Digital signal processing - Experiment with Data - TRAN Gia Quoc Bao, ASI

Table of Contents

Default commands 1
I. Context and variables to be analyzed 1
II. Spectral analysis 1
2.1. Preliminaries 1
2.2. Spectral analysis 3
III. Digital FIR filters 6
3.1 Low-pass filter for yearly pattern 6
3.2 Band-pass filter for monthly pattern 9
3.3 High-pass filter for daily pattern 13
IV. Comparison and conclusion 16

Default commands

clear all

```
close all
clc
```

I. Context and variables to be analyzed

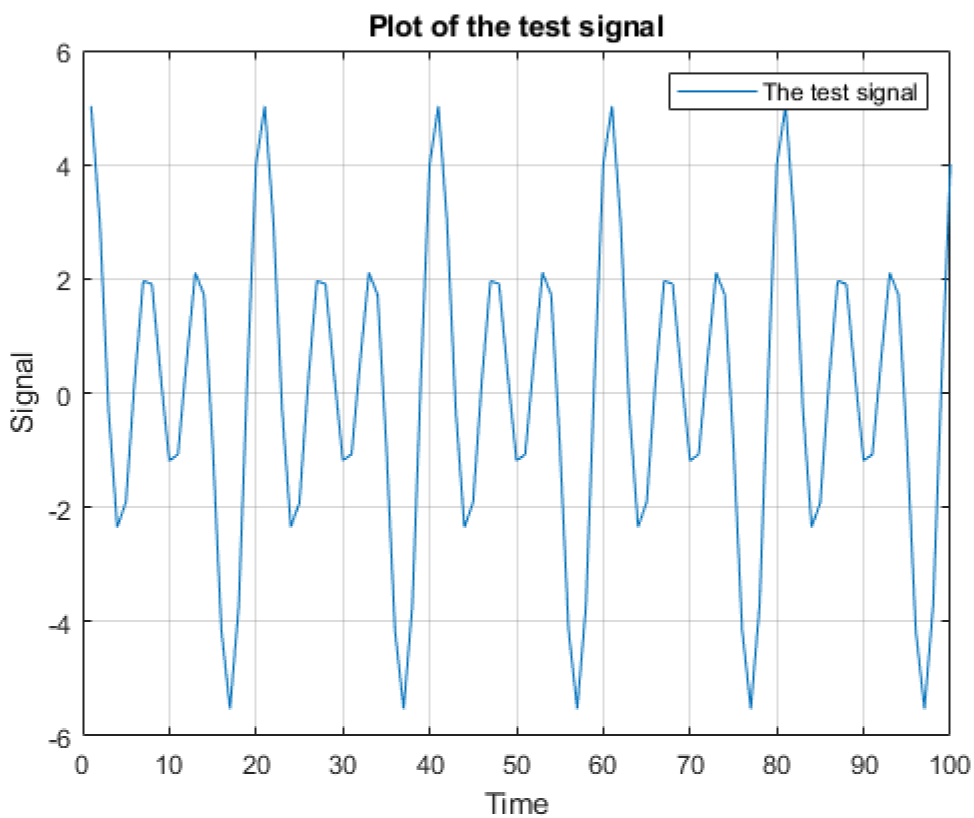
```
% Here we need to analyze the frequencies inside a signal to determine
% them (part II) and then to design suitable filters to extract those
% frequencies (part III). Finally there will be some conclusions on
 what we
% have done so far, in part IV.
```

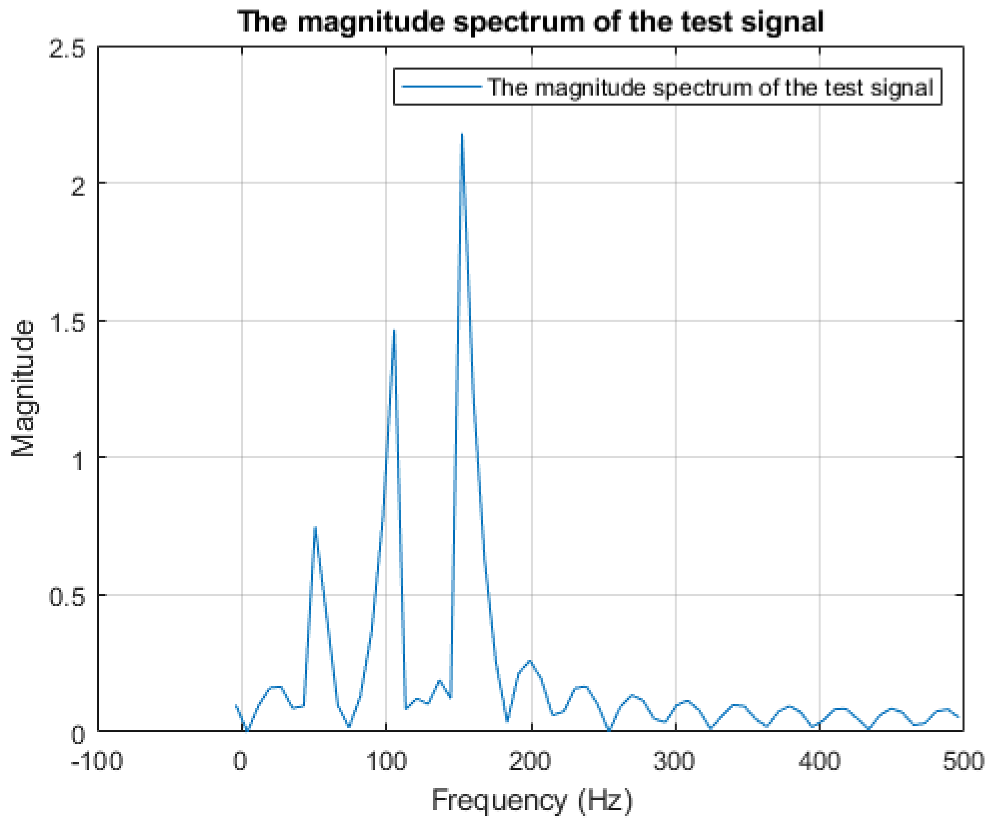
II. Spectral analysis

2.1. Preliminaries

```
% We are going to build a test signal and then do some experiments
 with it
% to make sure our tools for signal analysis work well.

% Making a test signal to test the new TFDVisu file
fs_test = 1000; %sampling frequency
A1 = 1; f1 = 50; fo1 = f1/fs_test; phi1 = 0;
A2 = 2; f2 = 100; fo2 = f2/fs_test; phi2 = pi/4;
```

```
A3 = 3; f3 = 150; fo3 = f3/fs_test; phi3 = pi/3;
Ntest = 100;
t = 1:Ntest;
sig_test = A1*sin(2*pi*fo1*t + phi1) + A2*sin(2*pi*fo2*t + phi2) +
 A3*sin(2*pi*fo3*t + phi3);

% Plotting the test signal
figure(1);
plot(t, sig_test);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The test signal');
title('Plot of the test signal');

% Applying the new TFDVisu file (now it's called "Spectrum")
figure(2);
[magnitude_test frequency_test] = Spectrum('test signal', sig_test,
 fs_test);

% Here we see that the "spectrum" file was able to tell us the 3
 special
% frequencies in the test signal
```

