TD TNS
n°1

H. Garnier
M. Jha

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.

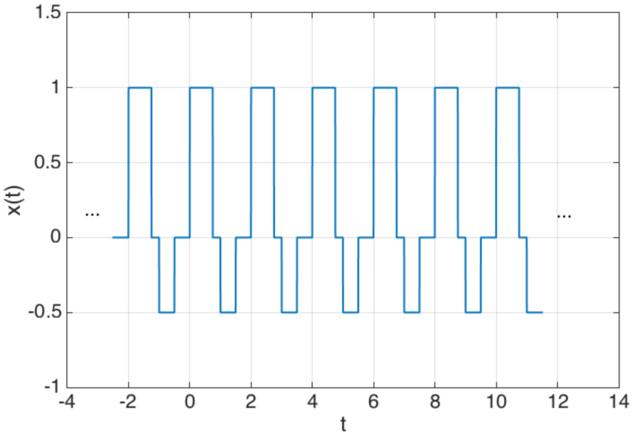

Figure 1.1: Signal evolution in the time-domain

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
- ☐ This is a signal with finite energy
- ☐ This is a signal with infinite energy
- ☐ This is a signal with finite average power
- ☐ This is a signal with infinite average power

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.

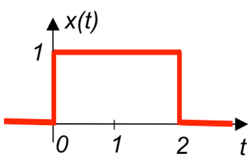

Figure 1.3: Continuous-time signal

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 $X(f)$.
4. Express $X(f)$ in terms of the sine cardinal function which is defined as

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

5. Determine its amplitude spectrum $|X(f)|$ and phase spectrum $\varphi(f)$.
6. Plot its amplitude spectrum $|X(f)|$.

Note that the amplitude and phase spectra can be easily obtained from the Matlab code given below:

```
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');
```

The resulting spectra are displayed in Figure 1.4.

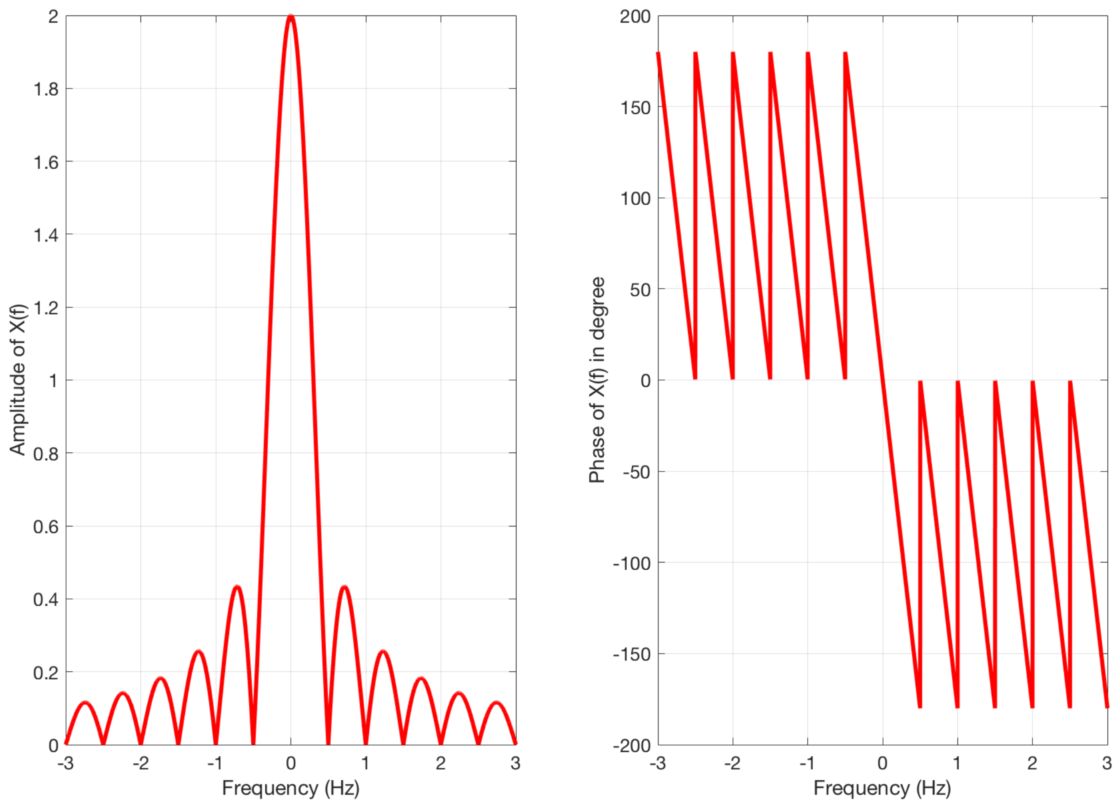

Figure 1.4: Amplitude and phase spectra of the delayed rectangular window $x(t)$

Exercise 1.4 - Ideal sampling and spectrum of a sinusoidal signal
Consider a continuous-time sinusoidal signal defined as

$$x(t) = \cos(2\pi f_0 t)$$

with $f_0 = 10$ kHz.

