# Digital Signal Processing Fundamentals

5ESC2: Labs

4 september 2017

| | | |
|---|---|---|
| Responsible Lecturer | : | Piet Sommen (Flux 7.101) |
| Lecturer | : | Fons van der Sommen (Flux 5.092) |
| Lecturer | : | Bruno Arsenali (Flux 7.104) |
| Assistant | : | Anton Alexeev (Flux 7.062) |
| Assistant | : | Patrick Langenhuizen (Flux 5.092) |
| Assistant | : | Stephan Heunis (Flux 7.062) |
| Assistant | : | Simona Turco (Flux 7.078) |

## Purpose of the labs

By doing the assignments of the labs a better understanding of the concepts of the most important signal processing operations, that are treated in the course 5ESC1, can be achieved. There are three different DSP-labs which treat topics such as filtering, convolution of simple discrete-time signals and system, sampling, DFT, filter design and stochastic signal processing.

### General information

- All participants must register in Canvas.
- After registration in Canvas, all participants must register in a DSP-group (max. 2 students each).
- All information (assignments, etc) will be available in Canvas.
- Each group has to carry out all assignments and hand in three reports: Lab1, Lab2 and Lab3. These reports are predefined documents (see Canvas).
- Do not forget to fill in your names, group number and ID's at the front page.
- Print out and deliver each report ultimately at the deadline to one of the assistents, or put the document in one of their post boxes (opposite SPS secretary Flux 7.068, or Flux 5.092).
- Plagiarism will not be accepted. In the worst case the lab(s) will be judged by zero credits.

## Deadlines and credits:

The three labs count in total for 30% of the course 5ESC0 "Digital Signal Processing Fundamentals". The deadlines and credits of the labs are according the following scheme:

| Code | Assignment | Deadline / Date | Credits |
|---|---|---|---|
| Lab1 | Signals Systems and sampling | September 27, 13:30 | 10% |
| Lab2 | DFT and Filter design | October 11, 13:30 | 10% |
| Lab3 | Stochastic signal processing | October 25, 13:30 | 10% |

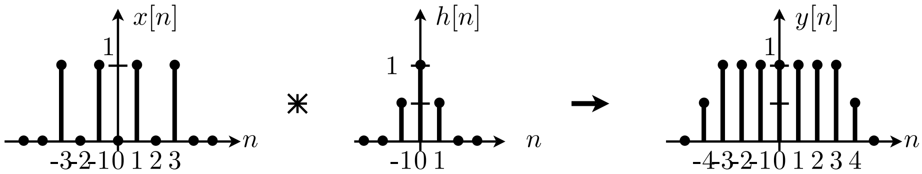
# 1 Lab 5ESC2: Signals, Systems and Sampling

## 1.1 Convolution operation

The convolution operation $y[n] = x[n] * h[n]$ is defined as:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{1}$$

In this equation the input signal samples are represented by the symbol $x[n]$, the output samples by $y[n]$ and the impulse response of the filter by $h[n]$. A simple example of the result of such a convolution operation is depicted in Fig. 1. In this simple example the input signal $x[n]$ is a finite




Figuur 1: Simple example of convolution operation

sequence of samples and is defined as:

$$x[n] = \begin{cases} 1 & \text{for } n = -3, -1, 1, 3 \\ 0 & \text{elsewhere.} \end{cases} \tag{2}$$

while the, **non-causal**, impulse response $h[n]$ is defined as:

$$h[n] = \begin{cases} \frac{1}{2} & \text{for } n = -1, 1 \\ 1 & \text{for } n = 0 \\ 0 & \text{elsewhere.} \end{cases} \tag{3}$$

With the equation (1) for the convolution operation the finite output sequence $y[n]$ becomes:

$$y[n] = \begin{cases} \frac{1}{2} & \text{for } n = -4, 4 \\ 1 & \text{for } n = -3, -2, -1, 0, 1, 2, 3 \\ 0 & \text{elsewhere.} \end{cases} \tag{4}$$

**Assignment 1: Convolution**

Evaluate, first by hand, the convolution operation of the signal $x[n]$, as defined in (2), with impulse response

$$h[n] = \begin{cases} -\frac{1}{2} & \text{for } n = -1, 1 \\ 1 & \text{for } n = 0 \\ 0 & \text{elsewhere.} \end{cases}$$

When you have finished this, use the MATLAB command `conv` to verify your calculations. **Make a plot using MATLAB command `stem` and take care that both x- and y-axis denotes the correct values!**

In practice most signals have a finite length. In such a case we have to take care of the fade-in and fade-out properties. Some leading and trailing samples from the convolution output need to be handled with caution.

