

*Ask your question, you will be an idiot for a second.
Do not ask and you will be an idiot all your life. Albert Einstein.*

Fourier analysis of continuous-time signals

Exercise 1.1 - Analysis of a signal in the time-domain

Consider the continuous-time signal plotted in Figure 1.1 where the ... indicates that the pattern continues outside the interval shown.

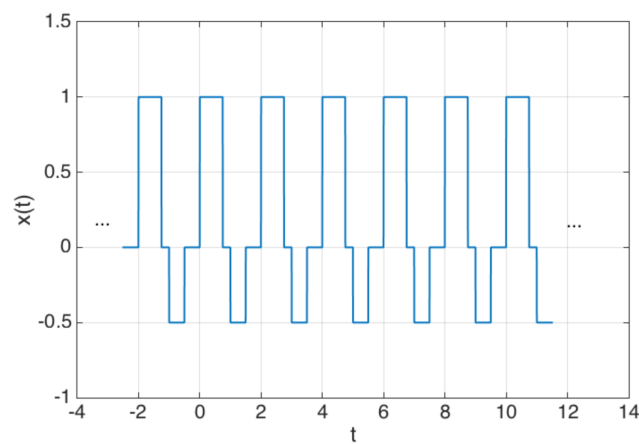


Figure 1.1: Signal evolution in the time-domain

Select all the correct statements.

- This is a discrete-time signal
- This is a continuous-time signal
- This is a causal signal
- This is a non-causal signal
- This is a non periodic signal
- This is a periodic signal with fundamental period 1 second
- This is a periodic signal with fundamental period 2 second
- This is a periodic signal with fundamental frequency 0.5 Hz
- This is a periodic signal with fundamental frequency 0.25 Hz
- This is a signal with zero-mean value

Exercise 1.2 - Analysis of a signal in the frequency-domain

Consider the amplitude and phase spectrum of a signal plotted in Figure 1.2.



Figure 1.2: Amplitude and phase spectrum of a signal

Select all the correct statements.

- The original signal is a discrete-time signal
- The original signal is a continuous-time signal
- The original signal is a non periodic signal
- The original signal is periodic with fundamental period 0.1 second
- The original signal is periodic with fundamental period 10 second
- The original signal has a zero-mean value
- The original signal has the following description $s(t) = \sin(20\pi t)$
- The original signal has the following description $s(t) = \cos(20\pi t)$
- The original signal has the following description $s(t) = \frac{1}{2} \sin(20\pi t)$
- The original signal has the following description $s(t) = \frac{1}{2} \cos(20\pi t)$
- The original signal has the following description $s(t) = \frac{1}{2} \sin(20\pi t - \frac{\pi}{2})$
- The original signal has the following description $s(t) = \frac{1}{2} \cos(20\pi t - \frac{\pi}{2})$

Exercise 1.3 - Spectrum of a continuous-time signal

Consider the continuous-time signal $x(t)$ plotted in Figure 1.3.

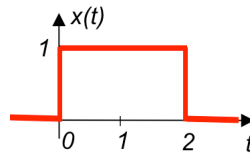


Figure 1.3: Continuous-time signal

1. Recall its usual name.
2. Give a mathematical definition of $x(t)$ by using a usual continuous-time signal.
3. Determine its continuous-time Fourier transform $X(f)$ and show it is given as:

$$X(f) = \frac{1 - e^{-j4\pi f}}{j2\pi f}$$

4. Show that $X(f)$ can be expressed as:

$$X(f) = 2\text{sinc}(2f)e^{-j2\pi f}$$

where

$$\text{sinc}(\alpha) = \frac{\sin(\pi\alpha)}{\pi\alpha} \tag{1}$$

is the cardinal sine function.

5. Determine its amplitude spectrum $|X(f)|$ and phase spectrum $\varphi(f) = \text{Arg}(X(f))$.
6. Plot its amplitude spectrum $|X(f)|$.

From now on, we will make use of Matlab which will be central to the development of the chosen learn-by-doing approach. The course has been indeed designed to follow a structured learning path, with examples using Matlab being an integral part.