1. Recall $X(f)$, the spectrum of $x(t)$ obtained by using the continuous-time Fourier transform.
2. Plot $X(f)$.
3. Remind the conditions for the choice of the sampling frequency f_s to (ideally) sample the signal $x(t)$ without any loss of information. The resulting signal will be denoted as $x_s(t)$.
4. By using a graphical reasoning and from the properties of the Fourier transform, plot between $-f_s$ et f_s, one below the other, $X(f)$, $\delta_{f_s}(f)$ and $X_s(f)$ the spectrum of $x(t)$, $\delta_{T_s}(t)$ (Dirac comb) and $x_s(t)$ respectively when $f_s = 30$ kHz.
5. From the ideally sampled signal, a continuous-time signal is reconstructed by applying an ideal brick-wall low-pass filter of cut-off frequency $f_s/2$. By using a graphical reasoning, plot, one below the other, the frequency response of the ideal low-pass filter $H(f)$ and the spectrum $X_r(f)$ of the reconstructed signal $x_r(t)$.
6. From the obtained spectrum $X_r(f)$, the reconstructed signal is given in the time-domain by

   $$x_r(t) = \cos(2\pi f_r t)$$

   Determine the value of the fundamental frequency f_r from the frequency-domain analysis.
7. Repeat questions 4 to 6 above when $f_s = 15$ kHz.

H. Garnier
M. Jha

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 - 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 2.4 - 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. Properties of geometric series given in the appendix can be useful to compute the spectrum of $x_3(k)$.

(a) $x_1(k) = \delta(k)$

(b) $x_2(k) = \delta(k - k_0)$

(c) $x_3(k) = (\alpha)^k u(k)$ for $0 < \alpha < 1$

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 - 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:

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

4. Determine and plot the magnitude and phase spectra for $f \in [-\frac{f_s}{2}; \frac{f_s}{2}]$.

Exercise 3.2 - Discrete Fourier transform (DFT)
Consider the following periodic discrete-time signal defined as:

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

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 - 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

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

5. Compute its DFT (via the FFT) $X_1(m)$ for $m = 0, \ldots, N-1$ 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.

TD TNS
n°4
H. Garnier
M. Jha

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 practical implementation of digital filters

Exercise 4.1 - Digital filter analysis from its frequency response

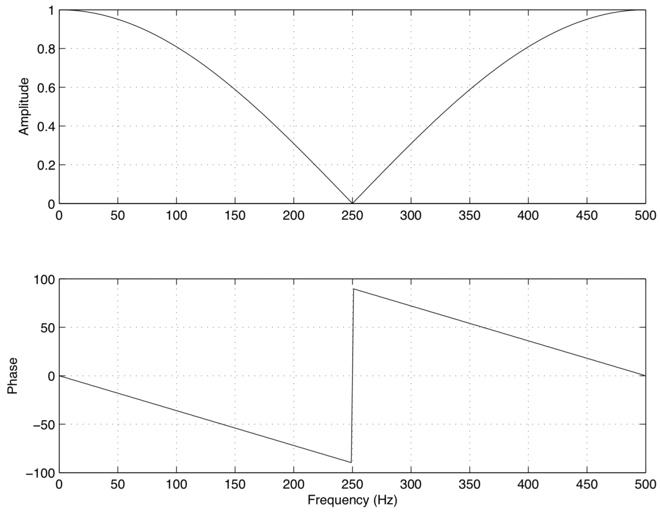

Figure 4.1: Magnitude and phase frequency responses

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,

(a) the sampling frequency in Hz.

(b) whether it is an FIR or IIR filter.

(c) the type of filtering (low-pass, high-pass, band-pass or stop-band).

(c) the cut-off frequencies in Hz.

Exercise 4.2 - Digital filter analysis

Consider the discrete-time filter described by the following difference equation:

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

(a) Determine the type of filter (FIR or IIR) ? 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. Justify your answer.

(e) Determine the frequency response $H(f)$.

(f) Compute magnitude and phase and sketch the magnitude and phase frequency responses in the interval $0 \leq f \leq \frac{f_s}{2}$, where f_s 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 discrete-time filter.