**Assignment 2: Fade-in and -out of convolution**

Assume the finite length $N$ input signal $x[n]$ is defined as:

$$x[n] = \begin{cases} 1 & \text{for } n = 0, 1, \cdots, N-1 \\ 0 & \text{elsewhere.} \end{cases}$$

and the length $M = 3$ impulse response $h[n]$ equals equation (3).

a) Determine the length of the output of $x[n] * h[n]$, expressed in $N$ and $M$. How long is the fade-in and fade-out phenomenon?

b) If we choose $N = 8$ and $M = 3$, how many (and which) of the output samples will have no fade-in and fade-out?

## 1.2 Causality of a filter

In the previous section we evaluated the convolution operation of an input signal $x[n]$ with a non-causal impulse response $h[n]$. The result of this non-causality is that the output samples $y[n]$ appears before the input samples $x[n]$. This can be seen in Fig. 1. In practice this may cause a problem. For this reason we have to make the impulse response $h[n]$ causal. In general this can be done by applying a shift operation to this causal impulse response $h[n]$. Mathematically such a shift operation can be performed by applying a convolution operation of $h[n]$ with a shifted delta pulse over $L$ samples, with $L \geq 0$. Thus

$$h_{causal}[n] = h[n] * \delta[n - L] \tag{5}$$

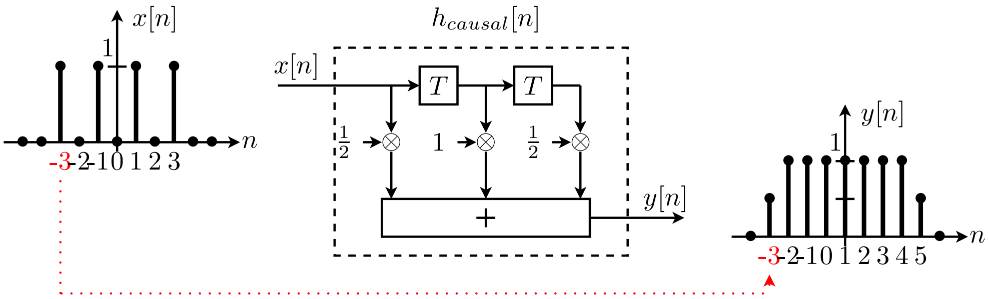
in which the shifted delta pulse is defined as:

$$\delta[n - L] = \begin{cases} 1 & \text{for } n = L \\ 0 & \text{elsewhere.} \end{cases} \tag{6}$$

As a simple example, we can make impulse response $h[n]$ of equation (3) causal, by convolving with $\delta[n - 1]$, which results in

$$h_{causal}[n] = \begin{cases} \frac{1}{2} & \text{for } n = 0, 2 \\ 1 & \text{for } n = 1 \\ 0 & \text{elsewhere.} \end{cases} \tag{7}$$

This causal impulse response can be used to realize a filter in practice. A realization scheme of this simple example is depicted in Fig.2. In this realization scheme the delays are represented by




Figuur 2: Realization of a filter with causal impulse response

the symbols $T$.

*Note: In practice the signal samples $x[n]$ and $y[n]$ can have non-zero values for $n < 0$ when the* *samples are recorded.*

**Assignment 3: Causality**

Now we are given the following non-causal impulse response:

$$h[n] = \begin{cases} 1 & \text{for } n = -5, -4, -3 \\ 0 & \text{elsewhere} \end{cases}$$

a) How do you have to choose $L$ of a shifted delta pulse in order to make this impulse response causal?

b) Choose a value for $L$ according your answer in a) and denote the causal impulse response as $h_{causal}[n]$. Use the input signal $x[n]$ as defined in equation (2) and evaluate both the non-causal $y[n]$) and causal ($y_{causal}[n]$) convolution results. Verify your results with MATLAB plots (**Make sure that both x- and y-axis denotes the correct values!**).