If you need a refresher about the Matlab programming language, it is recommended that you take the free self-paced tutorial **Matlab Onramp** which can be assessed and run simply from any browser at:

<https://fr.mathworks.com/learn/tutorials/matlab-onramp.html>

To illustrate the use of Matlab, the following code generates the amplitude and phase spectra determined in the previous exercise. Copy and paste the code in Matlab to reproduce the spectra displayed in Figure 1.4.

```
f=linspace(-3,3,10000);
X=2*sinc(2*f).*exp(-j*2*pi*f);
subplot(1,2,1)
plot(f,abs(X),'-r','linewidth',3)
ylabel('Amplitude of X(f)')
xlabel('Frequency (Hz)')
grid
set(gca,'FontSize',14,'FontName','helvetica');
subplot(1,2,2)
plot(f,180/pi*angle(X),'-r','linewidth',3)
ylabel('Phase of X(f) in degree')
xlabel('Frequency (Hz)')
grid
set(gca,'FontSize',14,'FontName','helvetica');
```

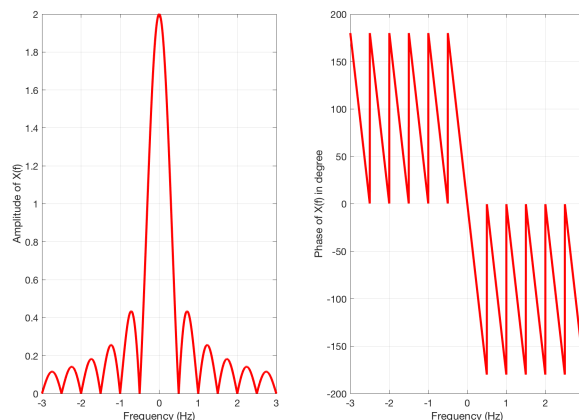


Figure 1.4: Amplitude and phase spectra of the delayed rectangular window $x(t)$ represented in Figure 1.3.

Exercise 1.4 - Signal Processing Onramp

Start the free self-paced tutorial entitled **Signal Processing Onramp** that provides an interactive introduction to practical signal processing methods for spectral analysis available at:

<https://fr.mathworks.com/learn/tutorials/signal-processing-onramp.html>

It includes videos and hands-on exercises with automated assessments and feedback. Topics covered in Signal Processing Onramp include preprocessing signals, spectral analysis, filtering, and signal measurements. This tutorial will take about two-hours to complete but you can leave anytime and come back later. Do not forget to bring your earphone to listen to the short videos.

It is mandatory to use your email address from the University of Lorraine to login.

Note that you will be asked to send the **Course Completion Certificate** to the instructor when you have finished the self-paced tutorial Signal Processing Onramp.

The deadline to send your certificate is: **December 15, 2022 at 6:00 pm.**

*Ask your question, you will be an idiot for a second.
Do not ask and you will be an idiot all your life. Albert Einstein.*

Sampling theorem & Fourier analysis of discrete-time signals

Exercise 2.1 - Sampling theorem

The highest frequency in a continuous-time signal is 20 kHz. Which of the following sampling frequencies will prevent aliasing when sampling this signal ?

Select all that apply.

- 11 kHz
- 21 kHz
- 41 kHz
- 61 kHz
- 101 kHz

Exercise 2.2 - Choice of the sampling frequency

We want to record two sounds, the first being low-pitched with a frequency of $f_1 = 100$ Hz, the second being high-pitched with a frequency of $f_2 = 10$ kHz.

1. A non-specialist in digital signal processing suggests to record the data with the sampling frequency of 10 kHz. Is this choice appropriate?
2. If this is not the case, propose a sampling frequency value for recording the signal. Justify your answer.

Exercise 2.3 - Spectrum of sampled signal

Suppose that an analog signal is given as

$$x(t) = 5 \cos(2\pi 1000t)$$

and is sampled at the rate 8 kHz.

1. Sketch the spectrum $X(f)$ of the analog signal.
2. Sketch the spectrum $X_s(f)$ of the sampled signal from -20 kHz to 20 kHz.

Exercise 2.4 - Reconstruction of the original signal

Assume that an analog signal is given by

$$x(t) = 5 \cos(2\pi 2000t) + 2 \cos(2\pi 3000t)$$

and is sampled at the rate of 8 kHz.

1. Sketch the spectrum $X_s(f)$ of the sampled signal from -16 kHz to 16 kHz.
2. Sketch the recovered analog signal spectrum $Y(f)$ if an ideal lowpass filter with a cutoff frequency of 4 kHz is used to filter the sampled signal to recover the original signal.

