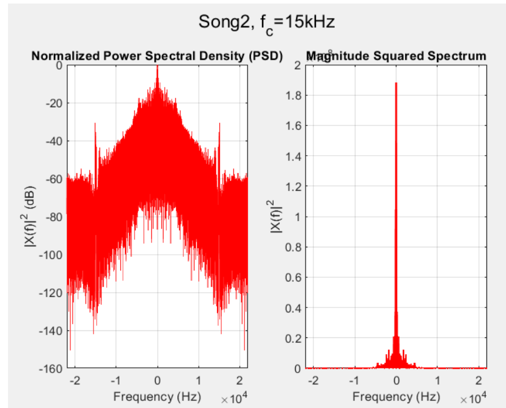
(a)



(b)



(c)



(d)

Figure 22: In sub-figure (a), (b) optimization process, based on SIR values to find the best poles' frequency for my own LPF, respectively for Song1 and Song2. Filtered signal in the frequency domain, (c) Song1, (d) Song2

7 Conclusions

This project consisted in discovering how different filtering techniques influences the quality (in terms of SIR) of the output signal, considering different types of noise to be cleaned. Throughout all the project a comparison between 4-th order Bessel low-pass filter and second order filter designed by directly placing zeroes and poles has been carried out, highlighting the benefit and drawbacks of each design.

In the first part of the project the filters were used to remove a sinusoidal disturbance from two different songs. This showed that, even though the disturbance was identical for both the songs, due to different spectral characteristic between the two songs, optimal cut-off frequency (in the case of Bessel filter) or optimal poles' frequency (in the case of mine design), differs a little for the two songs. Another observation was how, by designing a specific filter for this type of disturbance, it was possible to obtain signals with a very high SIR, where it was almost impossible to spot the disturbance, while, using a general low-pass filter, since all the frequencies above the cut-off are attenuated, the SIR is always lower.

In the second part, filtering was necessary to clean the signal after modulation. Given a channel, with its baseband frequencies occupied by some noise, it was required to transmit the songs on the channel itself, both simultaneously and only one song at a time. Double sided amplitude modulation was used, to maximize the power of the song received; different carrier frequencies were tested to minimize the error due to aliasing between high frequency components of the different songs. Since it was not possible to remove a specific frequency, in this case the usage of both a Bessel filter or a filter designed by me produced a similar output.