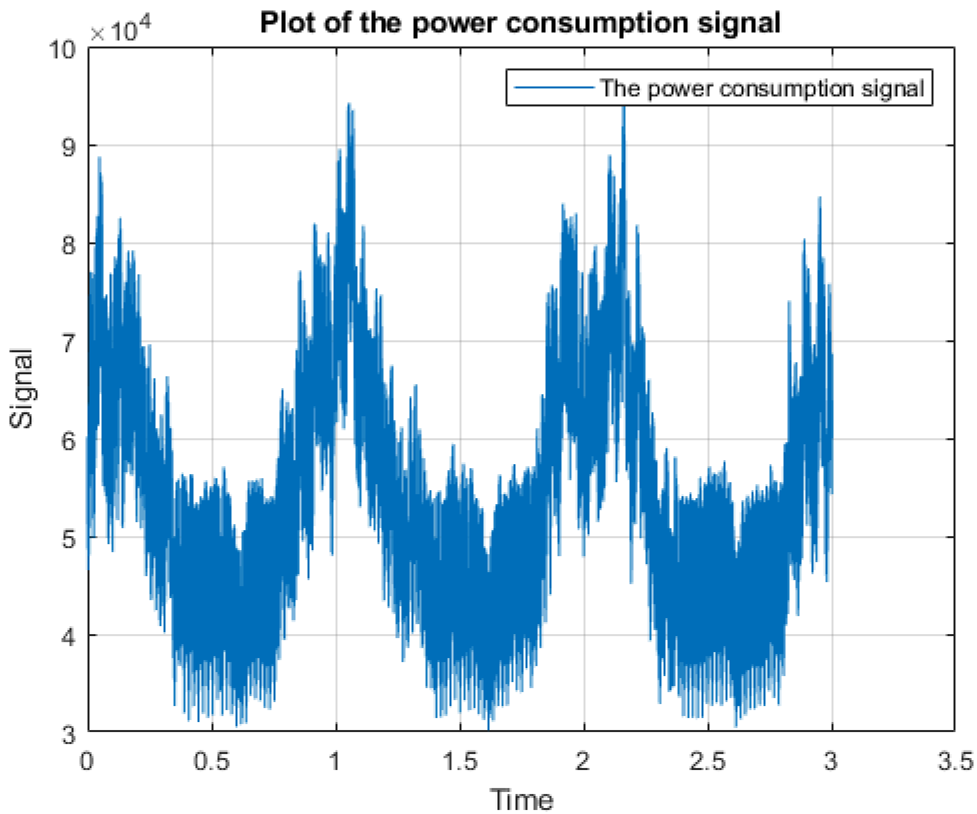


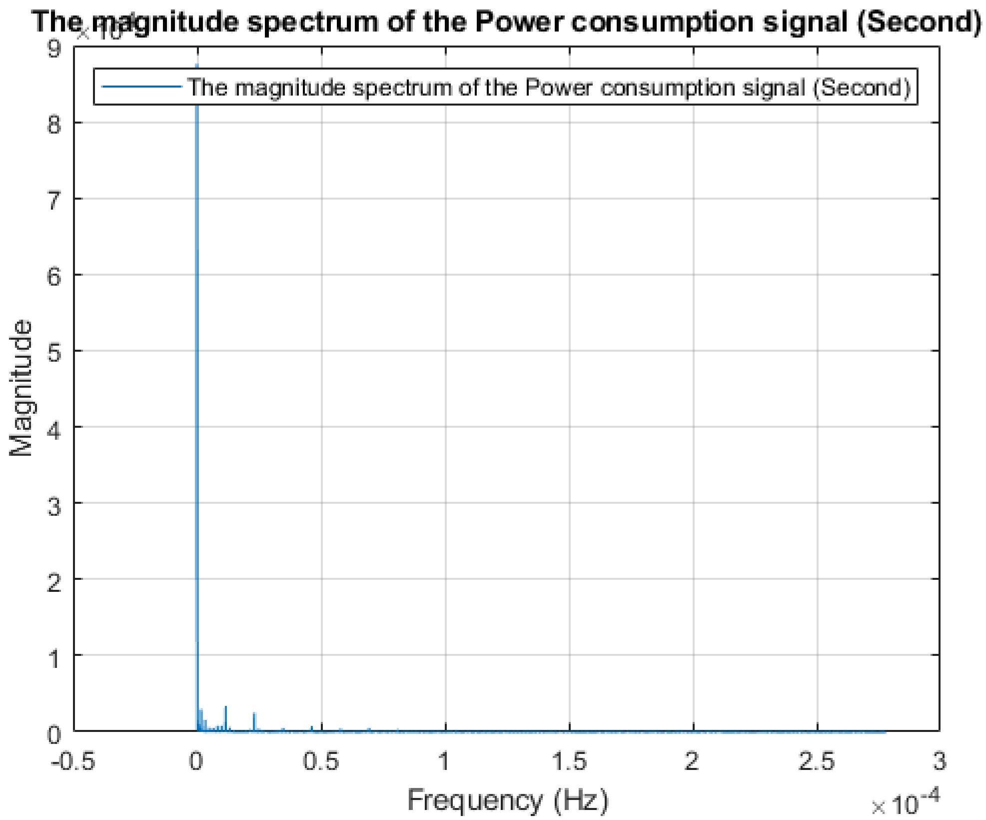


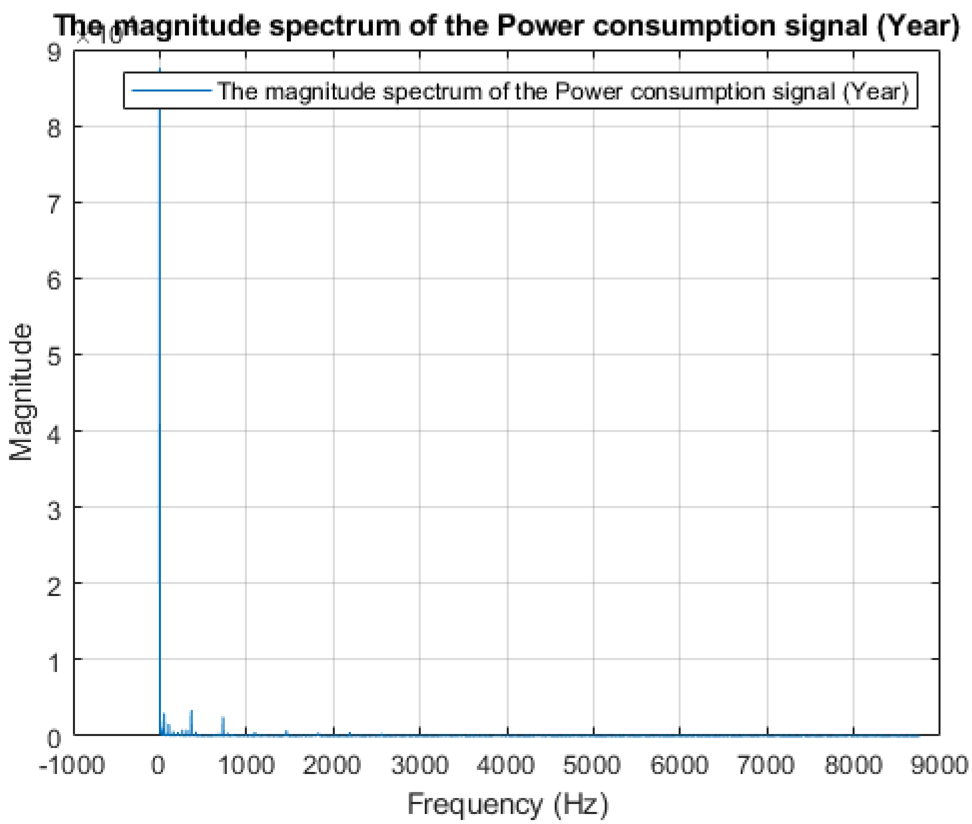
2.2. Spectral analysis

```
% The signal we want to analyze is electricty consusmption in France
 for a
% few years. We would like to spot the special, interesting
 frequencies in
% order to know what kinds of filters need to be made.

% signalConsumption = ConsommationbrutelectricitMWRTE;
% signalConsumption = consoelec20160101 20181231.ConsommationMW';
% signalTime = consoelec2016010120181231.Heure';
signalConsumption = ConsommationMW;
N = length(signalConsumption);
fsSecond = 1/(30*60);
fsYear = 2*24*365;
fsWeek = 2*24*7;
fsDay = 2*24;

time = 0 : 1/fsSecond : (N - 1)/fsSecond;
timeYear = 0 : 1/fsYear : (N - 1)/fsYear;
timeWeek = 0 : 1/fsWeek : (N - 1)/fsWeek;
timeDay = 0 : 1/fsDay : (N - 1)/fsDay;

% Plotting the power consumption signal
figure(3);
```

```
plot(timeYear, signalConsumption);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The power consumption signal');
title('Plot of the power consumption signal');

% Applying spectral analysis on the power consumption signal
figure(4);
[magnitude_signalSecond frequency_signalSecond] = Spectrum('Power
 consumption signal (Second)', signalConsumption(2:end), fsSecond);
figure(5);
[magnitude_signalYear frequency_signalYear] = Spectrum('Power
 consumption signal (Year)', signalConsumption(2:end), fsYear);

% Here in the year time scale analysis, I observed 1 peak at 1.203
 (yearly
% pattern), 1 at 52.2638 (weekly pattern) and 1 at 365.0446 (daily
% pattern). So I need to design 1 low-pass, 1 band-pass, and 1 high-
pass
% filter respectively for each of these important frequencies.
```




