Digital Signal Processing, Lab 5

Krzysztof Rudnicki
Student number: 307585

January 13, 2022

Chapter 1

Task 1

1.1 Task 1 code

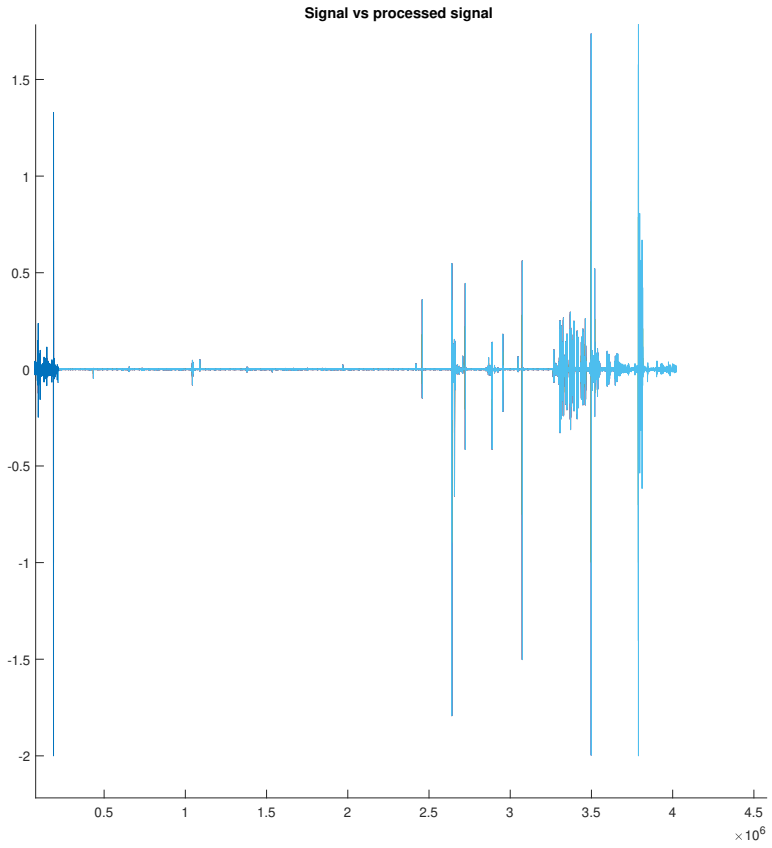
1.2 Experiment

1.2.1 Estimation of delay

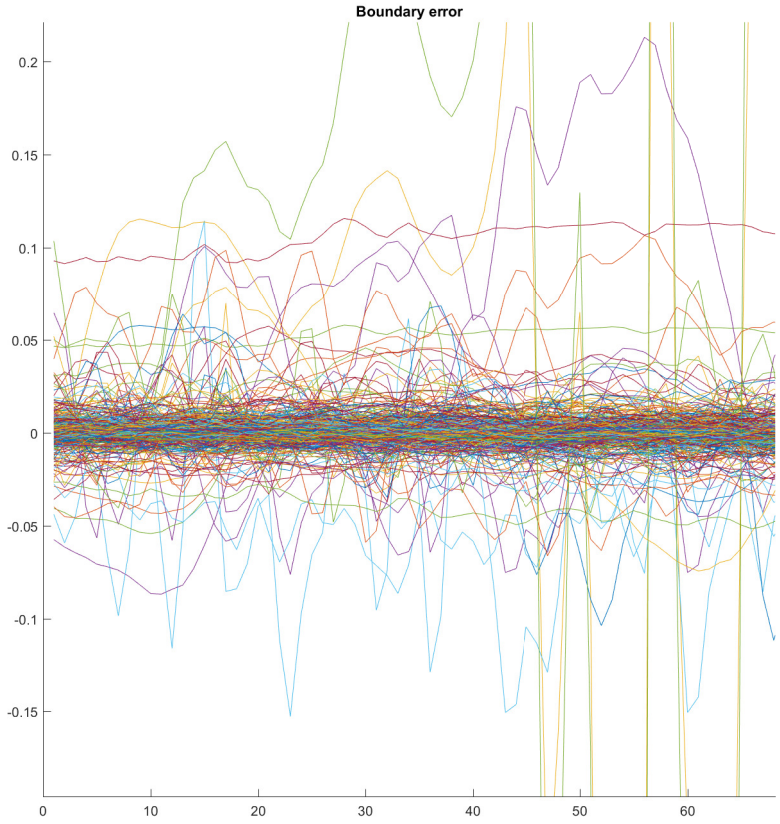
Block size	Delay [s]
1024	0.3
2048	0.4
4096	0.5
8192	0.72
16384	1

1.2.2 Graphs for 1024 block size

Signal and processed signal on one graph



Visible boundary error



1.2.3 How the block size influences delay

The bigger the block size, the longer the delay, for block sizes 4096 and lower the difference was almost indiscernible for human ear.
As we can see signals at the boundary with no filter are more or less random, we will try to change it with filters.