Exercise 2.5 - Aliasing effect

The following code generates 2 seconds of a low-pitched sine wave with a frequency of $f_0=200$ Hz.

```
f0 = 200; % low-pitched sound frequency in Hz
fs = 2000; % sampling frequency in Hz
Ts = 1/fs;
tmax = 2;
t = 0:Ts:tmax; % discrete-time points
x = sin(2*pi*f0*t);
figure
plot(t(1:50),x(1:50),'o',t(1:50),x(1:50))
xlabel('Time (sec)')
sound(x,fs);
```

1. Copy and paste the code in Matlab to generate the sine wave and listen to the sound produced.
2. Determine from the code above the sampling frequency. Is this choice appropriate?
3. Increase the frequency f_0 of the sine wave to 1800 Hz and re-execute the code. Why does the sound produced appear to be identical as before ? Explain this so-called aliasing effect by using a graphical analysis in the frequency domain ?
4. Choose an appropriate sampling frequency for the sine wave of frequency 1800 Hz and listen to the sound produced. Can you hear the difference with the low-pitched sine wave of 200 Hz ?

Exercise 2.6 - Energy and average power of discrete-time signals

Consider the following finite support signal:

$$x(k) = \begin{cases} (-1)^k k & \text{for } k = 1, 2, 3 \\ 0 & \text{otherwise} \end{cases}$$

Consider also its periodic version:

$$y(k) = \sum_{l=-\infty}^{+\infty} x(k + 7l)$$

1. Sketch $x(k)$ and $y(k)$.
2. Compute the energy of $x(k)$ and $y(k)$.
3. Compute the average power of $x(k)$ and $y(k)$.

Exercise 2.7 - Discrete-time Fourier transform (DTFT)

Compute the DTFT of the following discrete-time signals. In each case, plot the signal in the time-domain and express their magnitude and phase spectra.

- (a) $x_1(k) = \delta(k)$
- (b) $x_2(k) = \delta(k - k_0)$
- (c) $x_3(k) = (\alpha)^k u(k)$ for $0 < \alpha < 1$

Properties of geometric series recalled in the Appendix can be useful to compute the spectrum of $x_3(k)$.

Exercise 2.8 - Discrete-time Fourier transform (DTFT)

Consider the following discrete-time non-periodic signal defined as:

$$x(k) = \delta(k) + \delta(k - 1)$$

1. Plot $x(k)$ for $k \in [-2; 8]$.
2. Determine its DTFT $X(f)$.

3. Show that $X(f)$ can be written as:

$$X(f) = 2 \cos\left(\pi \frac{f}{f_s}\right) e^{-j\pi \frac{f}{f_s}}$$

4. Determine and plot the magnitude and phase spectra limited to the range $f \in [-f_s; f_s]$.

5. Conclude about the main properties of the spectra of discrete-time signals computed by DTFT.

*Ask your question, you will be an idiot for a second.
Do not ask and you will be an idiot all your life. Albert Einstein.*

Fourier analysis of discrete-time signals and the FFT algorithm

Exercise 3.1 - Inverse discrete-time Fourier transform (inverse DTFT)

Consider the following DTFT spectrum of a discrete-time signal:

$$X(f) = \begin{cases} 1 & \text{if } |f| \leq \frac{f_s}{4} \\ 0 & \text{if } |f| > \frac{f_s}{4} \end{cases} \quad X(f + lf_s) = X(f) \quad l \in \mathbb{Z}$$

1. Plot $X(f)$.
2. Determine the corresponding the time-domain signal $x(k)$ by inverse DTFT.
3. Plot $x(k)$.

Exercise 3.2 - Discrete Fourier transform (DFT)

Consider the following periodic discrete-time signal defined as:

$$x(k) = \delta(k) + \delta(k - 1), \quad \text{and} \quad x(k + 4l) = x(k), l \in \mathbb{Z}$$

1. Plot $x(k)$ for $k \in [-4; 3]$.
2. Determine its DFT coefficients $X(m)$ for $m = 0, 1, 2, 3$.
3. Determine and plot the magnitude and phase spectra for $m \in [-4; 3]$.
4. There is an algorithm which implements in a numerically efficient way the calculation of the DFT coefficients. The algorithm is called Fast Fourier Transform (FFT) and its name in Matlab is `fft`. Use this function to compute the DFT coefficients and verify your calculation.
5. Plot in Matlab by using the `subplot` function the magnitude and phase spectra for $m = 0$ to 3. You can use the function `stem` instead of `plot` to highlight the sampled nature of the two spectra.