Exercise 4.3 - Digital filter analysis
Consider a digital filter given by the following difference equation

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

where a is a real constant.

(a) Determine the type of filter (FIR or IIR) ? Justify your answer.

(b) Derive the filter transfer function, $H(z)$, from the difference equation. Give the constraints on λ in order for the filter to be stable. We will assume λ within the stability range in the following.

(c) Determine and sketch the impulse response.

(d) Compute and sketch the magnitude and phase frequency responses in the interval $0 \leq f \leq \frac{f_s}{2}$, where f_s is the sampling frequency in Hz.

(e) Determine the type of filtering (low-pass, high-pass, etc) and give the cut-off frequency (in Hz) of the discrete-time filter.

Exercise 4.4 - Digital filter implementation - Smoothing of temperature anomalies
Consider the simple moving average (MA) filter described by the following difference equation:

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

(a) Determine and plot the first five step response samples.

(b) Write a Matlab program that computes the N step response samples of the MA filter described by its difference equation. Test it for $N = 5$ and compare the step response samples obtained with those from your hand calculation.

(c) Plot the step response computed with your implementation and compare it to the response given by the Matlab function (the MA filter is below defined from its numerator `[1/2 1/2]` and denominator `[1 0]` polynomial coefficients in descending power)
`ydstep=dstep([1/2 1/2],[1 0],5);`
Use the Matlab function `stem` to display the step response samples.

(d) Use your program to compute and sketch the MA filter response to a unit square wave input defined as

```
tk=0:1:2/0.1;
u=square(2*pi*0.1*tk);
```

(e) Use the Matlab function `filter` to compute the response to the square wave
`yfilter=filter([1/2 1/2],[1 0],u);`
Compare it to the response of your MA filter implementation program. The two responses should be identical.

(f) Download the temperature anomalies from the website. Compute and plot the smoothed signal obtained when the MA filter is applied.

(g) Use now the Matlab function `filtfilt` to compute the smoothed version of the temperature anomalies and compare it to the response previously obtained by using `filter`.

(h) Repeat the previous question for a MA filter of length $N = 10$ to obtain a smoother version of the temperature anomalies.

TD TNS
n°5
H. Garnier
M. Jha

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.

Design of FIR and IIR digital filters

Exercice 5.1. Causal/non causal digital filters
State whether the filters described below are causal or non-causal. Justify your answer:

(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 of each filter.

Exercice 5.2. Frequency response and Bode diagram from the filter specifications
It is required to design a FIR bandpass digital filter to meet the following specifications

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

(a) Specify the cut-off frequencies, 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 magnitude frequency response and the magnitude Bode diagram of the desired digital filter according to the specifications.

(d) Use the Matlab App filterDesigner to determine the minimum order FIR digital filter designed by the window method with a kaiser window that fulfills the specifications. The designed filter can be exported in the Matlab workspace for further use (tab file then export).

Exercice 5.3. FIR filter design by the window method
We want to use the window method to design a linear-phase and causal FIR system with $N = 7$ coefficients to approximate an ideal lowpass filter whose cutoff frequency is $f_c = \frac{f_s}{10}$ when $f_s = 1$ Hz.

(a) The impulse response of an ideal low-pass FIR filter is recalled below:

$$h(k) = 2f_c \text{sinc}(2f_c k T_s) \quad \text{for } \frac{-N+1}{2} \leq k \leq \frac{N-1}{2}$$

Compute the 7 impulse response coefficients.

(b) Use the Matlab command
```
fs=1;
h=fir1(6,0.1/(fs/2),boxcar(7),'noscale')
```
to check your filter design.

(c) Give the difference equation of the causal FIR filter and plot its discrete-time impulse response by using the `stem` function.

(d) We will use, in the following, the scaling version of the FIR filter so that its gain in the passband is unity. Compute the FIR filter coefficients with the `fir` routine and plot the Bode diagram with the command `freqz(h,1,1024,1)`
From the magnitude plot, determine the value of the stopband ripple (A_s in dB).

Exercice 5.4. IIR filter design by the bilinear method
It is desired to determine, using the bilinear method, the transfer function and difference equation of the digital equivalent of the analogue resistance-capacitance (RC) filter.

(a) Remind the transfer function $H(s)$ of the analogue lowpass RC filter. Specify the filter order and its cutoff frequency f_c in terms of R and C.

(b) Using the bilinear method, determine the transfer function, $H(z)$, of the digital filter assuming a -3 dB cutoff frequency of 30Hz and a sampling frequency of 60πHz.

(c) Determine the order of the digital IIR filter and study its stability.