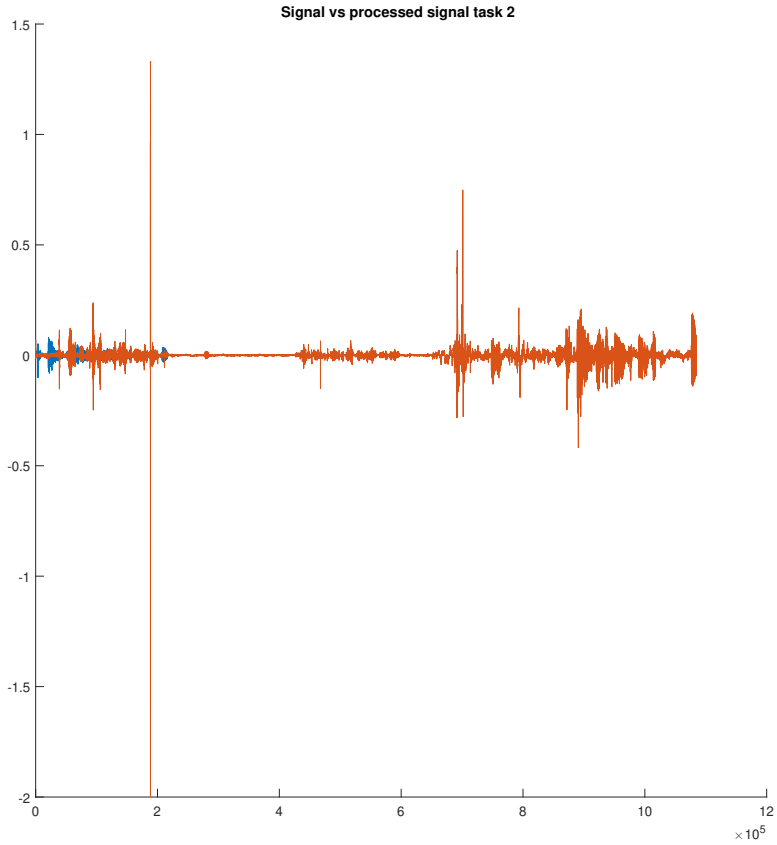
Chapter 2

Task 2

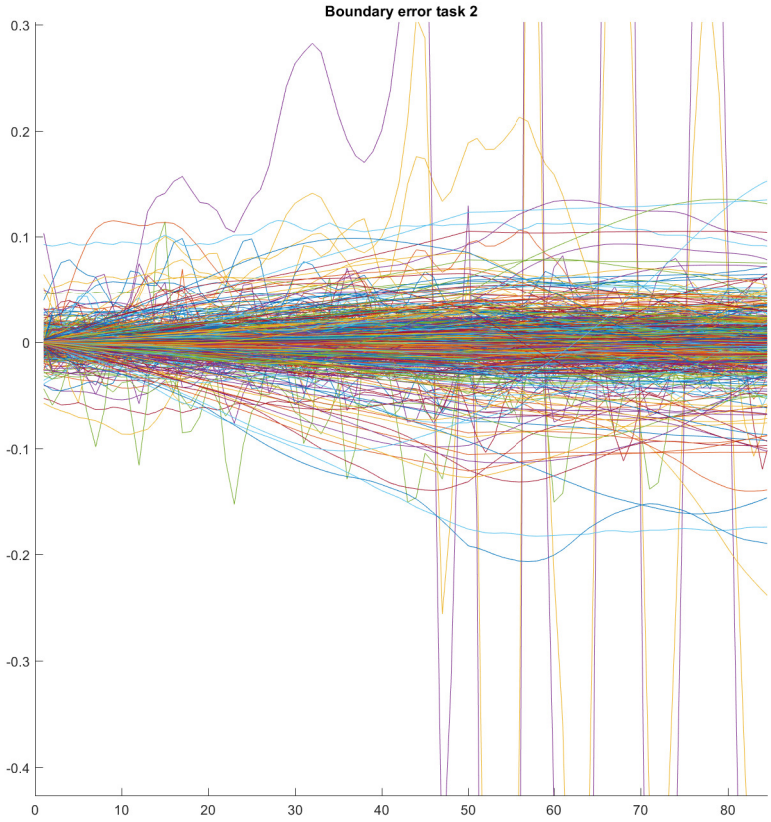
There is some kind of disturbance in sound at the begining of the processed signal. This time, the smaller the block size the more of this sound we can hear. Which means that this disturbance is the representation of the processing error at block boundaries.

