Digital Signal Processing

Lab 3 report

Marco Casagrande – Julius Rauscher | IE6-DSP | 25.4.17

Contents

[Attachment A 2](#_Toc481848645)

[Calculations 2](#_Toc481848646)

[Attachment B 3](#_Toc481848647)

[Input sequence 3](#_Toc481848648)

[Calulation results 3](#_Toc481848649)

[MATLAB results 4](#_Toc481848650)

[MATLAB script 4](#_Toc481848651)

[Comments 5](#_Toc481848652)

[Attachment C 5](#_Toc481848653)

[Input sequence 5](#_Toc481848654)

[Calulation results 5](#_Toc481848655)

[MATLAB results 5](#_Toc481848656)

[MATLAB script 6](#_Toc481848657)

[Comments 7](#_Toc481848658)

[Question a) 7](#_Toc481848659)

[Question b) 7](#_Toc481848660)

[Question c) 7](#_Toc481848661)

[Attachment D 8](#_Toc481848662)

[Input sequence 8](#_Toc481848663)

[DSK6713 output - overflow 8](#_Toc481848664)

[Butterfly rescaling implementation 8](#_Toc481848665)

[DSK6713 output – no overflow 9](#_Toc481848666)

[Attachment E 10](#_Toc481848667)

[Attachment F 12](#_Toc481848668)

[Attachment G 19](#_Toc481848669)

# Attachment A

Using the script provided in the lecture, the coefficients of a Low-Pass FIR filter are calculated using the Parks-McClellan algorithm, which implements an equiripple linear phase filter using the minimum number of coefficients.

The specifics of the filter are as follows:

* passband edge frequency: 620 Hz
* stopband edge frequency: 1020 Hz
* maximum passband ripple: 0.01
* minimum stopband attenuation: 40 dB
* sampling frequency: 8 kHz

The characteristic of the resulting filter is shown in Figure 1:

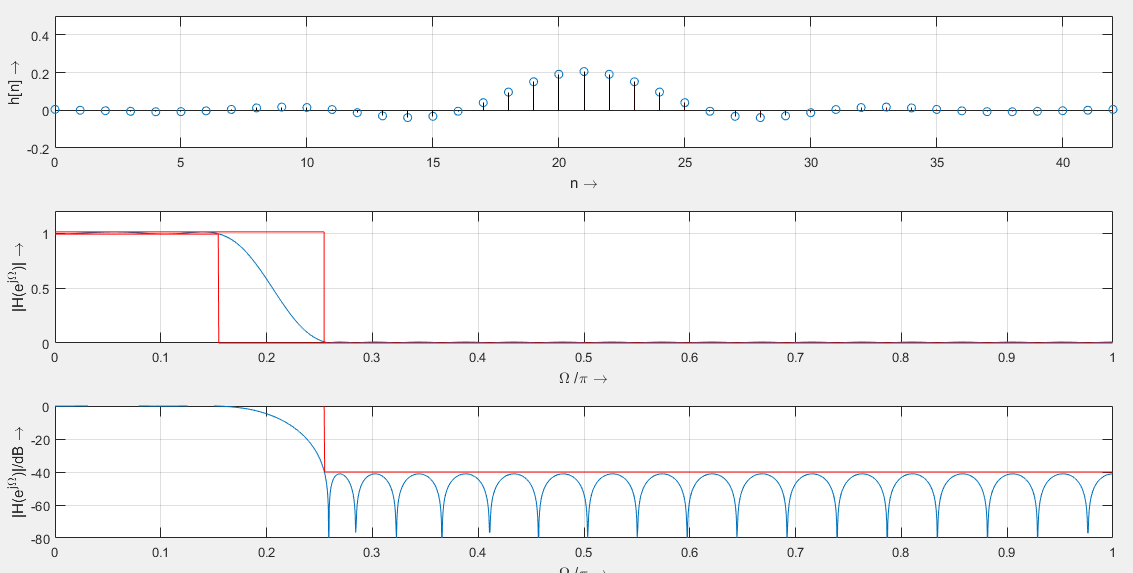


Figure 1 - Impulse response and amplitude response of the equiripple FIR Low-Pass filter

In the normalized amplitude response we can clearly appreciate the equiripple characteristic behavior, as well as the expected 40dB minimum attenuation in the stopband, which occours at fSB = FS/2 \* ΩSB/π = 1020 Hz. The other requirments are also satisfied.