(d) Use the Matlab function `bilinear` to compute the transfer function of the equivalent IIR filter and compare it to your solution computed by hand.

(e) Plot and compare the Bode diagram of the analogue and digital filter.

(f) Express the transfer function in negative power of z and give the difference equation of the IIR digital filter.

Exercice 5.5. Notch filter design to cancel out a spurious tone in a trumpet sound
Load the data stored in a file named `spurious_trumpet.mat`. It contains audio data recorded from an actual trumpet playing note B but where an interfering or spurious signal is added to it.

Listen to the sound made by the spurious trumpet.
`soundsc(y,fs);`

(a) By using the signalAnalyzer App, determine the fundamental frequency of the spurious tone.

(b) The goal now is to design a notch filter to cancel out the spurious harmonic. Refer to the course slides to recall the transfer function $H(z)$ of a so-called notch filter.

(c) Matlab has no function dedicated to this type of notch filter. It is possible to directly define the coefficients of the transfer function $H(z)$ based on the following solution:

```
f0=...; %interference frequency to be deleted \Omega0=2*pi*f0/fs; \\
a=...; %coefficient influencing the selectiveness of the notch filter \\
num=[...]; %coefficients of the numerator of the notch filter in decreasing power of z\\
den=[...];  %coefficients of the denominator of the notch filter in decreasing power of z\\
figure\\
freqz(num,den,1024,fs); %Calculation of the Bode diagram for 1024 points \\
yf=filter(num,den,y); % filtering of the trumpet signal by the designed notch filter\\
soundsc(yf,fs);
```

(d) The effectiveness of the filter may also be observed using the signalAnalyzer App by superimposing the spectrum of the original and filtered signals. Optimise the settings of the notch filter to most effectively remove the disruptive harmonic and recover the original trumpet signal with the less distorsion possible.

TD TNS
n°6
H. Garnier
M. Jha

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.

Stochastic process and random signal modelling

It is assumed in the following exercises that the signals are sampled with a sampling frequency of f_s=1 Hz. Stochastic process are denoted in bold $\mathbf{y}(k)$ while stochastic signals or realizations are denoted as $y(k)$. A useful summary about stochastic process is given in the Appendix.

Exercise 6.1. Basic descriptions of a discrete-time random process

Consider the discrete-time stationary stochastic process $\mathbf{y}(k)$ that is generated from:

$$\mathbf{y}(k) = 10 + \mathbf{e}(k) \tag{1}$$

where $\mathbf{e}(k)$ is a (zero-mean) white noise of variance $\sigma_{\mathbf{e}}^2 = 4$. Two realizations of this stochastic process are displayed in Figure 6.1 for $0 \leq k \leq 4$

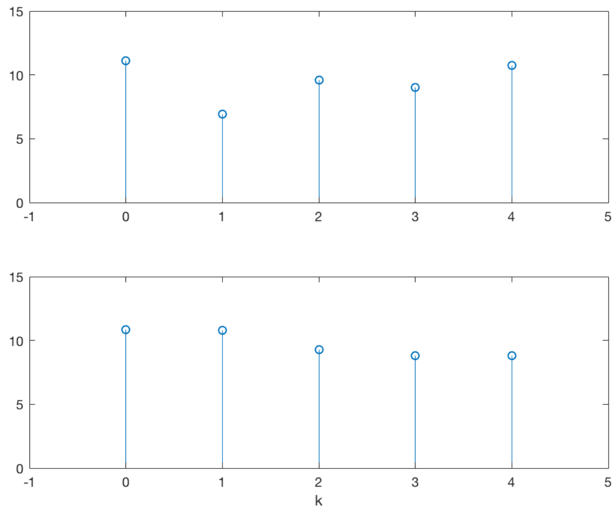

Figure 6.1: Two realizations of the stochastic process described by (1)

(a) Explain why the two realizations in Figure 6.1 are different.

(b) Determine the expected value $E\big(\mathbf{y}(k)\big)$ of $\mathbf{y}(k)$.

(c) Determine the variance $Var\big(\mathbf{y}(k)\big)$ of $\mathbf{y}(k)$ and its standard deviation $\sigma_{\mathbf{y}}$.

(d) Are the two realizations of the stochastic process plotted in Figure 6.1 consistent with your answers in (b) and (c) ?

(e) Determine the auto-correlation function $R_{\mathbf{y}}(\tau)$ of $\mathbf{y}(k)$ for every value of τ.

(f) Determine the power $P_{\mathbf{y}}$ of $\mathbf{y}(k)$.