Lets plot data and boundary errors:

2.0.1 Signal


Signal vs processed signal task 2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
2
4
6
8
10
12
×10^5

2.0.2 Boundary error

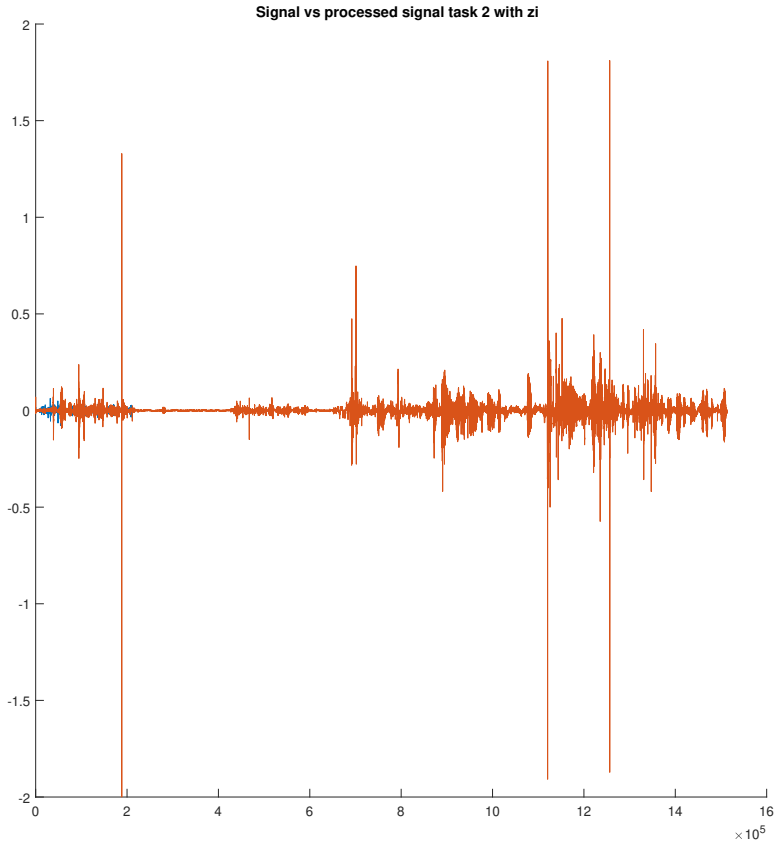


As we can see the boundary error was nicely decreased, especially the 'violet' signal looks promising.

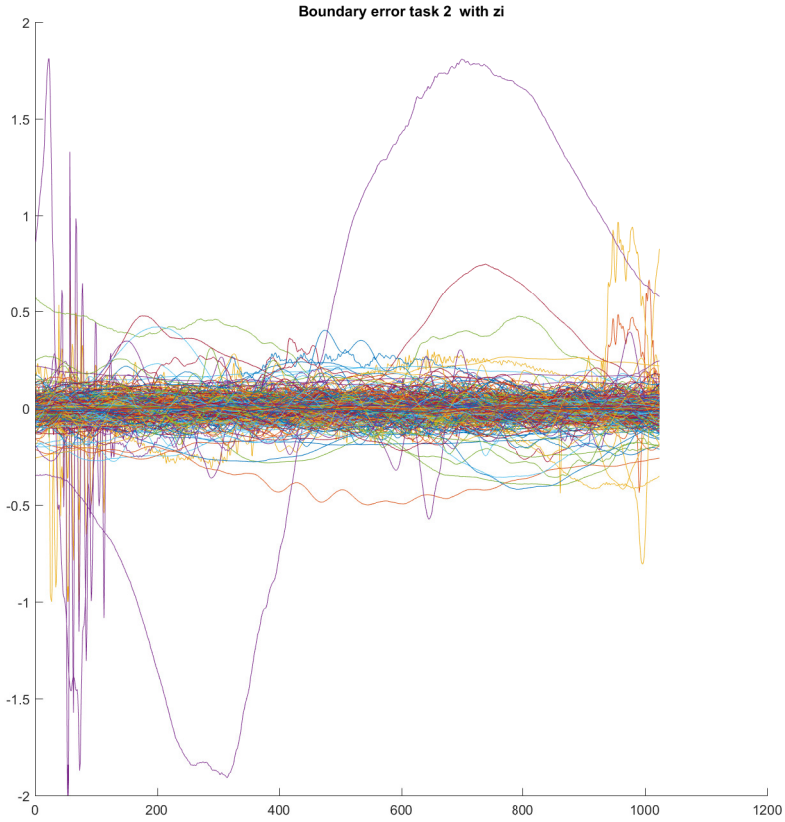
2.1 Filter with zi and zf parameters

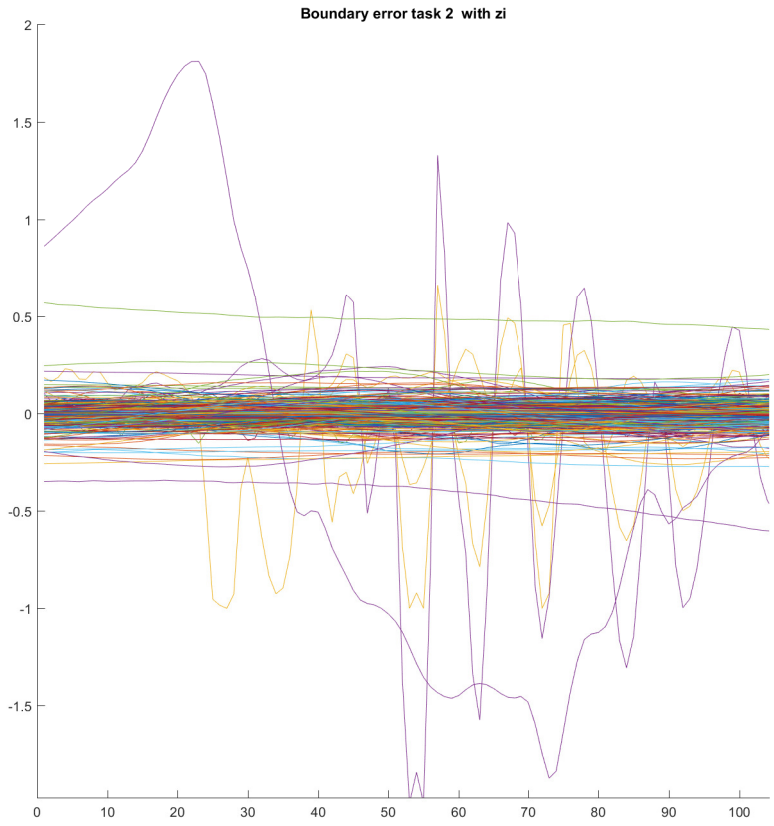
Lets plot data and boundary with z_i and z_f parameters:

2.1.1 Signal


Signal vs processed signal task 2 with zi
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
2
4
6
8
10
12
14
16
×10^5

2.1.2 Boundary error





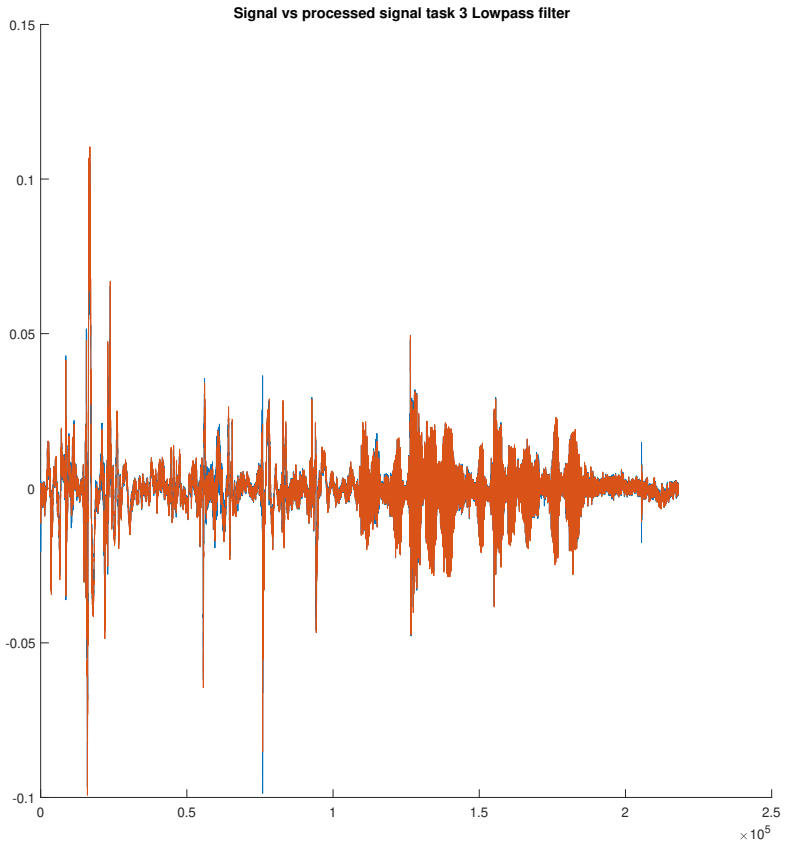
Boundary error was even further reduced.