## 1.3 The Fourier Transform for Discrete-time signals (FTD)

An important transformation in digital signal processing is the FTD, which transforms the discrete-time signal samples $x[n]$ to the frequency domain function $X(e^{j\theta})$ and is defined as:

$$x[n] \quad \circ\!\!-\!\!\circ \quad X(e^{j\theta}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\theta n}. \tag{8}$$

In this equation the relative frequency is denoted by the symbol $\theta$. Because of the fact the samples $x[n]$ are discrete in time, the resulting frequency domain representation $X(e^{j\theta})$ is a periodic function of $\theta$. Typically one period of $X(e^{j\theta})$ is the so called Fundamental Interval (FI): $-\pi < \theta \leq \pi$. When applying the FTD of the impulse response $h[n]$ of a filter we obtain the frequency response $H(e^{j\theta})$. Thus we have the following FTD pair:

$$h[n] \quad \circ\!\!-\!\!\circ \quad H(e^{j\theta}) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\theta n} \tag{9}$$

**Assignment 4: Frequency response**

a) Calculate the frequency response $H(e^{j\theta})$ of the non-causal impulse response $h[n]$ as defined in equation (3).
*Note: Use the Euler expression to write the sum of complex exponents as a sinusoidal function.*

b) Make a MATLAB function to plot in one figure, below each other using MATLAB function `subplot`, both the magnitude $|H(e^{j\theta})|$ and phase $\phi_H(e^{j\theta})$ as a function of $\theta$ in the FI. Make your MATLAB evaluations for `theta = [-pi:0.001:pi]`.

c) How would you rate the main character of this filter, as low-pass, high-pass or band-pass? Give a short explanation.

An important property of the FTD is that the convolution operation in time domain is represented by a multiplication in the frequency domain, symbolically: $* \quad \circ\!\!-\!\!\circ \quad \cdot$.
In one of the previous subsections, we have seen that we can make a non-causal filter, with impulse response $h[n]$, causal by applying an appropriate delay. Mathematically this is denoted by the

convolution operation as expressed in equation (5). By applying the FTD to this equation we obtain:

$$h_{causal}[n] = h[n] * \delta[n-L] \quad \circ\!\!-\!\!\circ \quad H_{causal}(e^{j\theta}) = H(e^{j\theta}) \cdot D(e^{j\theta}) \tag{10}$$

with $D(e^{j\theta})$ the FTD of the shifted delta function $\delta[n-L]$.

**Assignment 5: Frequency response of causal filter**

a) Calculate the frequency response $H_{causal}(e^{j\theta})$ of the causal impulse response $h_{causal}[n]$ as defined in equation (7).

b) Make a MATLAB function to plot in one figure, below each other, both the magnitude $|H_{causal}(e^{j\theta})|$ and phase $\phi_{H_{causal}}(e^{j\theta})$ as a function of $\theta$ in the FI. Make your MATLAB evaluations for `theta = [-pi:0.001:pi]`.

c) Write $H_{causal}(e^{j\theta})$ as the product $H(e^{j\theta}) \cdot D(e^{j\theta})$, with $H(e^{j\theta})$ the FTD of the impulse response as defined in equation (3). Explain the difference between $H(e^{j\theta})$ and $H_{causal}(e^{j\theta})$.

## 1.4 The Inverse Fourier Transform of Discrete-time (IFTD) signals

The IFTD is defined as follows:

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta}) \cdot e^{j\theta n} d\theta. \tag{11}$$

In practice the design of a filter very often starts by defining the frequency behavior of the filter. An example is the Low Pass Filter (LPF)with cutoff frequency $\theta_c$ which is defined as follows:

$$H(e^{j\theta}) = \begin{cases} 1 & \text{for } |\theta| <= \theta_c \\ 0 & \text{for } \theta_c < |\theta| \leq \pi \end{cases}$$