Exercise 6.2. AR process

Consider the discrete-time stationary stochastic process $\mathbf{y}(k)$ that is generated from:

$$\mathbf{y}(k) + d_1\mathbf{y}(k-1) = \mathbf{e}(k)$$

where $\mathbf{e}(k)$ is a white noise of variance $\sigma_{\mathbf{e}}^2$ and $|d_1| < 1$.

a) Write the stochastic process in polynomial form as

$$\mathbf{y}(k) = \frac{1}{D(q^{-1})}\mathbf{e}(k)$$

where q^{-1} is the shift operator. Give the usual name for this stochastic process (AR, MA or ARMA), its order and its block-diagram representation.

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

c) Is this filter stable ? Justify your answer.

d) Determine the power spectrum $\Phi_{\mathbf{y}}(f)$ of $\mathbf{y}(k)$ obtained at the output of the filter.

Exercise 6.3. Least squares estimation of a trend model for the temperature anomalies

Given N samples of a stochastic process $\mathbf{y}(k)$ which has the form of

$$\mathbf{y}(k) = \alpha k + \beta + \mathbf{e}(k)$$

where α and β are the parameters of interest and $\mathbf{e}(k)$ is an unknown zero-mean noise sequence.

a) The cost function $V(\theta, Z^N)$ to be minimized to estimate the model parameter vector $\theta = [\alpha\ \beta]^T$ is chosen here to be the sum of the squares of the prediction error. Recall the expression of the cost function and the condition to determine the minimum of the cost function.

b) Derive the analytical solution of the least squares (LS) estimates for α and β.

c) Generate 10 samples of the stochastic realization from the following Matlab lines saved in a m. file

```
N=10;
alpha=0.5;
beta=-1;
e=randn(N,1);
k=(0:N-1)';
y=alpha*k+beta+e;
figure
plot(k,y,'o'),grid
```

Compute the LS estimates by using the analytical solution derived in question (a). Display on the same figure the simulated trend model output and the measured samples.

d) Use the LS method to estimate an affine function model for the temperature anomalies exploited in Exercise 4.4. Display on the same figure the simulated affine function model output and the measured samples.

e) Use the LS method to estimate a quadratic function model[1]. Display on the same figure the simulated quadratic function model output and the measured samples.

f) Use your estimated model to predict the temperature anomaly in 2050 and compare with predicted values found in Internet[2].

[1] For a recent study discussing trend models for climate time series, see https://www.sciencedirect.com/science/article/pii/S0012825218303726

[2] Some information about the impact of the temperature rise can be obtained in the following 2018 report from the GIEC (Groupe d'experts intergouvernemental sur l'évolution du climat): https://www.ipcc.ch/site/assets/uploads/sites/2/2019/09/IPCC-Special-Report-1.5-SPM_fr.pdf.

Exercise 6.4. Least squares estimation of AR process model parameters

Consider an AR stochastic process described by:

$$\mathcal{S}: \quad D(q^{-1})\mathbf{y}(k) = \mathbf{e}(k) \tag{2}$$

where q^{-1} represent the shift operator, $\mathbf{e}(k)$ is a white Gaussian noise of variance 1, and the polynomial $D(q^{-1})$ is given by:

$$D(q^{-1}) = 1.0 - 1.5q^{-1} + 0.7q^{-2}$$

a) Determine the transfer function of the filter. Compute the poles and check that the filter is stable.

b) Give the difference equation of the stochastic process.

c) Generate 500 samples of the AR process from the following Matlab lines saved in a m. file

```
N=500;
D=[1  -1.5  0.7];
e=randn(N,1);
y=filter(1,D,e);
figure
stem(y)
title('One realization of the AR process of order 2')
data=iddata(y,[]);
```

We assume that the order of the AR process is known. The goal is now to use the 500 samples of data to estimate the two AR model parameters by the least squares method.
The AR process model takes therefore the following form:

$$\mathcal{M}_{\text{AR}} : D(q^{-1}, \theta)y(k) = e(k)$$

with the parameter vector $\theta = [d_1\, d_2]^T$.

d) Recall the difference equation of the AR process model of order 2.

e) At time-instant k, rewrite the AR model in linear regression form.

f) From N samples of the random signal, formulate the parameter estimation problem in matrix form. Specify the dimension of each matrix.

g) Recall the solution given the least squares method (see slides of the lectures).

h) Implement the LS estimator in Matlab for the available set of data.

i) Run your program several times to find the trend in the parameter estimates. Comment the quality of the two estimates.

j) Matlab includes different algorithms dedicated to the parameter estimation of AR, MA or ARMA process.
The `ar` function can estimate the parameters of an AR process model by *least squares*. The command is the following :

```
nd=2;
Mar=ar(data,nd,'ls');
```

`nd` specifies the order of the polynomial $D(q^{-1})$, the input argument `'ls'` specifies the least squares method.
Use the command `present(Mar)` to visualize the estimated modes. The standard deviation of the parameter estimates are given in bracket.

k) Compare your estimated parameters with those given the Matlab `ar` function.

l) Estimate the power spectrum of the AR process using the estimated parameters

$$\hat{\Phi}_{\mathbf{y}}(f) = \frac{1}{\left|1 + \hat{d}_1 e^{-2j\pi f} + \hat{d}_2 e^{-4j\pi f}\right|^2} \hat{\sigma}_{\mathbf{e}}^2$$

Compare it with the true power spectrum.