The magnitude spectrum of the Power consumption signal (Second)
The magnitude spectrum of the Power consumption signal (Second)
Magnitude
Frequency (Hz)


The magnitude spectrum of the Power consumption signal (Year)
The magnitude spectrum of the Power consumption signal (Year)
Magnitude
Frequency (Hz)

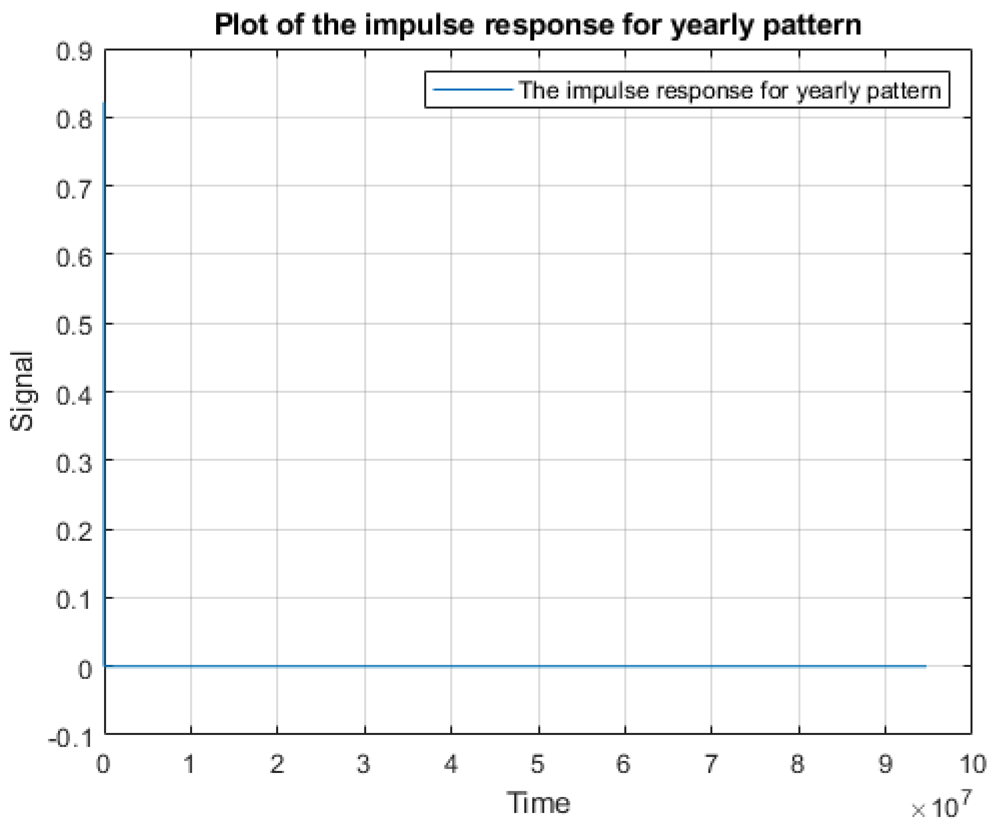
III. Digital FIR filters

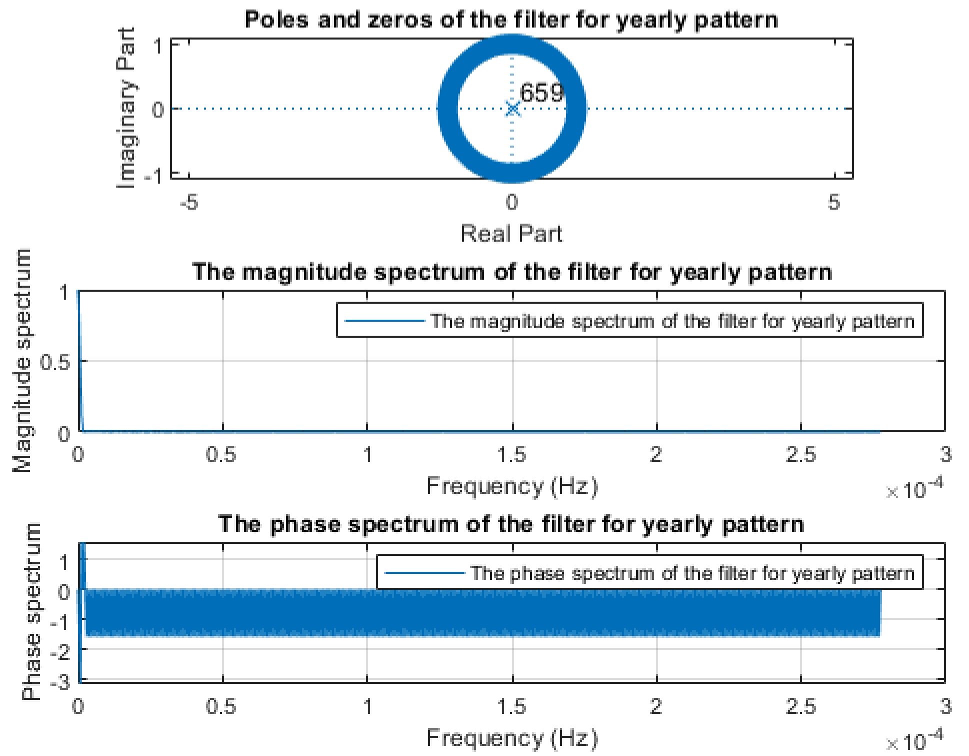
```
% We are trying to separate the components corresponding to each of
 the
% cycles we determined.
% So we will first use a low-pass FIR filter stopping before f = 52
% cycles/years, then a band-pass around 52 and before 365 cycles/year
% and a high-pass from 365 cycles/year.
```