Exercise 3.3 - DFT and FFT

Direct computation of N DFT coefficients from N samples of a signal requires $O(N^2)$ operations. The FFT algorithm computes all DFT coefficients with $O(N \log_2(N))$ operations when N is a power of 2.

1. Complete the table below.

N	10^3	10^6	10^9
N^2			
$N \log_2(N)$			

2. As observed from the completed table, the FFT achieves a very large reduction in the cost of computation as the number of data N becomes very large. To get a better idea of the advantages of the FFT algorithm, assume that it takes 1 ns to execute one operation when of $N = 10^9$. Determine the approximated time required by the computation of the DFT and the FFT implementation. Give the time required by the DFT in years.

Exercise 3.4 - Denoising signal by FFT

The goal is to illustrate, with a simple example, the use of the FFT and inverse FFT to denoise a signal. Consider a continuous-time noise-free signal

$$x_0(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t)$$

with frequencies $f_1 = 50$ and $f_2 = 120$ Hz.

A large amount of Gaussian white noise is added to this signal, as shown in the top panel of Figure 3.1.

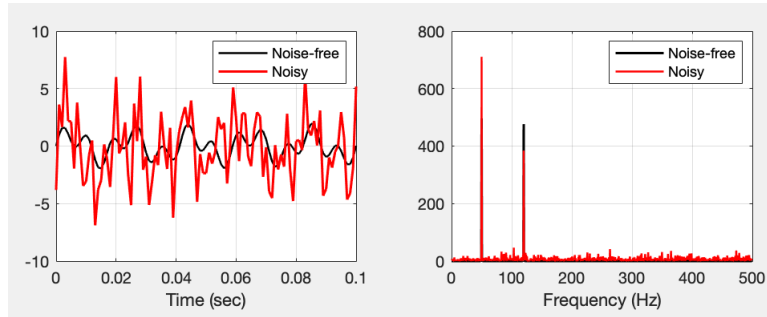


Figure 3.1: Noise-free and noisy signals and their power spectral density

It is possible to compute the fast Fourier transform of this noisy signal x using the `fft` command.

The power spectral density (PSD) is the normalized squared magnitude of $X(f)$, and indicates how much power the signal contains in each frequency. In the top right plot of Figure 3.2, it is clear that the PSD of the noisy signal contains two large peaks at 50 Hz and 120 Hz. It is possible to zero out spectral components that have power below a threshold (for example 100 here) to remove noise from the signal. After inverse transforming the denoised spectrum, we find the original noise-free and denoised signals (by using the `ifft` command) match quite well (Fig. 3.2, bottom-left).

Download the Matlab code from the course website to reproduce the plots of Figure 3.2. Understand each step of this basic denoising method. Modify the threshold to see its effects.

Give a flow chart that summarizes the various steps that lead to the filtering/denoising of a noisy signal by FFT.

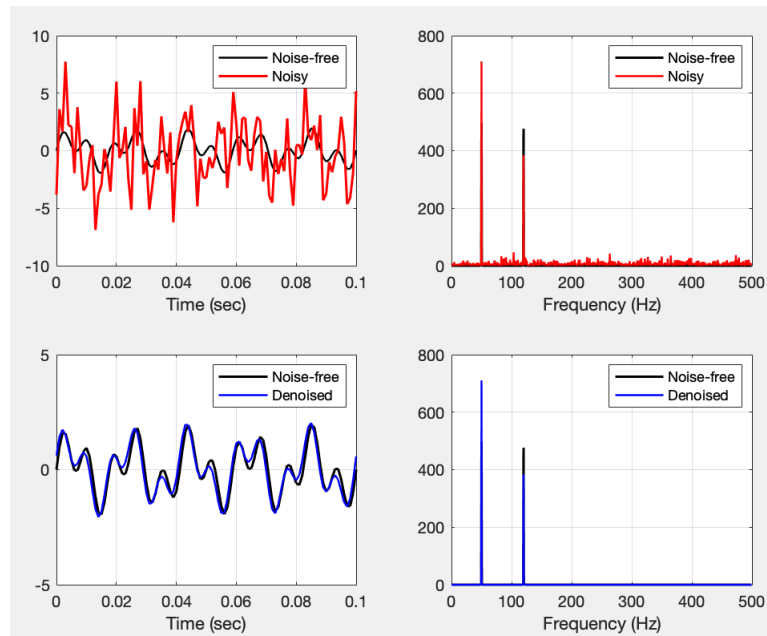


Figure 3.2: Denoising signal by FFT. (top-left) Noise is added to a simple signal given by a sum of two sine waves. (top-right) In the Fourier domain, dominant peaks may be selected and the noise filtered. (bottom-right) spectral components that have power below a threshold (for example 100 here) are zero out. (bottom-left) The denoised signal is obtained by inverse FFT recovering the two dominant peak sine wave