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

$$\begin{aligned}
F\left(ax(t)+by(t)\right) =& aX(f)+bY(f)\\
F\left(x(t-t_0)\right) =& e^{-j2\pi ft_0}X(f)\\
F\left(e^{j2\pi f_0 t}x(t)\right) =& X(f-f_0)\\
F\left(x(at)\right) =& \frac{1}{|a|}X\left(\frac{f}{a}\right)\\
F\left(x(t)*y(t)\right) =& X(f)\times Y(f)\\
F\left(x(t)\times y(t)\right) =& X(f)*Y(f)
\end{aligned}$$

Discrete-time Fourier transform (DTFT)

The DTFT is defined as:

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

The inverse DTFT is defined as:

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

Discrete Fourier Transform (DFT)

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

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

The inverse DFT is defined as:

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

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	$\lim_{N\to+\infty}\sum_{k=0}^{N} q^k = \frac{1}{1-q}$	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}	$z \neq 0$
$u(k)$	$\frac{z}{z-1}$	$\lvert z\rvert > 1$
$ku(k)$	$\frac{z}{(z-1)^2}$	$\lvert z\rvert > 1$
$k^2u(k)$	$\frac{z(z+1)}{(z-1)^3}$	$\lvert z\rvert > 1$
$a^k u(k)$	$\frac{z}{z-a}$	$\lvert z\rvert > \lvert a\rvert$
$ka^k u(k)$	$\frac{az}{(z-a)^2}$	$\lvert z\rvert > \lvert a\rvert$
$k^2a^k u(k)$	$\frac{az(z+a)}{(z-a)^3}$	$\lvert z\rvert > \lvert a\rvert$
$cos(\omega_0 k)u(k)$	$\frac{z(z-cos(\omega_0))}{z^2-2cos(\omega_0)z+1}$	$\lvert z\rvert > \lvert 1\rvert$
$sin(\omega_0 k)u(k)$	$\frac{sin(\omega_0)z}{z^2-2cos(\omega_0)z+1}$	$\lvert z\rvert > \lvert 1\rvert$

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)	$x(k-1)$	$z^{-1}X(z) + x(-1)$
	$x(k-i)$	$z^{-i}X(z)$
convolution	$y(k) = h(k) * u(k)$	$Y(z) = H(z)U(z)$
	$y(k) = \sum_{n=-\infty}^{\infty} h(k)u(n-k)$	
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 $k < 0$	$x(0) = \lim_{z\to+\infty} X(z)$
final value theorem	$\lim_{k\to+\infty} x(k) = \lim_{z\to 1}(z-1)X(z)$	if the limit exists

Appendix

Properties of the mean and variance of random variables

For any pair-wise independent random variables $X_1, X_2, \ldots, X_n$ and for any constants $a_1, a_2, \ldots, a_n$, we have

$$E(a_1X_1 + \ldots + a_nX_n) = a_1E(X_1) + \ldots + a_nE(X_n)$$
$$Var(a_1X_1 + \ldots + a_nX_n) = a_1^2Var(X_1) + \ldots + a_n^2Var(X_n)$$

If X and Y are two random variables such that

$$Y = aX + b$$

where a and b are two constant, then we have

$$E(Y) = E(aX + b) = aE(X) + b$$
$$Var(Y) = Var(aX + b) = a^2Var(X)$$

Stochastic process - Summary of the main results

In signal theory, we usually start analyzing random signals from time-domain data. Their evolution seems unpredictable. To model them, we use the mathematical description of *random process* which associates for each event a realization which is a function of time. Each realization of the random process presents the same global characteristics. For example, each realization oscillates around the same value, with the same average fluctuations and will have the same frequency content.
A random or stochastic signal, denoted as $y(k)$ is therefore assumed to be a realization of a random process denoted in bold $\mathbf{y}(k)$. In Matlab, the `rand` or `randn` functions can be used to generate easily realizations of a random process. The command lines below allow, for example, to generate and plot four realizations of a Gaussian random process of variance 1 observed over 100 points:

```
y=randn(100,4);
for i=1:4
subplot(2,2,i)
plot(y(:,i))
end
```

The four realizations of the Gaussian random process are represented on Figure 6.2. Although they seem totally unpredictable, we can deduce for the 4 realizations that the mean is zero and that the standard deviation is about 1.