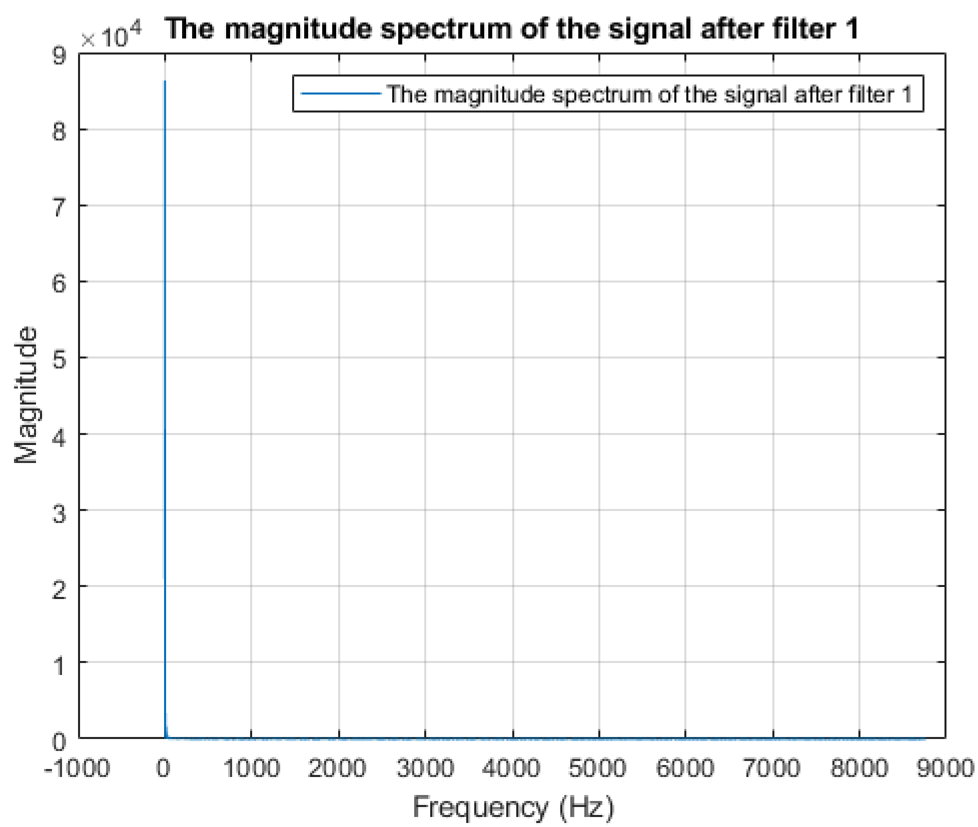
3.1 Low-pass filter for yearly pattern

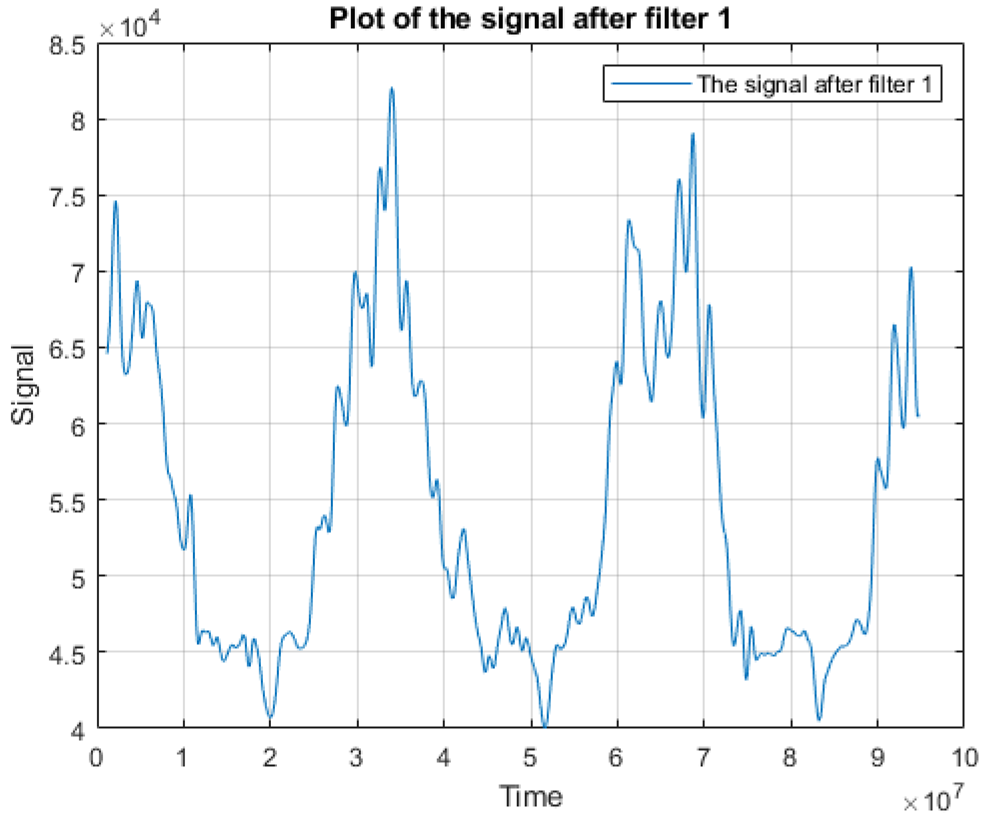
```
% The frequency we want is 1.203 cycles/year. We will use a cut
 frequency
% at 2 as we know that this will include the one we need.

% The desired impulse response
fc1 = 2*2/fsYear;
lambda1 = fc1/fsSecond;
impulseResponse1 = 2*lambda1*sinc(2*pi*lambda1*time);

% This step is optional. Normally with MATLAB we just need to use fir1
 and
% then filter.

% Plotting the desired impulse response
figure(6);
plot(time, impulseResponse1);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The impulse response for yearly pattern');
title('Plot of the impulse response for yearly pattern');

% Making the 1st window (Hamming)
deltaWindow1 = 0.005;
sizeWindow1 = round(3.3/deltaWindow1 - 1);
window1 = window(@hann, sizeWindow1);

% This step is optional. Normally with MATLAB we just need to use fir1
 and
% then filter.

% Making the 1st filter
filter1 = fir1(sizeWindow1, fc1, 'low');

% Analysis of the 1st filter
figure(7);
[magnitude1 phase1 frequency1] = FilterVisu('filter for yearly
 pattern', filter1, 1, sizeWindow1, fsSecond);

% Applying the 1st filter on the signal
signalFiltered1 = filter(filter1, 1, signalConsumption);
```

```
% Analysis of the obtained signal
figure(8);
[magnitude_signalFiltered1 frequency_signalFiltered1] =
 Spectrum('signal after filter 1', signalFiltered1(sizeWindow1 + 2 :
 end), fsYear);

% Looking at this we see that the high frequencies are filtered away
 and
% the small ones are much clearer. This shows the effectiveness of the
% low-pass filter we implemented.

figure(9);
plot(time, signalFiltered1);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The signal after filter 1');
title('Plot of the signal after filter 1');

% We observe the yearly pattern: the same cycle repeats year after
 year.
% We have a peak consumption in winter, a decrease in summer, as well
 as
% various more localized events (holidays, vacations, important
 events ...)
```




Poles and zeros of the filter for yearly pattern
Imaginary Part
Real Part
659
1
0
-1
-5
0
5
The magnitude spectrum of the filter for yearly pattern
Magnitude spectrum
The magnitude spectrum of the filter for yearly pattern
1
0.5
0
0
0.5
1
1.5
2
2.5
3
Frequency (Hz)
×10^-4
The phase spectrum of the filter for yearly pattern
Phase spectrum
The phase spectrum of the filter for yearly pattern
1
0
-1
-2
-3
0
0.5
1
1.5
2
2.5
3
Frequency (Hz)
×10^-4