Chapter 3

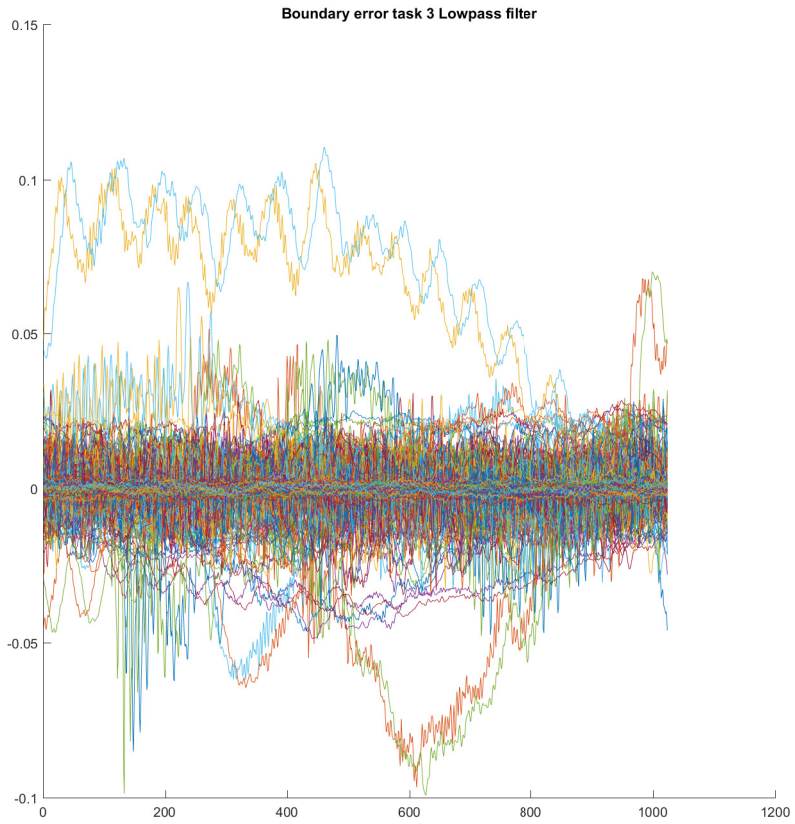
Task 3

3.1 Lowpass filter experiment

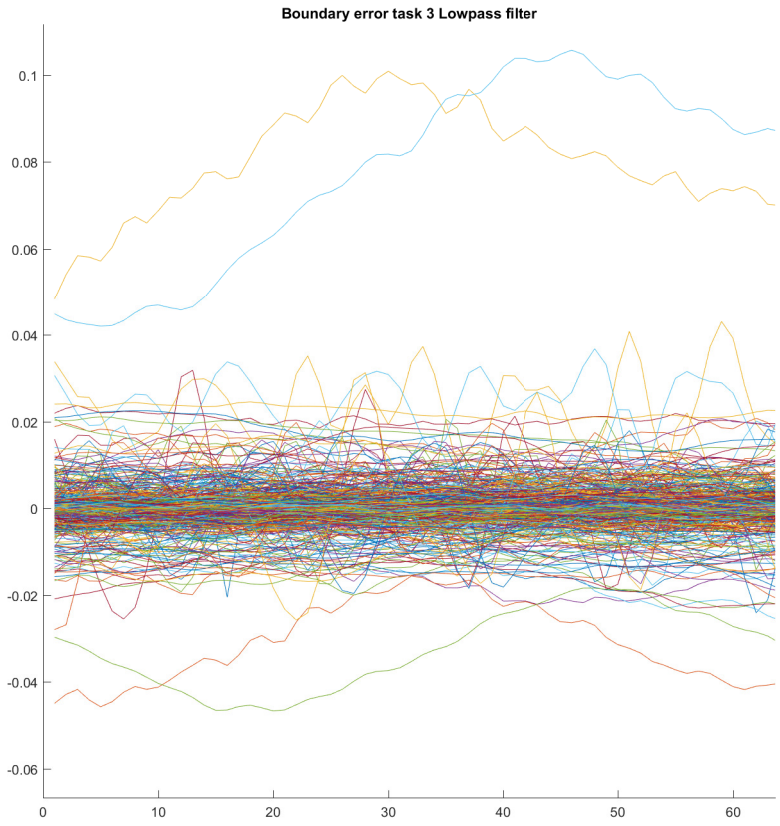
I decided to do lowpass filter with 8kHz passband and 12 kHz stopband Signal and processed signal plot:



Boundary error plot:

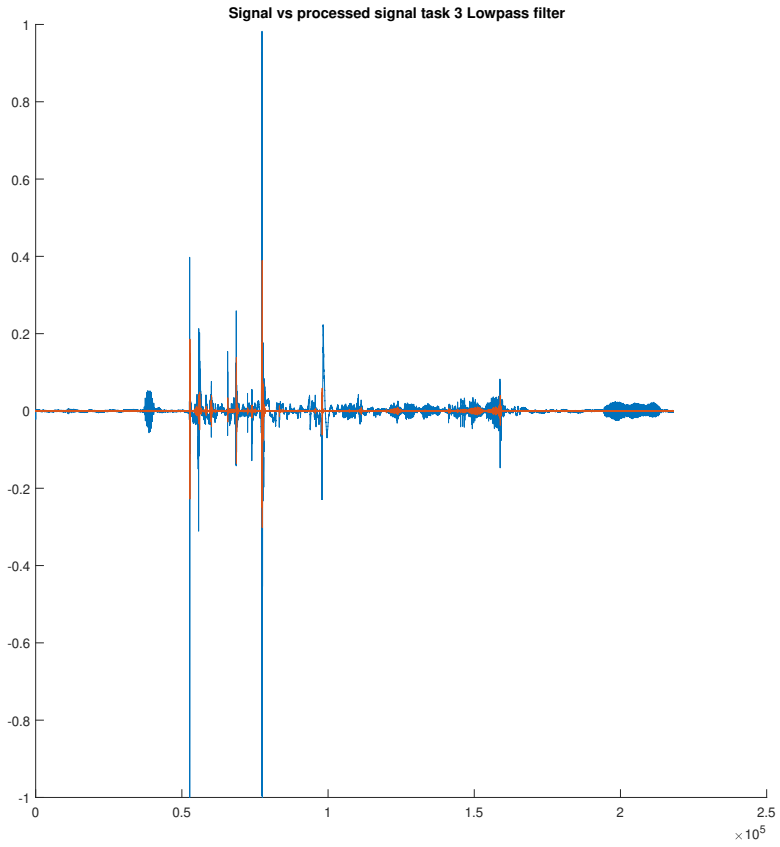


Boundary error plot zoom:

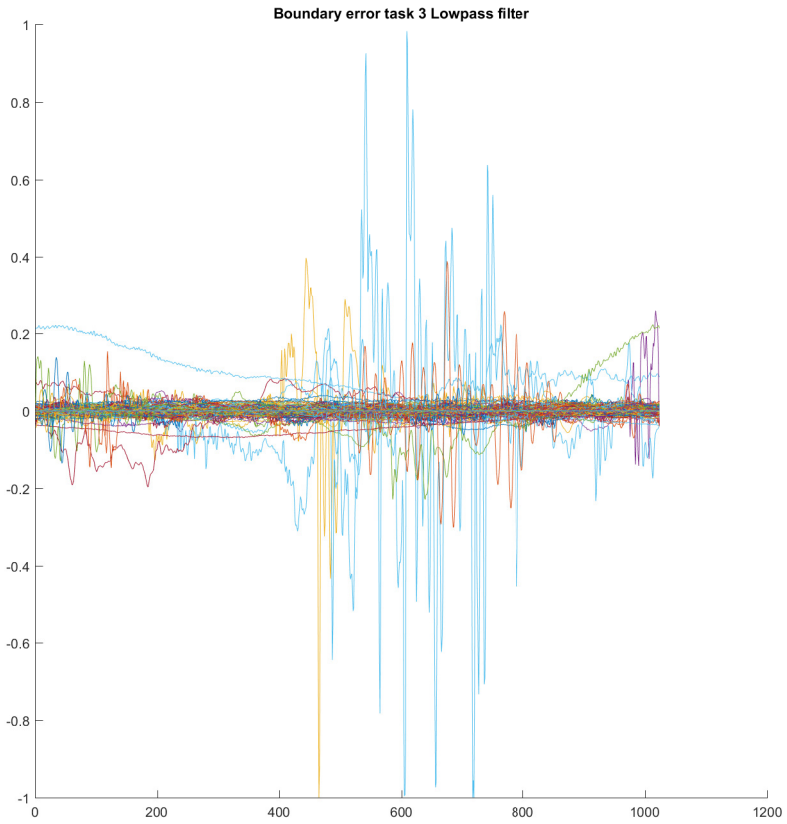


3.2 Bandpass filter experiment

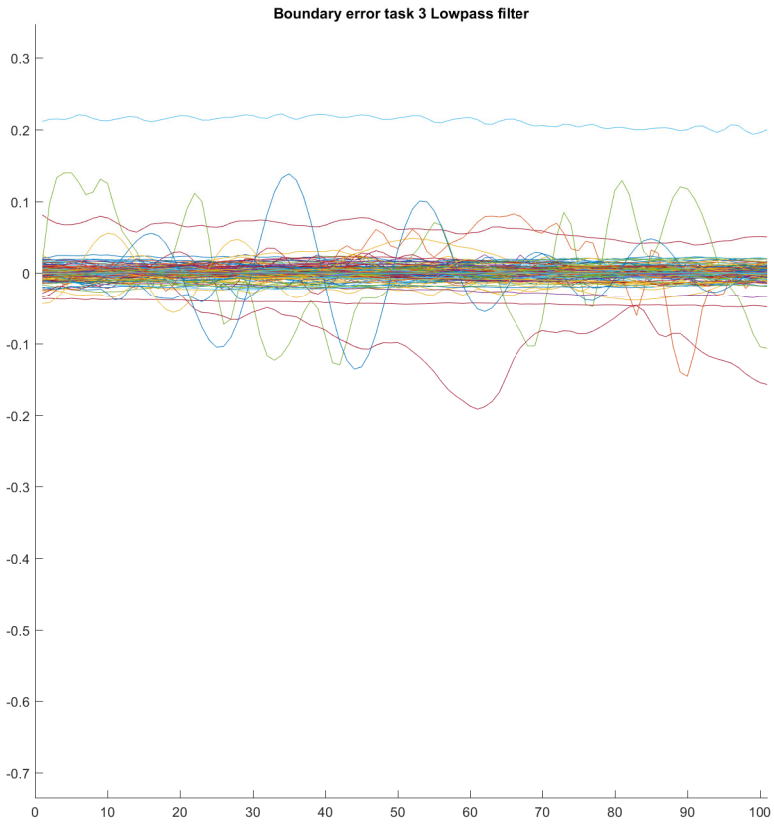
I decided to do Bandpass filter with first stopband = 1kHz, first passband = 2 kHz, second passband = 3 kHz, and second stopband = 4 kHz Signal and processed signal plot:



Boundary error plot:



Boundary error plot zoom:

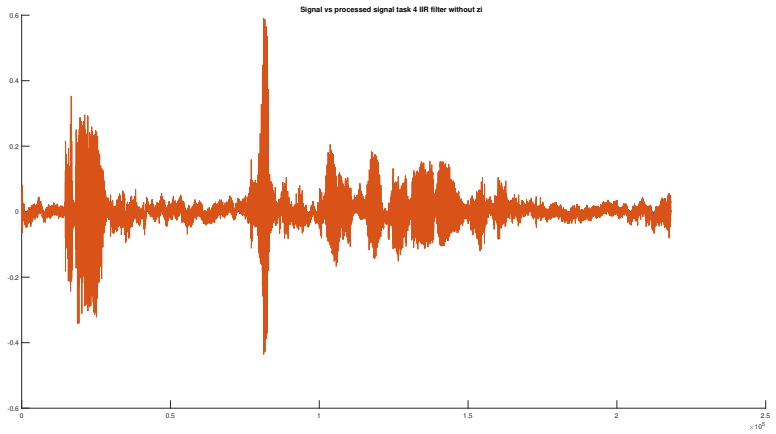

Boundary error task 3 Lowpass filter
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
0
10
20
30
40
50
60
70
80
90
100