Though originally the algorithm would implement the filter using only 39 coefficients, we force their number N = 43. This is both to compensate for the estimation error, which apparently is a feature of the MATLAB implementation of the algorithm and to achieve an equal number of coefficients to the respective High-Pass filter, implemented using the same tolerances and edge frequencies and listed in Attachment B.

# Attachment B

This time an equiripple FIR High-Pass filter is implemented:

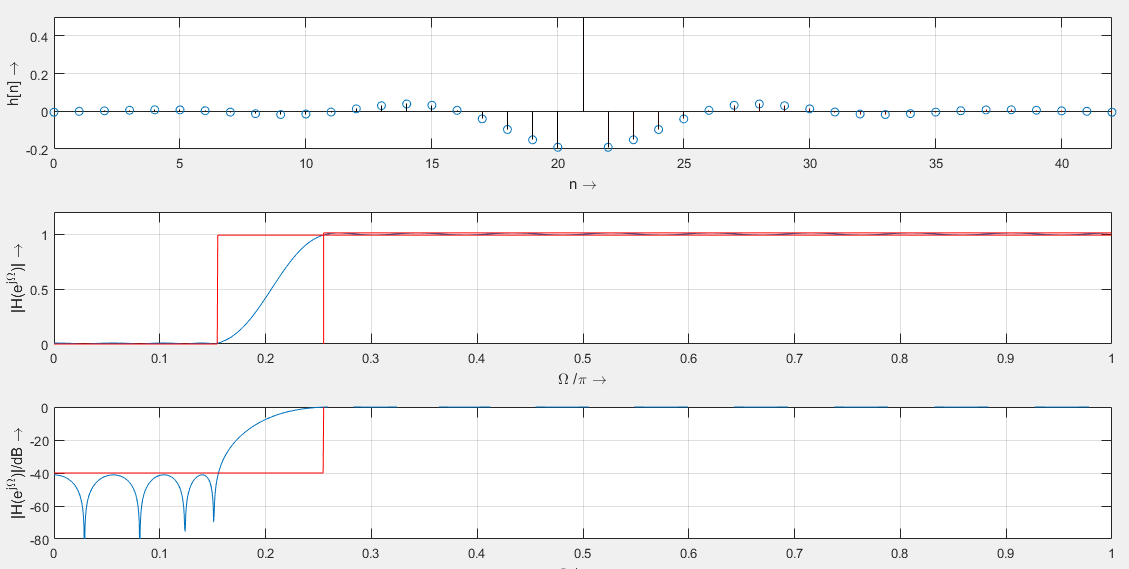


Figure 2 - Impulse response and amplitude response of the equiripple FIR High-Pass filter

# Attachment C

The coefficients of the filters, which compose their impulse response, are as follows:

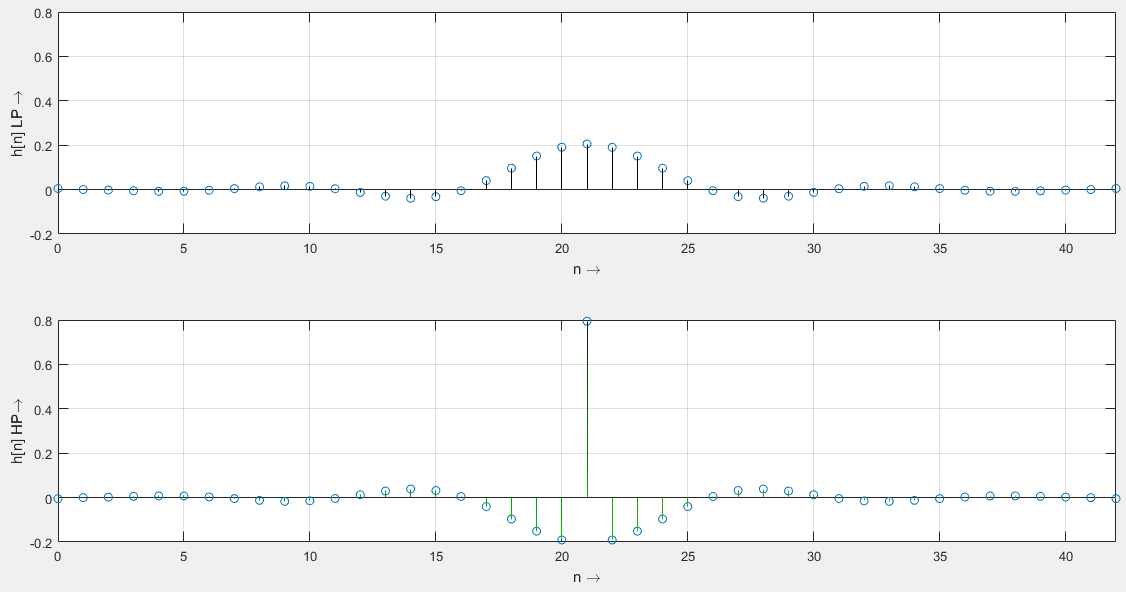


Figure 3 - Comparison between the equiripple, linear-phase Low- and High-Pass Filter impulse responses

|  |  |  |
| --- | --- | --- |
| n | LP h[n] | HP h[n] |
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| 3 | -0.01 | 0.01 |
| 4 | -0.01 | 0.01 |
| 5 | -0.01 | 0.01 |
| 6 | 0 | 0 |
| 7 | 0 | 0 |
| 8 | 0.01 | -0.01 |
| 9 | 0.02 | -0.02 |
| 10 | 0.01 | -0.01 |
| 11 | 0 | 0 |
| 12 | -0.01 | 0.01 |
| 13 | -0.03 | 0.03 |
| 14 | -0.04 | 0.04 |
| 15 | -0.03 | 0.03 |
| 16 | -0.01 | 0.01 |
| 17 | 0.04 | -0.04 |
| 18 | 0.1 | -0.1 |
| 19 | 0.15 | -0.15 |
| 20 | 0.19 | -0.19 |
| 21 ((N-1)/2 , FS/2) | 0.21 | 0.79 |
| 22 | 0.19 | -0.19 |
| 23 | 0.15 | -0.15 |
| 24 | 0.1 | -0.1 |
| 25 | 0.04 | -0.04 |
| 26 | -0.01 | 0.01 |
| 27 | -0.03 | 0.03 |
| 28 | -0.04 | 0.04 |
| 29 | -0.03 | 0.03 |
| 30 | -0.01 | 0.01 |
| 31 | 0 | 0 |
| 32 | 0.01 | -0.01 |
| 33 | 0.02 | -0.02 |
| 34 | 0.01 | -0.01 |
| 35 | 0 | 0 |
| 36 | 0 | 0 |
| 37 | -0.01 | 0.01 |
| 38 | -0.01 | 0.01 |
| 39 | -0.01 | 0.01 |
| 40 | 0 | 0 |
| 41 | 0 | 0 |
| 42 | 0 | 0 |

Table 1 - Comparison between the equiripple, linear-phase Low- and High-Pass Filter normalized rounded coefficients

In Figure 3 is shown that both impulse responses are even-symmetrical with respect to the middle n = 21 coefficient. Furthermore, in Table 1 it is clear that the two filter coefficients are reciprocal for any n ≠ (N-1)/2, which determine the filter high-frequency response.

# Attachment L

The transfer characteristic of the DSK-board is obtained by executing a frequency sweep on the spectrum analyzer while the board samples the input signal and outputs it back again at a sampling frequency of FS = 8 kHz.