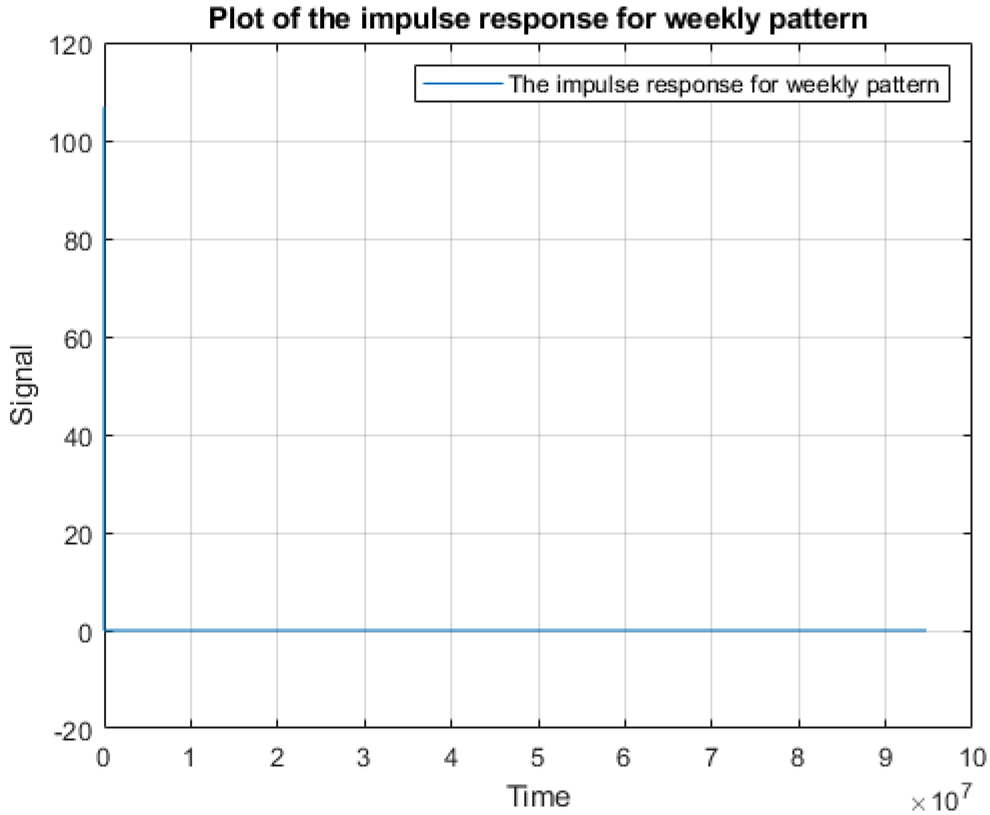

The magnitude spectrum of the signal after filter 1
×10^4
The magnitude spectrum of the signal after filter 1
Magnitude
9
8
7
6
5
4
3
2
1
0
-1000
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Frequency (Hz)

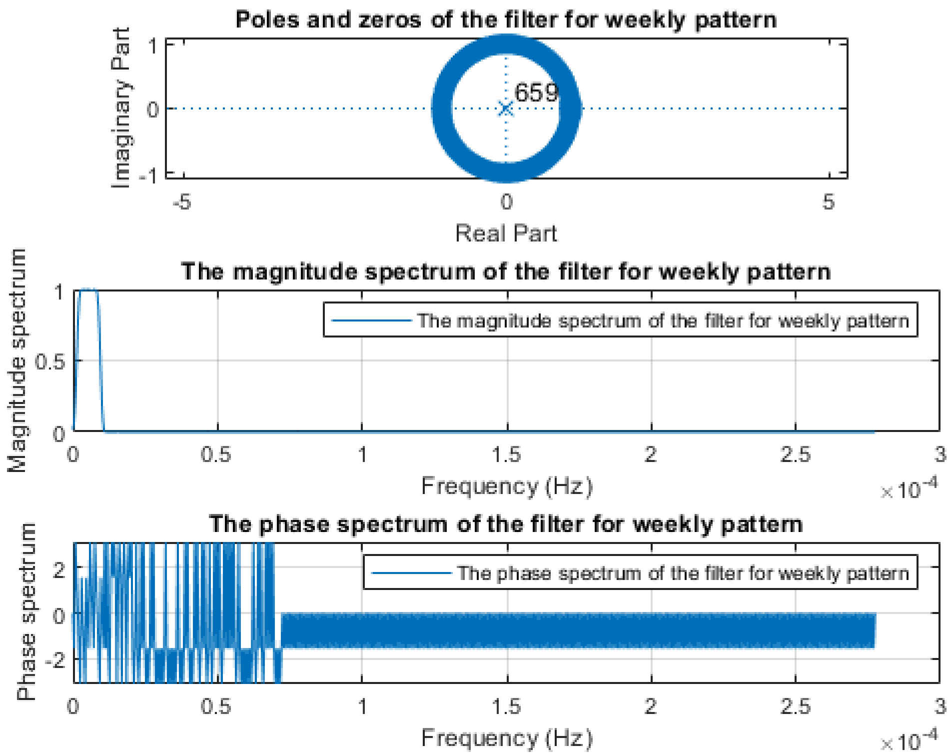


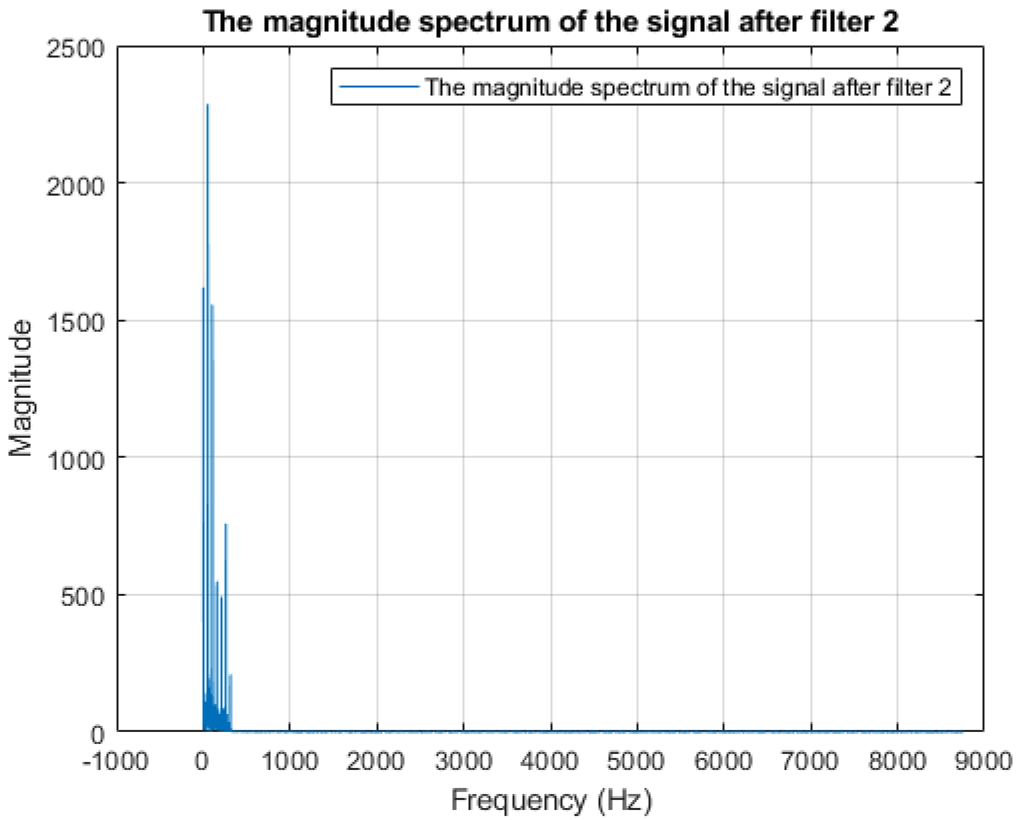
3.2 Band-pass filter for monthly pattern