Exercise 3.5 - Approximation of DTFT spectra by the FFT - Importance of zero padding

Consider the following infinite non-periodic discrete-time signal

$$x(k) = \begin{cases} 0 & \text{if } k < 0 \\ 1 & \text{if } 0 \leq k < L \\ 0 & \text{if } k \geq L \end{cases}$$

with L an integer greater than 0.

1. Plot $x(k)$.
2. Determine its DTFT $X(f)$ and show that it can be expressed as (*properties of geometric series given in the appendix can be useful*)

$$X(f) = \frac{1 - e^{-j2\pi f L T_s}}{1 - e^{-j2\pi f T_s}}$$

We will assume is the following a sampling frequency $f_s = 1000$ Hz.

3. We want to visualize the magnitude of $X(f)$ using Matlab. However, Matlab cannot handle continuous sequences as $X(f)$. We need therefore to compute $|X(f)|$ for a finite number of frequencies. Set $L = 20$ and using Matlab, plot 10000 points of $|X(f)|$ (abs in Matlab) over one period (from 0 to f_s). You can use/adapt the Matlab code given below.

```
clear; close all; clc
M=10000;
n=0:M-1;
fs=1000;
Df=fs/M;
f=n*Df;
w=2*pi*f/fs;
L=20;
X=((1-exp(-j*L*w))./(1-exp(-j*w)));
plot(f,abs(X),'linewidth',2)
xlabel('Frequency (Hz)')
ylabel('Magnitude of the DTFT of X(f)')
title('Theoretical DTFT magnitude spectra plotted over one period')
grid
set(gca,'FontSize',14,'FontName','helvetica');
```

4. The DTFT is mainly a theoretical analysis tool. In practical situations, the DFT is used to approximate the spectrum of the discrete-time signals. Recall that in Matlab we use the FFT, which is an efficient algorithm to compute the DFT. Generate a finite sequence $x_1(k)$ of length $N = 30$ such that

$$x_1(k) = x(k) \text{ for } k = 0, \dots, N - 1$$

5. Compute its DFT (via the FFT) $X_1(m)$ for $m = 0, \dots, N - 1$ and plot its magnitude. Compare it with the theoretical magnitude plot obtained by computing the DTFT. You can use/adapt the Matlab code given below.

```
N=30;
x1=[ones(1,L),zeros(1,N-L)];
X1=fft(x1,N);
df=fs/N;
f1=(0:N-1)*df;
figure
plot(f,abs(X),'-',f1,abs(X1),'-o','linewidth',2)
legend('DTFT','FFT for N=30')
xlabel('Frequency (Hz)')
ylabel('Magnitude of the FFT of X(m)')
title('DFT of the rectangular window plotted over one period for N=30')
grid
set(gca,'FontSize',14,'FontName','helvetica');
```

6. Repeat the last two previous questions for different values when $N = 64, 128, 1024$. What can you observe ?

Adding zero to the end of a time-domain signal to increase its length is known as **zero-padding**. There are a few reasons why you might want to zero pad time-domain data. The most common reason is to make a signal have a power-of-two number of samples. When the time-domain length of a signal is a power of two, radix-2 FFT algorithms, which are extremely efficient, can be used to speed up processing time. Zero-padding will also improve the "FFT resolution" given by

$$\Delta f = \frac{f_s}{N}$$

which defines the frequency distance between two points of the spectrum. As seen from the formula, the FFT resolution is inversely proportional to the number of points, denoted as N here, used in the FFT. This explains why the fit between the theoretical and approximated spectra is better when N increases.