Chapter 4

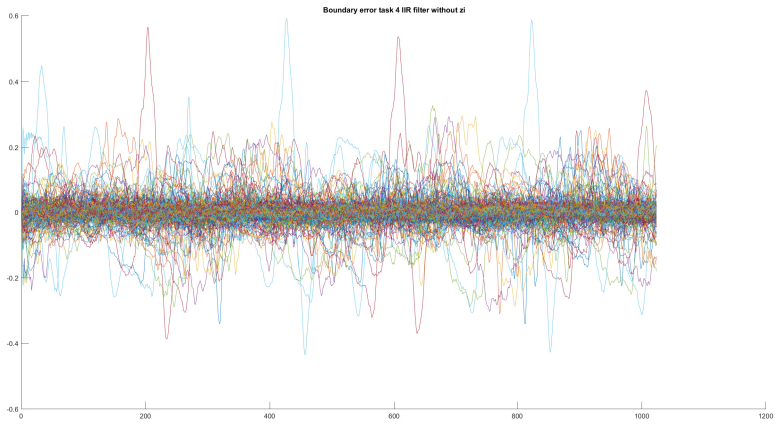
Task 4

Unfortunately we had to use not the most optimal way of transforming sos by using function sos2tf() in order to use filter function in order to be able to receive z_i paramter from it. It is possiblr if we do not need the parameter to use sosfilt() function but for sake of consistency I opt out of this solution. I made a IIR filter with 10^{th} order and stop frequency of 10 kHz.

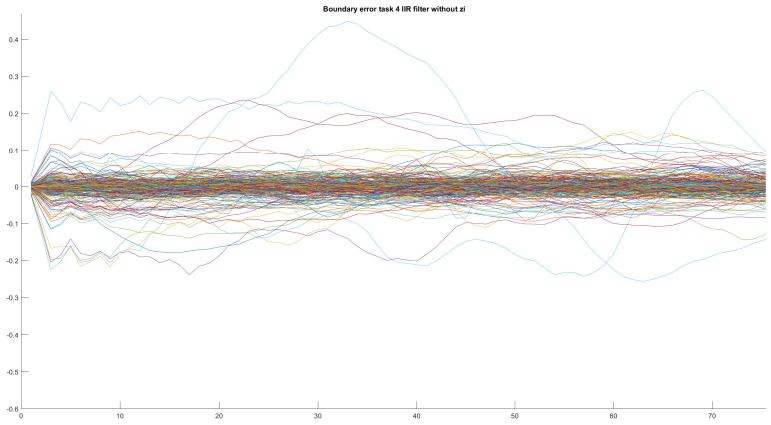
4.1 IIR filter without z_i



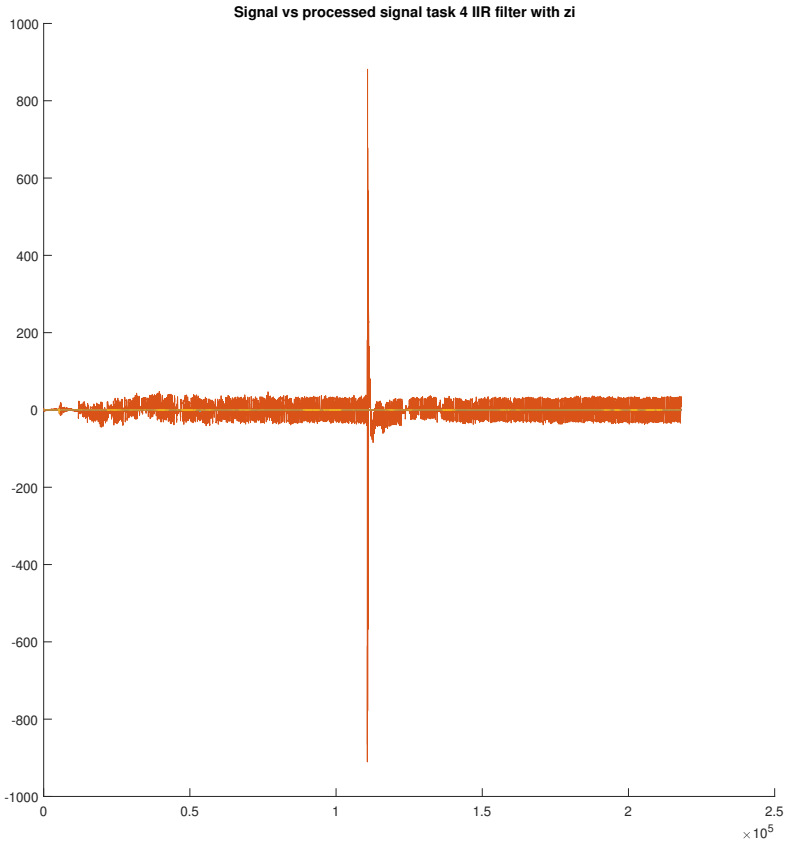
Boundary error plot:



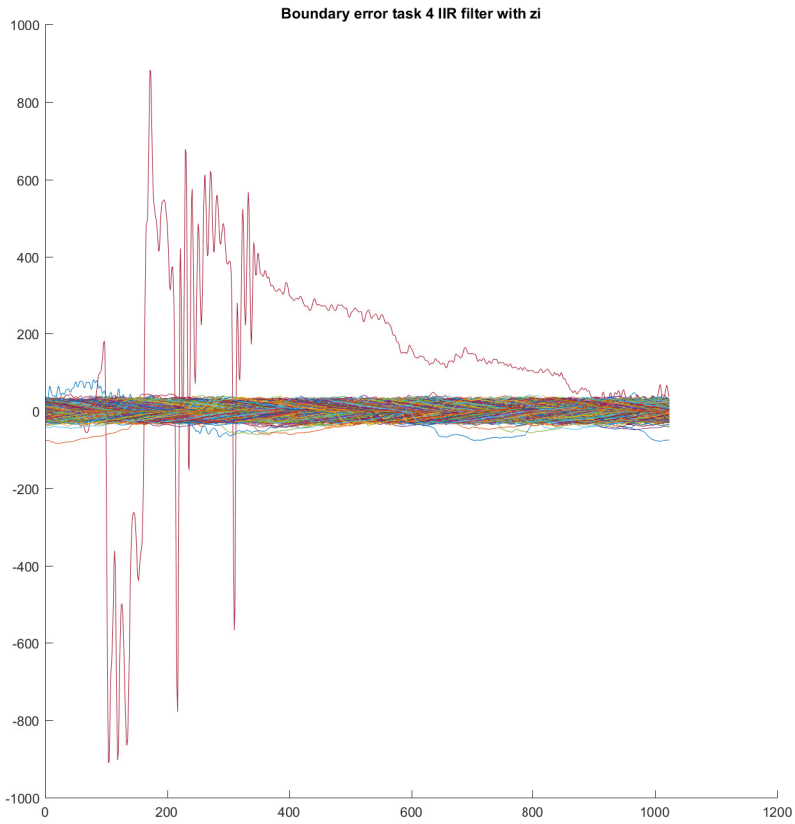
Boundary error plot zoom:



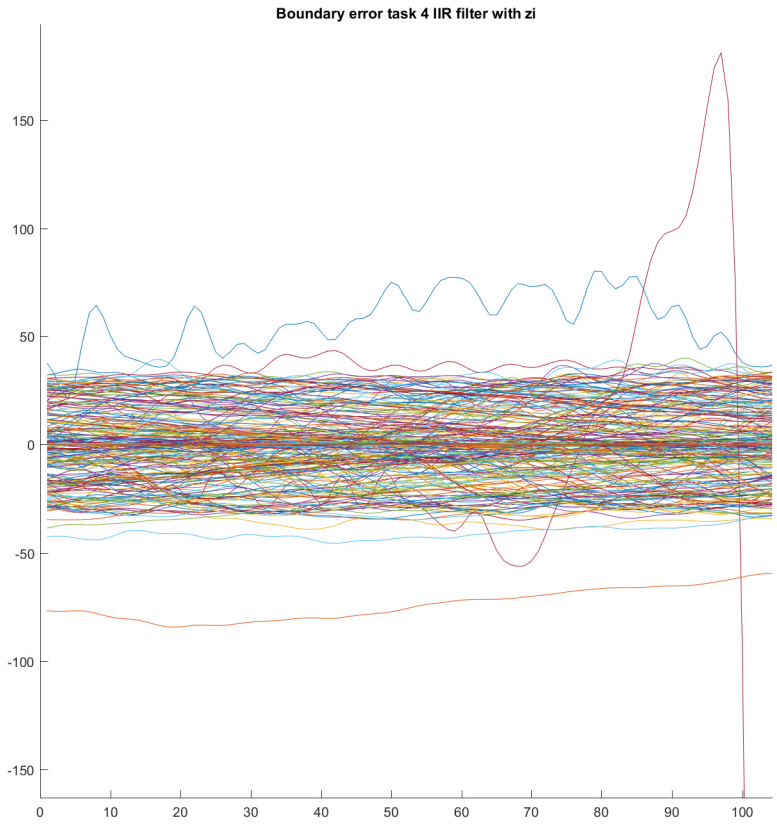
4.2 IIR filter with z_i



Boundary error plot:



Boundary error plot zoom:



4.3 Questions

- Is it possible to correct errors now by remembering a fragment of the previous block? Yes
- If zi is a vector, then its lenghth must be max (length(a), length(b)) − 1, why? Since in the loop first iteration skips the zi we really start using it in second iteration so we need one less length of zi vector than the max of one of coeffcients,