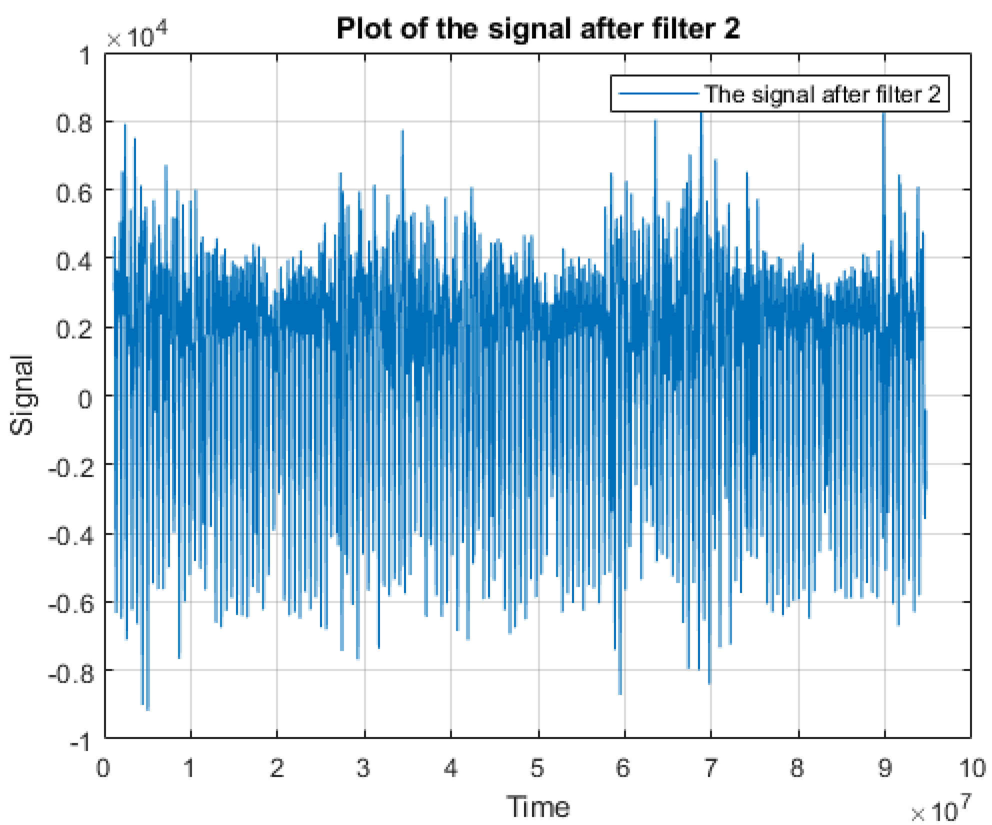
```
% The frequency we want is 52.2638 cycle/year. We will use cut
 frequencies
% at 40 and 300 as we know that this will include the one we need.

% The desired impulse response
fc2_small = 2*40/fsYear;
fc2_big = 2*300/fsYear;
lambda2_small = fc2_small/fsSecond;
lambda2_big = fc2_big/fsSecond;
impulseResponse2 = 2*lambda2_big*sinc(2*pi*lambda2_big*time) -
 2*lambda2_small*sinc(2*pi*lambda2_small*time);

% This step is optional. Normally with MATLAB we just need to use fir1
 and
% then filter.

% Plotting the desired impulse response
figure(10);
plot(time, impulseResponse2);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The impulse response for weekly pattern');
```

```
title('Plot of the impulse response for weekly pattern');

% Making the 2nd window (Hamming)
deltaWindow2 = 0.005;
sizeWindow2 = round(3.3/deltaWindow2 - 1);
window2 = window(@hann, sizeWindow2);

% This step is optional. Normally with MATLAB we just need to use fir1
 and
% then filter.

% Making the 2nd filter
filter2 = fir1(sizeWindow2, [fc2_small fc2_big]);

% Analysis of the 2nd filter
figure(11);
[magnitude2 phase2 frequency2] = FilterVisu('filter for weekly
 pattern', filter2, 1, sizeWindow2, fsSecond);

% Applying the 2nd filter on the signal
signalFiltered2 = filter(filter2, 1, signalConsumption);

% Analysis of the obtained signal
figure(12);
[magnitude_signalFiltered2 frequency_signalFiltered2] =
 Spectrum('signal after filter 2', signalFiltered2(sizeWindow2 + 2 :
 end), fsYear);

% By looking at this we see that the components with frequencies in
 our
% desired zone are kept and the ones outside are reduced greatly by
 the
% filter.

figure(13);
plot(time, signalFiltered2);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The signal after filter 2');
title('Plot of the signal after filter 2');

% We also have a weekly cycle, where the 5 working days with
 consumption
% generally stronger and on weekends when consumption decreases.
```


Plot of the impulse response for weekly pattern
The impulse response for weekly pattern
Signal
Time
$\times 10^7$


Poles and zeros of the filter for weekly pattern
Imaginary Part
Real Part
659
The magnitude spectrum of the filter for weekly pattern
Magnitude spectrum
Frequency (Hz)
$\times 10^{-4}$
The phase spectrum of the filter for weekly pattern
Phase spectrum
Frequency (Hz)
$\times 10^{-4}$

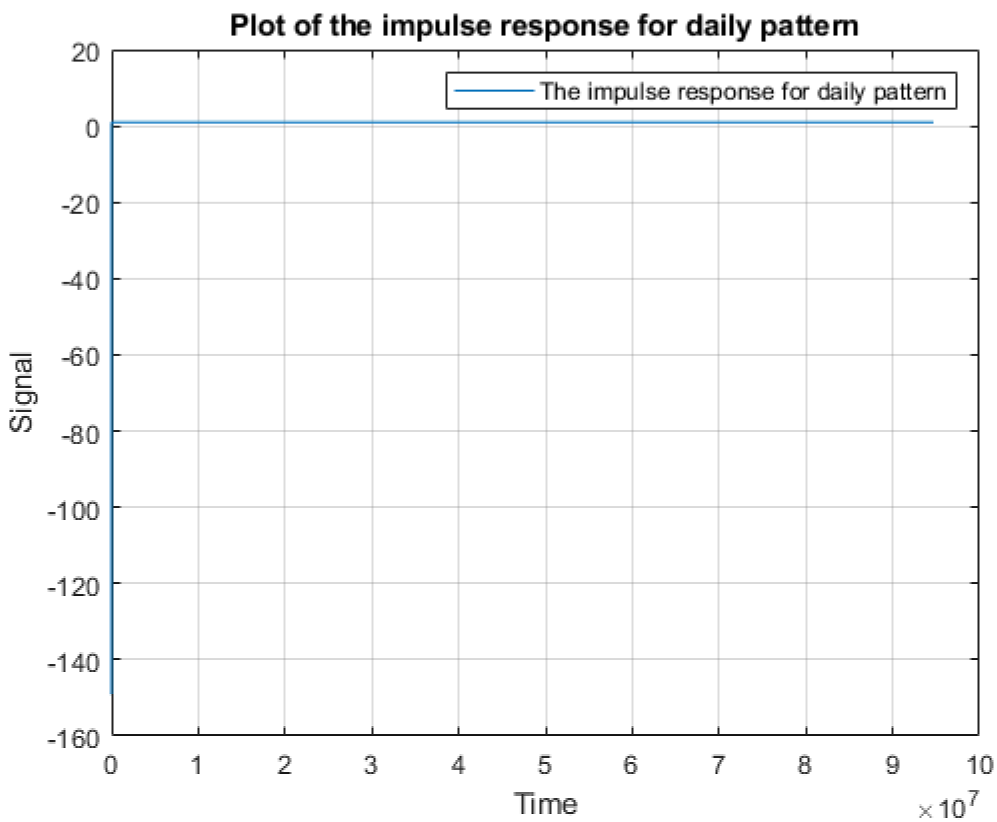

The magnitude spectrum of the signal after filter 2
The magnitude spectrum of the signal after filter 2
Magnitude
2500
2000
1500
1000
500
0
-1000
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Frequency (Hz)