*Ask your question, you will be an idiot for a second.
Do not ask and you will be an idiot all your life. Albert Einstein.*

Analysis and design of digital filters

Exercise 4.1. FIR/IIR and causal/non causal digital filters

State whether the filters described below are FIR or IIR and causal or non-causal. Motivate your answers:

(a) $y(k) = \frac{1}{2}y(k-1) + \frac{1}{2}e(k+1)$

(b) $y(k) = \frac{1}{2}e(k) + \frac{1}{2}e(k-1)$

(c) Determine the transfer function $H(z)$ of each filter.

Exercise 4.2 - Digital filter analysis from its frequency response

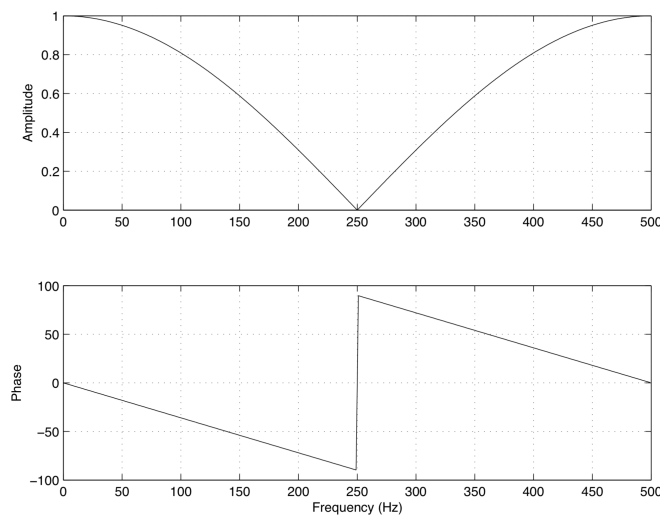


Figure 4.1: Magnitude and phase frequency responses

The magnitude and phase frequency responses for a digital filter in the interval $0 \leq f \leq f_s/2$ are displayed in Figure 4.1. From these plots, determine, with justification,

- the sampling frequency in Hz.
- whether it is an FIR or IIR filter.
- the type of filtering (low-pass, high-pass, band-pass or stop-band).
- the cut-off frequencies in Hz.

Exercise 4.3 - Digital filter analysis from its difference equation

Consider the digital filter described by the following difference equation:

$$y(k) = \frac{1}{2}e(k) - \frac{1}{2}e(k-1)$$

- Determine the type of filter (FIR/IIR, non recursive/recursive) ? Justify your answer.

- (b) Derive its transfer function $H(z)$.
- (c) Determine and sketch the impulse response.
- (d) Determine and plot the zeros and poles. Conclude about the stability of the filter. Explain your answer.
- (e) Determine the frequency response $H(f)$.
- (f) Compute the magnitude and argument of $H(f)$ and sketch the magnitude and phase frequency responses in the interval $0 \leq f \leq \frac{f_s}{2}$, where $f_s = 1000$ is the sampling frequency in Hz.
- (g) Determine the type of filtering (low-pass, high-pass, etc) and give the cut-off frequency (in Hz) of the digital filter.
- (h) Check your plots by using Matlab and the code given below.

```

B=[0.5 -0.5];
A=[1 0];
fs=1000;
[H,f]=freqz(B,A,512,fs);
subplot(2,1,1)
plot(f,abs(H)),grid
subplot(2,1,2)
plot(f,angle(H)*180/pi),grid
xlabel('frequency (Hz)')

```

Exercise 4.4 - Digital filter analysis from its difference equation

Consider a digital filter given by the following difference equation

$$y(k) = \lambda y(k-1) + (1-\lambda)e(k)$$

where λ is a real constant.

- (a) Determine the type of filter (FIR or IIR) ? Explain your answer.
- (b) Derive the filter transfer function, $H(z)$, from the difference equation.
- (c) Determine and plot the zeros and poles. Give the constraints on λ in order for the filter to be stable. We will assume $\lambda = 0.5$ in the following.
- (d) Determine and sketch the impulse response.
- (e) Compute and sketch the magnitude and phase frequency responses in the interval $0 \leq f \leq \frac{f_s}{2}$, where $f_s = 1000$ is the sampling frequency in Hz.
- (f) Determine the type of filtering (low-pass, high-pass, etc) and give the cut-off frequency (in Hz) of the digital filter.
- (g) Check your plots by using Matlab (modify the A and B variables in the code given above).