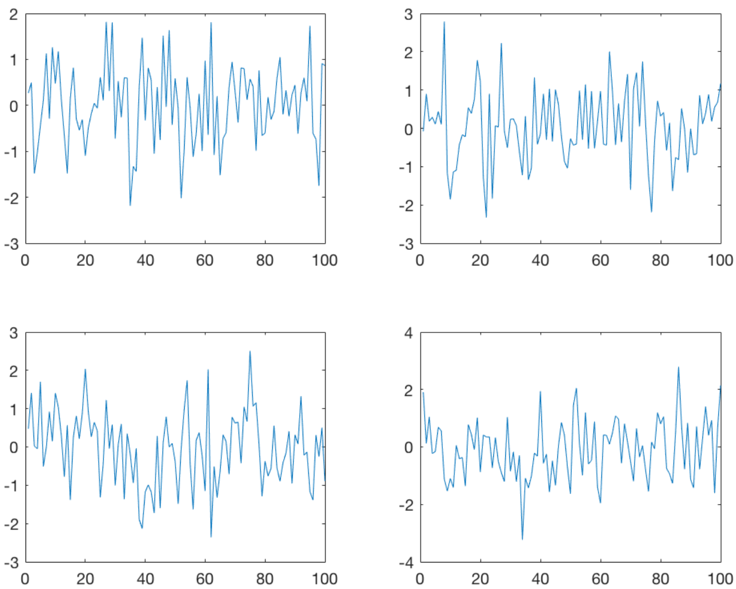

Figure 6.2: Four realizations Gaussian random process

We will restrict our attention to **stationary** stochastic processes. As the term *stationary* indicates, a stationary (in a wide-sense) stochastic process has its mean and auto-correlation independent of the time-instant where they are evaluated.
We will also assume that the stochastic process is **ergodic** which means that its statistical properties can be deduced from a single, sufficiently long, realization of the random process. This definition implies that the average statistical properties of the random process can be determined from a single realization. It is reminded that, for a process to be ergodic, it has to necessarily be stationary. But not all stationary processes are ergodic.
Given two (jointly) stationary stochastic ergodic processes $\mathbf{x}(k)$ and $\mathbf{y}(k)$, we have the following definitions:

- **Mean** of $\mathbf{y}(k)$
$$m_{\mathbf{y}} = E\big(\mathbf{y}(k)\big)$$
- **Variance** of $\mathbf{y}(k)$
$$Var\big(\mathbf{y}(k)\big) = E\big(\mathbf{y}(k) - E(\mathbf{y}(k)^2)\big) = \sigma_{\mathbf{y}}^2$$
- **Cross-correlation function**
$$R_{\mathbf{yx}}(\tau) = E\big(\mathbf{y}(k)\mathbf{x}(k+\tau)\big) \quad \forall \quad \tau \in Z$$
- **Auto-correlation function** of $\mathbf{y}(k)$
$$R_{\mathbf{y}}(\tau) = E\big(\mathbf{y}(k)\mathbf{y}(k+\tau)\big) \quad \forall \quad \tau \in Z$$
- **Power** of $\mathbf{y}(k)$:
$$P_{\mathbf{y}} = E\big((\mathbf{y}(k))^2\big) = R_{\mathbf{y}}(0) = \sigma_{\mathbf{y}}^2 + m_y^2$$
If the signal is zero-mean ($m_{\mathbf{y}} = 0$), $P_{\mathbf{y}} = R_{\mathbf{y}}(0) = \sigma_{\mathbf{y}}^2$
- **Power spectrum or power spectral density** of $\mathbf{y}(k)$
It is defined as the discrete-time Fourier transform (DTFT) of the auto-correlation function:
$$\Phi_{\mathbf{y}}(f) = \sum_{\tau=-\infty}^{\infty} R_{\mathbf{y}}(\tau) e^{-j2\pi f\tau}$$
- **Correlation**
If the value of $\mathbf{y}(k)$ at time-instant k is not correlated with the value of $\mathbf{x}(k-\tau)$ (for a given value τ) then
$$R_{\mathbf{yx}}(\tau) = 0$$
- **Dependency**
If $\mathbf{y}(k)$ and $\mathbf{x}(k)$ are independent then
$$R_{\mathbf{yx}}(\tau) = 0 \quad \forall \quad \tau \in Z$$
- **White noise**
A white noise commonly denoted as $\mathbf{e}(k)$ is a very special zero-mean stationary random process. Its power spectrum is constant over all frequencies. Its auto-correlation function is a Kronecker impulsion at $\tau = 0$ of magnitude $\sigma_{\mathbf{e}}^2$. The sample at time-instant k is uncorrelated with the samples at other time-instants. The properties of a white noise $\mathbf{e}(k)$ of variance $\sigma_{\mathbf{e}}^2$ are summarized below :
$$E\big(\mathbf{e}(k)\big) = 0 \qquad R_{\mathbf{e}}(\tau) = \begin{cases} \sigma_{\mathbf{e}}^2 & \text{pour } \tau = 0 \\ 0 & \text{otherwise} \end{cases} \qquad \Phi_{\mathbf{e}}(f) = \sigma_{\mathbf{e}}^2 \quad \forall \quad f$$
- **Filtering of a stationary random process**
Consider a stationary random process $\mathbf{x}(k)$ of power spectrum $\Phi_{\mathbf{x}}(f)$ sent to the input of a linear filter defined by its impulse response $h(k)$ and its frequency response (or DTFT) $H(f)$. It can be shown that the filtered signal is also a random process with the following properties:
  - Its mean is :
$$m_{\mathbf{y}} = H(f)\big|_{f=0}\, m_{\mathbf{x}}$$