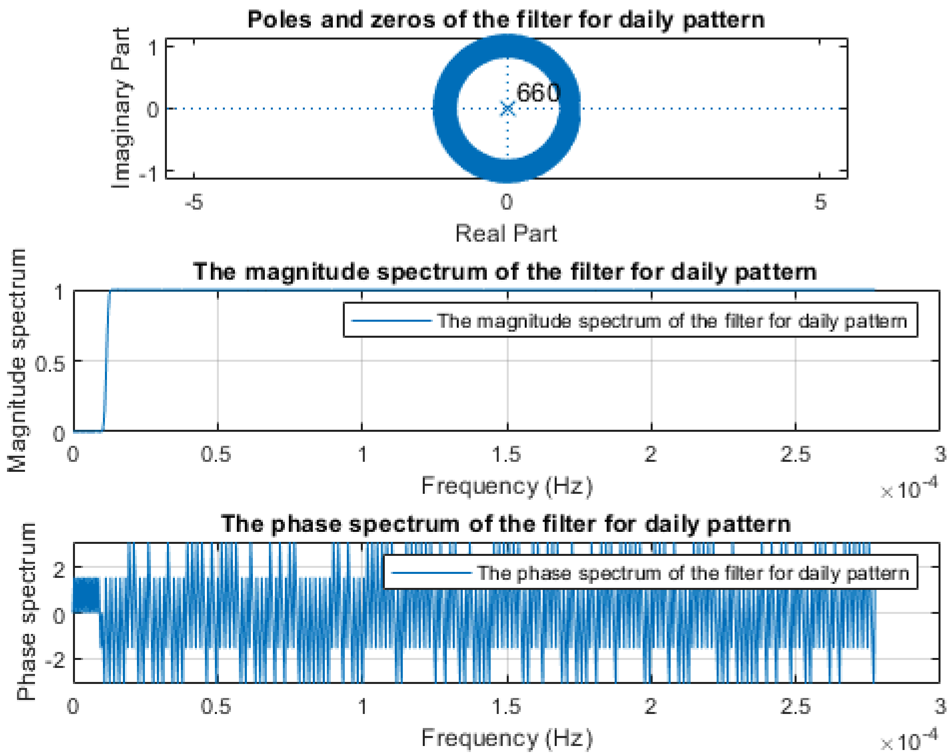

Plot of the signal after filter 2
×10^4
The signal after filter 2
Signal
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
7
8
9
10
Time
×10^7