Exercise 4.5. Frequency response and Bode diagram from the filter specifications

It is required to design a bandpass digital filter to meet the following specifications

Passband	150 – 250Hz
Transition width	50 Hz
Passband ripple	0.1 dB
Stopband attenuation	60 dB
Sampling frequency	1 kHz

- (a) Specify the cut-off frequencies (centered in the transition bands), low and high bandpass and stopband frequencies from the given specifications.
- (b) Give the value of the desired stopband ripple (A_s in dB) along with the desired passband ripple (A_p in dB). Determine from A_s and A_p the value of δ_s and δ_p .

- (c) Plot the desired magnitude frequency response and the magnitude Bode diagram of the digital filter according to the specifications.
- (d) Use the Matlab FilterDesigner App (see Figure 4.2) to determine the FIR digital filter of order 60 designed by the window method with a blackman window. Verify that the filter design fulfills the 60 dB stopband attenuation specifications. If needed, the designed filter can be exported in the Matlab workspace for further use (tab file then export).
- (e) Use the FilterDesigner App to determine the minimum order IIR digital filter designed by the bilinear method that fulfills the required specifications. Compare the computed minimum IIR filter order with the order of the FIR filter.

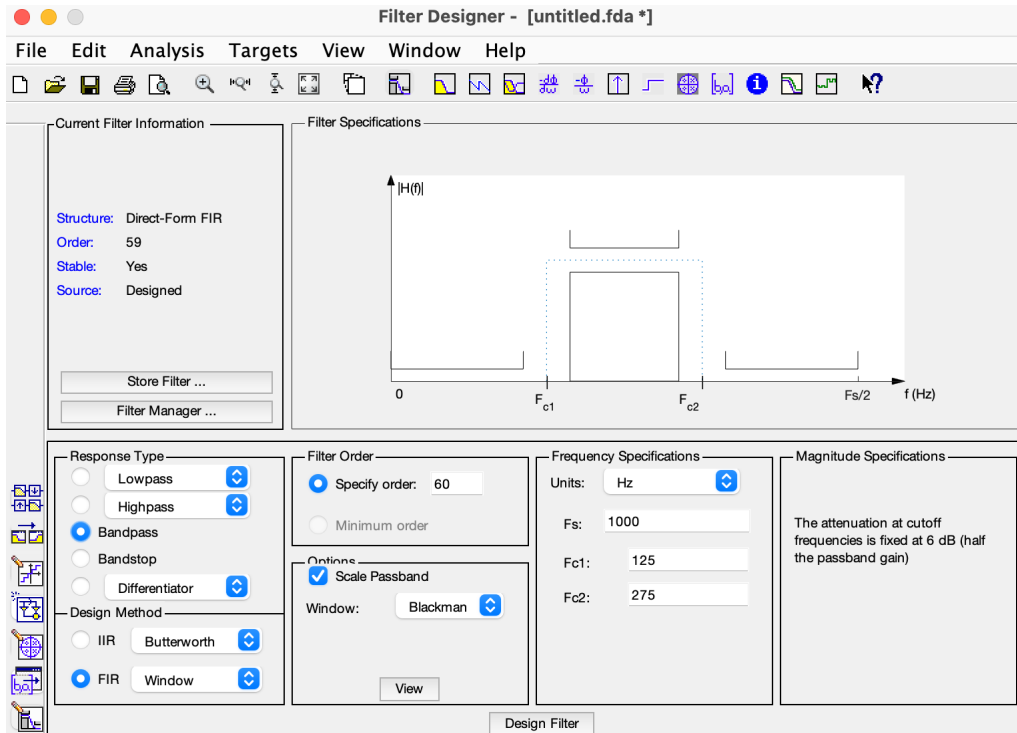


Figure 4.2: The Matlab FilterDesigner App

Appendix

Continuous-time Fourier transform (CTFT)

The CTFT is defined as:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt$$

The inverse CTFT is defined as:

$$x(t) = \int_{-\infty}^{+\infty} X(f)e^{j2\pi ft} df$$

Some useful properties of the Fourier transform

$$F(ax(t) + by(t)) = aX(f) + bY(f)$$

$$F(x(t - t_0)) = e^{-j2\pi ft_0} X(f)$$

$$F(e^{j2\pi f_0 t} x(t)) = X(f - f_0)$$

$$F(x(at)) = \frac{1}{|a|} X\left(\frac{f}{a}\right)$$

$$F(x(t) * y(t)) = X(f) \times Y(f)$$

$$F(x(t) \times y(t)) = X(f) * Y(f)$$

Discrete-time Fourier transform (DTFT)