- Its autocorrelation function is :

$$R_{\mathbf{y}}(\tau) = R_h(\tau) * R_{\mathbf{x}}(\tau)$$

- Its power spectrum is:

$$\Phi_{\mathbf{y}}(f) = |H(f)|^2 \Phi_{\mathbf{x}}(f).$$

- **AR process of order n_d**

  - Its difference equation takes the following form:

$$\mathbf{y}(k) + d_1\mathbf{y}(k-1) + d_2\mathbf{y}(k-2) + \ldots + d_{n_d}\mathbf{y}(k-n_d) = \mathbf{e}(k)$$

    where $\mathbf{e}(k)$ is a white noise of variance $\sigma_{\mathbf{e}}^2$.

  - Its power spectrum is:

$$\Phi_{\mathbf{y}}(f) = \frac{1}{\left|1 + \sum_{k=1}^{n_d} d_k e^{-2j\pi kf}\right|^2} \sigma_{\mathbf{e}}^2$$

  In Matlab, it is easy to generate a realization of an AR process by using the function `filter` and `randn`. The command lines below allow, for example, to generate and plot four realizations of an AR process of order 1 with unit variance observed over 100 points:

```
e=randn(100,1);
y=filter(1,[1 -0.9],e)
plot(y)
```

- **MA process of order n_c**

  - Its difference equation takes the following form:

$$\mathbf{y}(k) = \mathbf{e}(k) + c_1\mathbf{e}(k-1) + \ldots + c_{n_c}\mathbf{e}(k-n_c)$$

    where $\mathbf{e}(k)$ is a white noise of variance $\sigma_{\mathbf{e}}^2$.

  - Its power spectrum is:

$$\Phi_{\mathbf{y}}(f) = \left|1 + \sum_{k=1}^{n_c} c_k e^{-2j\pi kf}\right|^2 \sigma_{\mathbf{e}}^2$$

  In Matlab, it is easy to generate a realization of a MA process by using the function `filter` and `randn`. The command lines below allow, for example, to generate and plot four realizations of a MA process of order 2 with unit variance observed over 100 points:

```
e=randn(100,1);
y=filter([1 1.5 -1.2],1,e)
plot(y)
```

- **ARMA(n_c, n_d) process**

  - Its difference equation takes the following form:

$$\mathbf{y}(k) + d_1\mathbf{y}(k-1) + d_2\mathbf{y}(k-2) + \ldots + b_{n_d}\mathbf{y}(k-n_d) = \mathbf{e}(k) + c_1\mathbf{e}(k-1) + \ldots + c_{n_c}\mathbf{e}(k-n_c)$$

    where $\mathbf{e}(k)$ is a white noise of variance $\sigma_{\mathbf{e}}^2$.

  - Its power spectrum is:

$$\Phi_{\mathbf{y}}(f) = \frac{\left|1 + \sum_{k=1}^{n_c} c_k e^{-2j\pi kf}\right|^2}{\left|1 + \sum_{k=1}^{n_d} d_k e^{-2j\pi kf}\right|^2} \sigma_{\mathbf{e}}^2$$

- **Shift operator q^{-1}**
  It is defined as

$$y(k-1) = q^{-1}y(k)$$