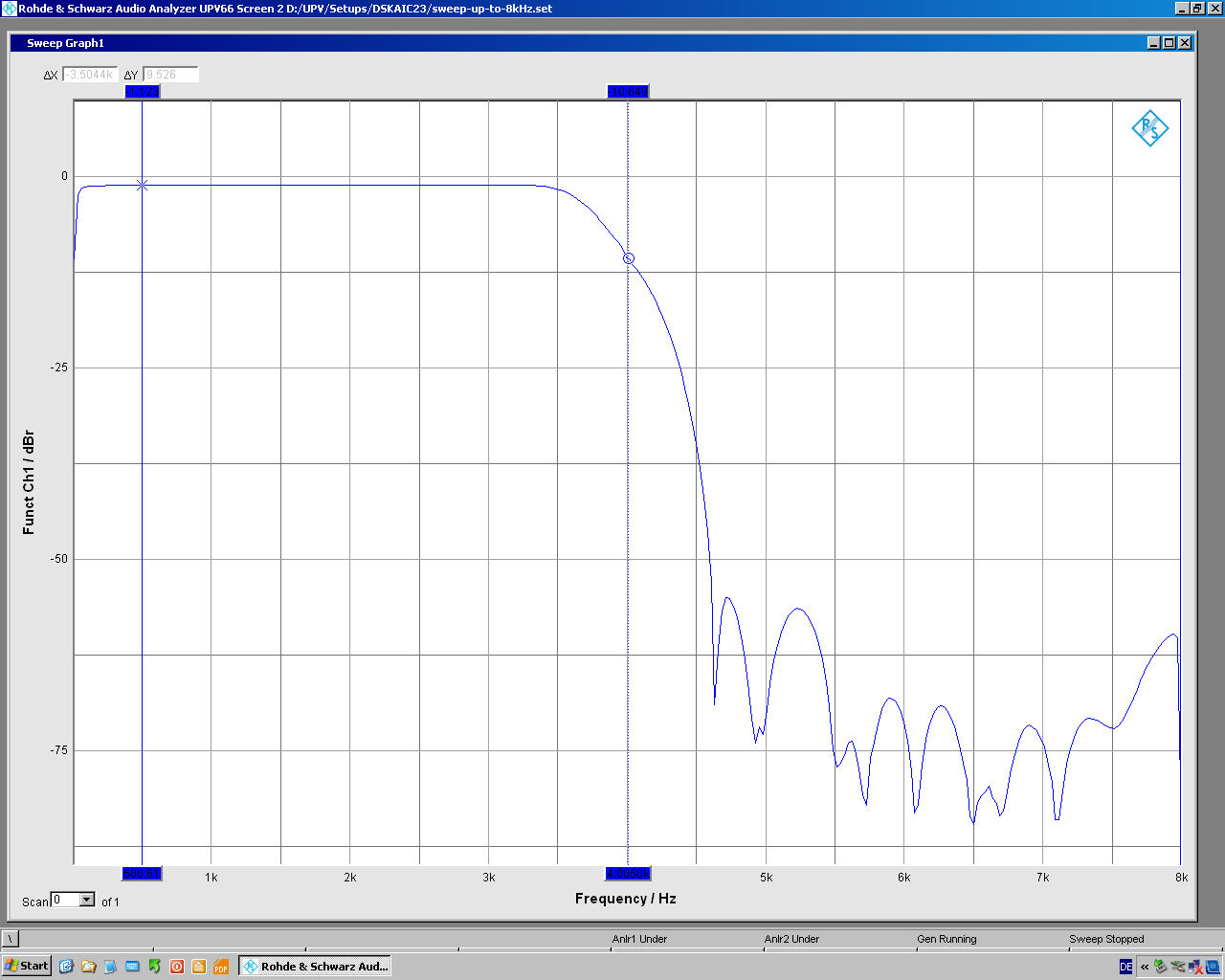


Figure 4 - Transfer characteristic of the DSK board with FS = 8kHz

From Figure 4 we can conclude that the DSK board implements a Low-Pass anti-aliasing filter that attenuates the frequency components f > FS/2.

# Attachment M

Implementing the filter on the DSK board requires two array. One holds the input samples which get shifted by one position each time a new sample is inserted and one that contains the coefficient obtained in MATLAB, adjusted and rounded to the short int 16 bit range. Every time the filtered output is sent to the DAC, the script sums N 32 bits products between the input and the respective coefficients. By storing the not approximated partial sum in a 32 bit int variable and carrying out the conversion to a 16 bit short int value at the end, the noise/accuracy ratio is maximized. If instead the partial sum variable would have been a 16 bit short int variable and each 32 bit product would have been converted to 16 bits before the summation, the conversion error of each product would have been summed as well, degrading the performance of the filter.

# Attachment N

The amplitude response of the Low-Pass filter implemented on the board, obtained by sweeping the frequency on the Spectrum Analyzer while the filter script is running, is as follows:

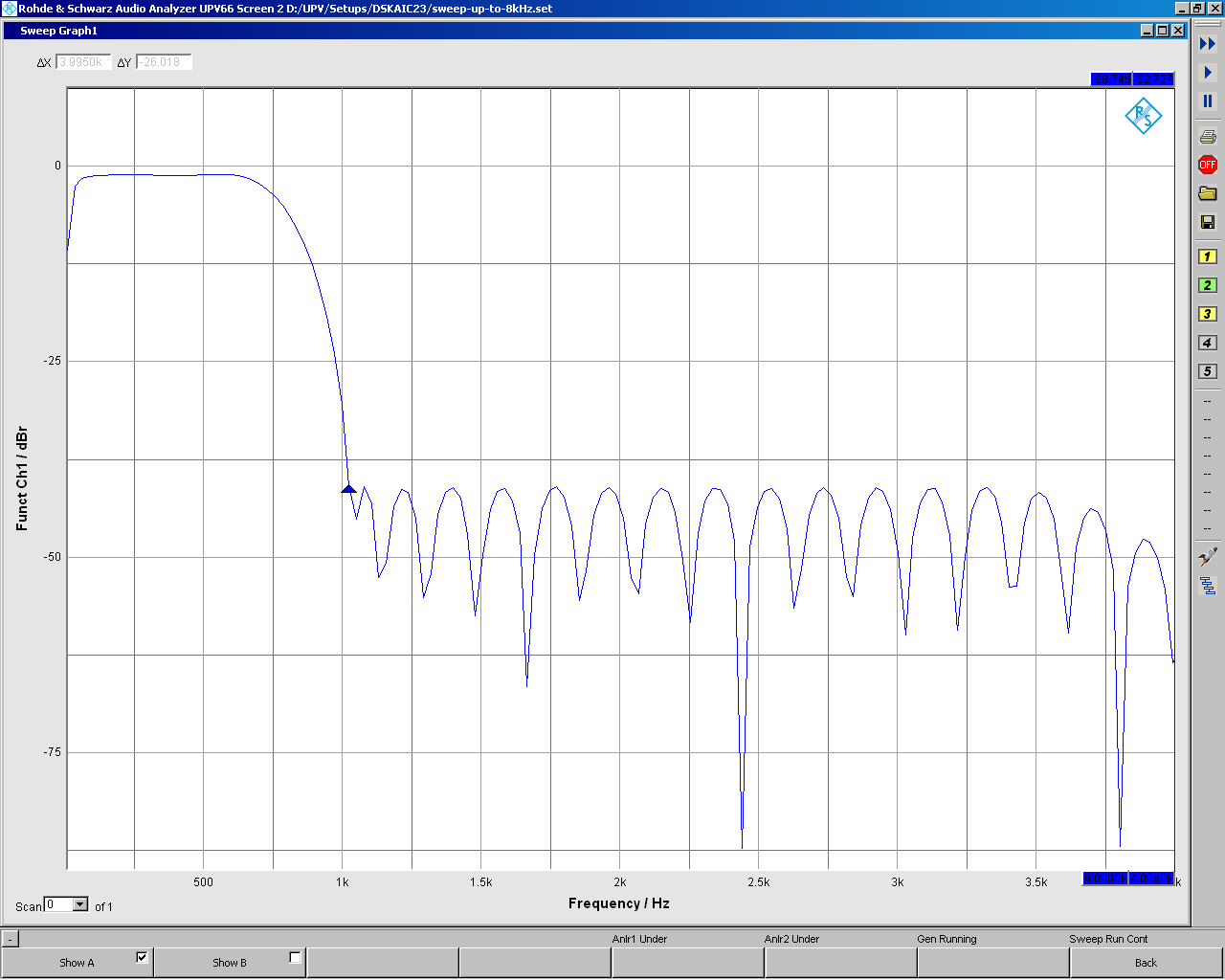


Figure 5 - Low-Pass Filter implementation on the DSK board impulse response

In Figure 5 we can observe that the Amplitude response is as expected according to the MATLAB simulation for any frequency f < 3,5 kHz, where the expected equiripple characteristic no longer applies, due to the added attenuation of the anti-aliasing filter on the DSK board.

# Attachment O

The High-Pass filter coefficients calculated in MATLAB are implemented. The amplitude response is as follows:



Figure 6 - High-Pass Filter implementation on the DSK board impulse response

The same considerations expressed in Attachment N also apply for the high filter case, and the added attenuation from the anti-aliasing DSK board filter are even more evident, as frequency components f > 3,5 kHz which should ideally have an attenuation of 0dB are instead increasingly damped.

To obtain the High-Pass filter output signal from the respective Low-Pass implementation, an approach would be the following:

yHP[n] = -yLP[n] + (32767 - 2 hLP[(N-1)/2]) u[(N-1)/2]

where yLP / yHP indicated the Low-/ High-Pass filter output, u is the input signal and hLP[(N-1)/2] is the coefficient corresponding to FS/2, obtained by inputting u[n] = δ[n – (N-1)/2] once.