The DTFT is defined as:

$$X(f) = \sum_{k=-\infty}^{+\infty} x(k)e^{-j2\pi f k T_s}$$

The inverse DTFT is defined as:

$$x(k) = \int_{-f_s/2}^{+f_s/2} X(f)e^{j2\pi f k T_s} df$$

Discrete Fourier Transform (DFT)

The DFT is defined as (set $N = K_0$ if the discrete-time signal is periodic of period K_0):

$$X(m) = \sum_{k=0}^{N-1} x(k)e^{-j2\pi \frac{km}{N}} \text{ for } m = 0, 1, \dots, N - 1.$$

The inverse DFT is defined as:

$$x(k) = \frac{1}{N} \sum_{m=0}^{N-1} X(m)e^{j2\pi \frac{km}{N}} \text{ for } k = 0, 1, \dots, N - 1.$$

Useful properties of geometric series

finite sum of geometric series	$\sum_{k=0}^N q^k = N + 1$	if $q = 1$
--------------------------------	----------------------------	------------

	$\sum_{k=0}^N q^k = \frac{1 - q^{N+1}}{1 - q}$	if $q \neq 1$
--	--	---------------

infinite sum of geometric series	$\lim_{N \rightarrow +\infty} \sum_{k=0}^N q^k = \frac{1}{1 - q}$	if $ q < 1$
----------------------------------	---	--------------

Table of common unilateral z -transforms

The table below provides a number of z -transforms and their region of convergence (ROC).

Signal	z -transform	ROC
$\delta(k)$	1	All z
$\delta(k - i)$	z^{-i}	$z \neq 0$
$u(k)$	$\frac{z}{z-1}$	$ z > 1$
$ku(k)$	$\frac{z}{(z-1)^2}$	$ z > 1$
$k^2u(k)$	$\frac{z(z+1)}{(z-1)^3}$	$ z > 1$
$a^k u(k)$	$\frac{z}{z-a}$	$ z > a $
$ka^k u(k)$	$\frac{az}{(z-a)^2}$	$ z > a $
$k^2 a^k u(k)$	$\frac{az(z+a)}{(z-a)^3}$	$ z > a $
$\cos(\omega_0 k)u(k)$	$\frac{z(z-\cos(\omega_0))}{z^2-2\cos(\omega_0)z+1}$	$ z > 1$
$\sin(\omega_0 k)u(k)$	$\frac{\sin(\omega_0)z}{z^2-2\cos(\omega_0)z+1}$	$ z > 1$

Remarks:

- The notation for z found in the table above may differ from that found in other tables. For example, the basic z -transform of a unit discrete-time step $u(k)$ can be written as either of the following two expressions, which are equivalent:

$$Z(u(k)) = U(z) = \frac{z}{z-1} = \frac{1}{1-z^{-1}}$$

- The ROC for a given sequence $x(k)$, is defined as the range of z for which the z -transform converges. Since the z -transform is a power series, it converges when $x(k)z^{-k}$ is absolutely summable. Stated differently,

$$\sum_{k=0}^{+\infty} |x(k)z^{-k}| < \infty \quad \text{must be satisfied for convergence.}$$

Property 1. if $x(k)$ is of finite duration, then the ROC is the entire z -plane, except possibly $z = 0$ and/or $z = \infty$.

Property 2. The ROC does not contain any poles.

Useful properties of the unilateral z -transform

Some useful properties which have found practical use in signal processing are summarized below.

Property	signal	z -transform
linearity	$ax(k) + by(k)$	$aX(z) + bY(z)$
delays (or shifts)	$x(k - 1)$	$z^{-1}X(z)$
	$x(k - i)$	$z^{-i}X(z)$
convolution	$y(k) = h(k) * u(k)$ $y(k) = \sum_{i=-\infty}^{+\infty} h(i)u(k - i)$	$Y(z) = H(z)U(z)$
differentiation	$kx(k)$	$-z \frac{dX(z)}{dz}$
accumulation	$\sum_{k=0}^n x(k)$	$\frac{1}{1-z^{-1}}X(z)$
initial value theorem	if $x(k) = 0$ for $k < 0$	$x(0) = \lim_{z \rightarrow +\infty} X(z)$
final value theorem	$\lim_{k \rightarrow +\infty} x(k) = \lim_{z \rightarrow 1} (z - 1)X(z)$	if the limit exists