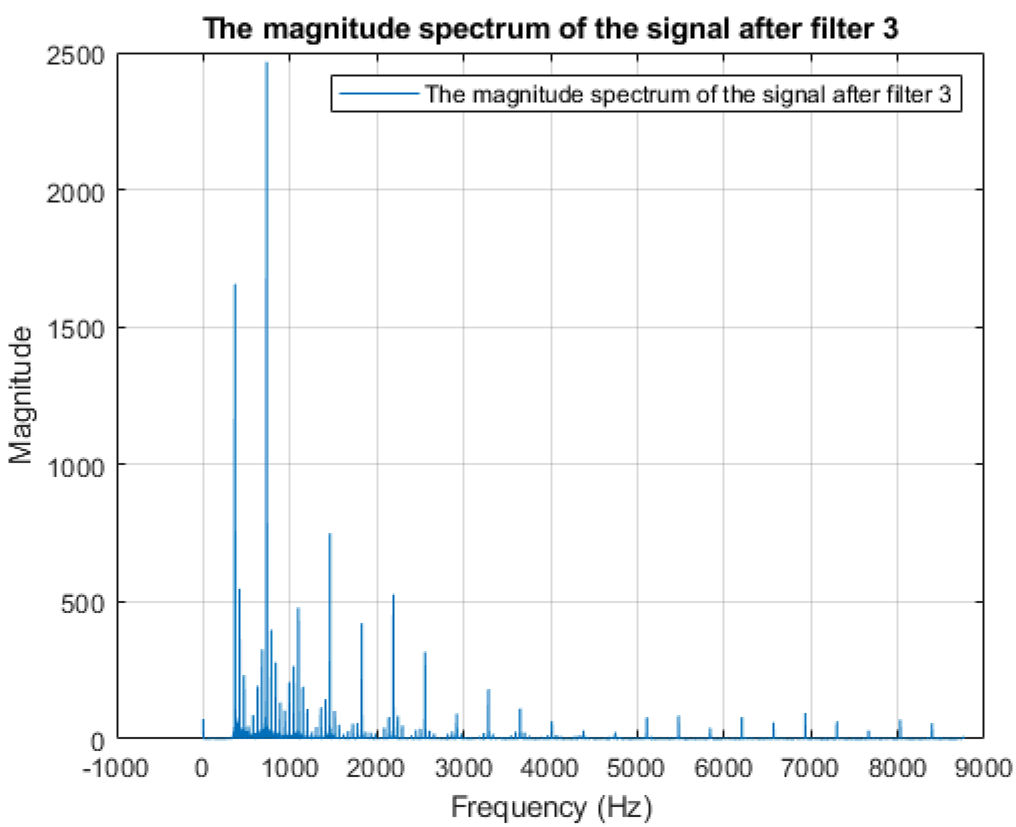
3.3 High-pass filter for daily pattern

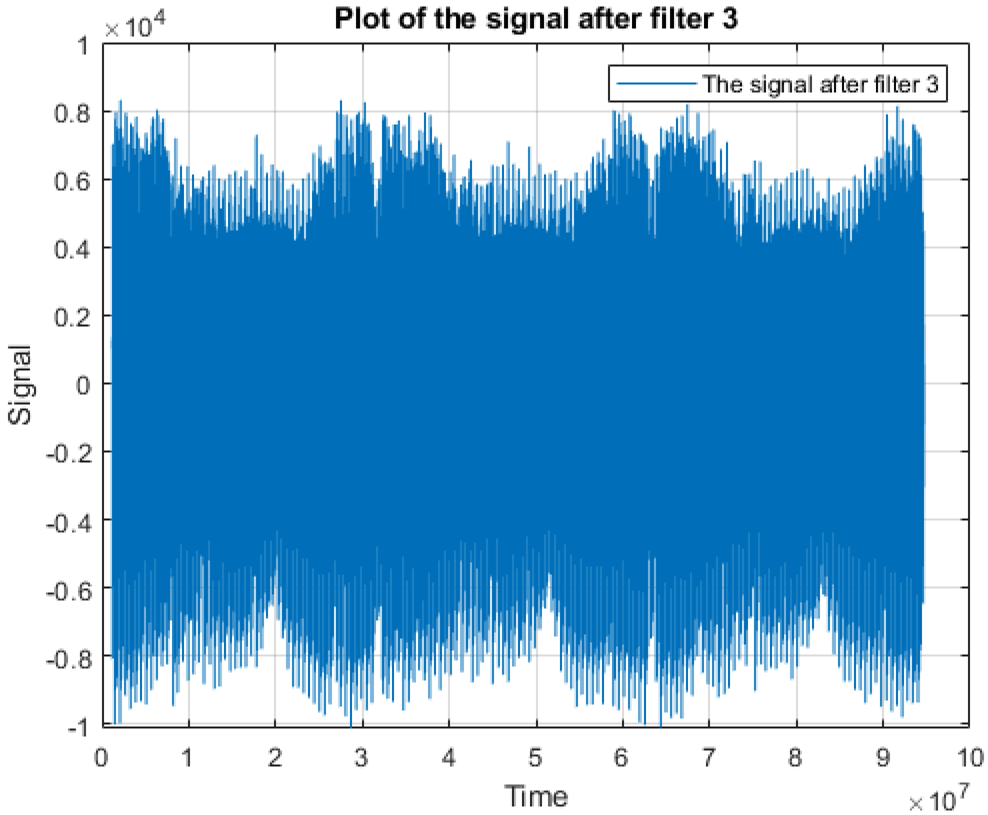
```
% The frequency we want is 365.0446 cycles/ year. We will use cut
% frequencies at 365.0446 as we know that this will include the one we
 need.

% The desired impulse response
fc3 = 2*365.0446/fsYear;
lambda3 = fc3/fsSecond;
impulseResponse3 = 1 - 2*lambda3*sinc(2*pi*lambda3*time);

% This step is optional. Normally with MATLAB we just need to use fir1
 and
% then filter.

% Plotting the desired impulse response
figure(14);
plot(time, impulseResponse3);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The impulse response for daily pattern');
title('Plot of the impulse response for daily pattern');

% Making the 3rd window (Hamming)
deltaWindow3 = 0.005;
sizeWindow3 = round(3.3/deltaWindow3 - 1) + 1;
window3 = window(@hann, sizeWindow3);

% This step is optional. Normally with MATLAB we just need to use fir1
 and
% then filter.

% Making the 3rd filter
filter3 = fir1(sizeWindow3, fc3, 'high');

% Analysis of the 3rd filter
figure(15);
[magnitude3 phase3 frequency3] = FilterVisu('filter for daily
 pattern', filter3, 1, sizeWindow3, fsSecond);

% Applying the 3rd filter on the signal
signalFiltered3 = filter(filter3, 1, signalConsumption);

% Analysis of the obtained signal
figure(16);
[magnitude_signalFiltered3 frequency_signalFiltered3] =
 Spectrum('signal after filter 3', signalFiltered3(sizeWindow3 + 3 :
 end), fsYear);

% By looking at this we see that the low frequencies have been greatly
% reduced by the filter. The 365 cycles/year is now very clear to see.
```

```
figure(17);
plot(time, signalFiltered3);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The signal after filter 3');
title('Plot of the signal after filter 3');

% Finally we have a daily cycle, consisting of the night increase
% corresponding to the minimum consumption over the 24 hours, the
 morning
% peak, the afternoon bottom and the evening peak. It can be noted
 that
% the maximum consumption is reached during the morning peak in
 summer,
% and the evening peak in winter
```




Poles and zeros of the filter for daily pattern
Imaginary Part
Real Part
660
The magnitude spectrum of the filter for daily pattern
Magnitude spectrum
Frequency (Hz)
The magnitude spectrum of the filter for daily pattern
The phase spectrum of the filter for daily pattern
Phase spectrum
Frequency (Hz)
The phase spectrum of the filter for daily pattern

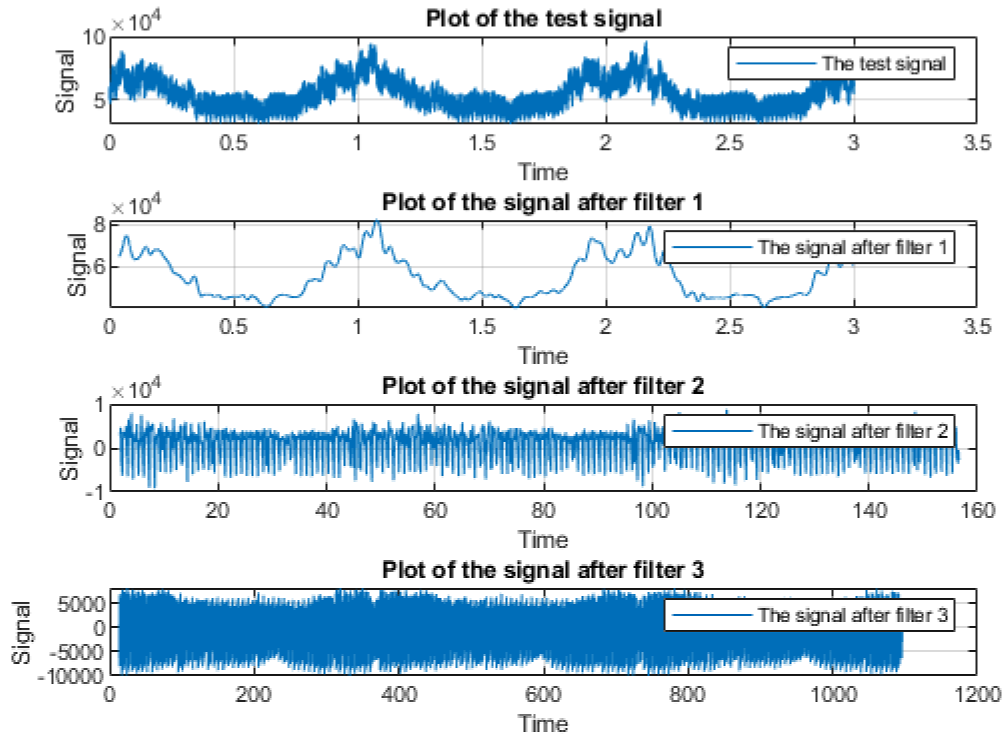

The magnitude spectrum of the signal after filter 3
The magnitude spectrum of the signal after filter 3
Magnitude
Frequency (Hz)



IV. Comparison and conclusion

```
figure(18);
subplot(411);
plot(timeYear, signalConsumption);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The test signal');
title('Plot of the test signal');
subplot(412);
plot(timeYear, signalFiltered1);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The signal after filter 1');
title('Plot of the signal after filter 1');
subplot(413);
plot(timeWeek, signalFiltered2);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The signal after filter 2');
title('Plot of the signal after filter 2');
subplot(414);
```

```
plot(timeDay, signalFiltered3);
grid on;
xlabel('Time');
ylabel('Signal');
legend('The signal after filter 3');
title('Plot of the signal after filter 3');

% In comparison we see that the signals after being filtered are very
% different from the original one and each has a meaning in terms of
 how
% electricity consumption varies according to specific frequencies.
 Also
% the filtered signals are delayed by the order of the filter so we
 could
% only plot them from there.

% What we have done so far is first to analyze the signal in order to
% detect the special frequencies thanks to the file "Spectrum". Then,
 with
% the information obtained, we designed different suitable filters,
 studied
% the filters with the file "FilterVisu" (plotting poles and zeros,
% frequency response), and applied them on the signal to process it.
% This way we can see how electricity consumption varies by a yearly,
% weekly, and daily pattern. Through this we learned how to study a
 signal
% effectively using this 2-step procedure and reviewed the tools we
 have
% had in the course.

% Some practical lessons from the exercise: we need to normalise the
 fft,
% which we need zero padding to improve the visual aspect. Depending
 on the
% nature of the signal, maybe analyzing it in a frequency other than
 Hertz
% to see everything clearer before switching back to Hertz for the
 filters
% can be useful.
```



Published with MATLAB® R2019a