The impulse response $h[n]$ can be obtained by evaluating the IFTD of the frequency response $H(e^{j\theta})$. Because of the fact that the ideal LPF has infinite sharp edges, the resulting impulse response $h[n]$ will have infinite length. In practice however we can only store a finite set of values, which can be obtained by multiplying the impulse response $h[n]$ with a finite length window $w[n]$ (of odd length $N$). This results into the following approximated version of the impulse response:

$$\tilde{h}[n] \quad = \quad \begin{cases} w[n] \cdot h[n] & \text{for } -\frac{N-1}{2} \leq n \leq \frac{N-1}{2} \\ 0 & \text{elsewhere} \end{cases} \tag{12}$$

**Assignment 6: Impulse response of low pass filter**

a) Calculate by hand via the IFTD the impulse response $h[n]$ of an ideal LPF with cut-off frequency $\theta_c = \frac{\pi}{3}$

b) Use MATLAB to plot the approximated impulse response $\tilde{h}[n]$ for $N = 11$ when using a rectangular window function $w[n]$.

c) Now we have the discrete-time input signal $x[n] = \sin(\frac{\pi}{6}n) + \sin(\frac{5\pi}{6}n)$. Filter this signal with $\tilde{h}[n]$ to produce the output signal $y[n]$. Play the input signal $x[n]$ and the output signal $y[n]$ by using a sample rate $f_s = 4$ [kHz]. Explain what you hear by using the Matlab function `soundsc`.

As we have seen before convolution in one domain (e.g. time-domain) is equivalent to a multiplication in the other domain (e.g. frequency-domain. Thus by using the property

$$y[n] = x[n] * h[n] \quad \circ\!\!-\!\!\circ \quad Y(e^{j\theta}) = X(e^{j\theta}) \cdot H(e^{j\theta}).$$

we are able to calculate a time-domain convolution operation via a frequency-domain multiplication. The procedure to do this is as follows:

1. $X(e^{j\theta}) = \mathrm{FTD}\{x[n]\}$ and $H(e^{j\theta}) = \mathrm{FTD}\{h[n]\}$,
2. $Y(e^{j\theta}) = X(e^{j\theta}) \cdot H(e^{j\theta})$,
3. $y[n] = \mathrm{IFTD}\{Y(e^{j\theta})\}$.

**Assignment 7: Calculation of convolution via FTD and IFTD**

Verify by hand the convolution result of Assignment 1 by using the above described procedure. Thus calculate $X(e^{j\theta})$, $H(e^{j\theta})$, $Y(e^{j\theta})$ and find the result $y[n] = \mathrm{IFTD}\{Y(e^{j\theta})\}$.

## 1.5 Sampling and Aliasing

A continuous-time signal can be filtered or manipulated efficiently in a discrete-time (or digital) system, such as a PC or a digital signal processor (DSP). To obtain a discrete-time signal, the continuous-time signal is sampled with a sampling frequency $f_s = 1/T_s$ [Hz]. If we start with a continuous-time signal $x_c(t)$, then its discrete-time representation is $x[n \cdot T_s] = x_c(t)|_{n \cdot T_s}$. This means that the sample values $x[n \cdot T_s]$ are obtained by evaluating the value of $x_c(t)$ for each $t = n \cdot T_s$, with $n$ an integer. In case that we use only one sampling frequency, usually the discrete-time samples are denoted by $x[n]$, so leaving out the symbol $T_s$.
Now let's start with a continuous-time sine wave, which can mathematically be described by: $x_c(t) = A \sin(\omega t + \phi)$, with $\omega = 2\pi f = 2\pi \frac{1}{T}$, where $f$ is the absolute frequency in [Hz] and $T = 1/f$ is the period of the sinusoidal signal. If we sample this continuous-time sinusoidal signal with a sample frequency $f_s = 1/T_s$ [Hz], we obtain:

$$x[n] = A \sin(\omega T_s n + \phi) = A \sin(2\pi \frac{f}{f_s} n + \phi) = A \sin(\theta n + \phi)$$

From this simple example it follows that in general the relation between the absolute frequency $f$ and the relative frequency $\theta$ is given by the following equation:

$$\theta = 2\pi \frac{f}{f_s} = \omega \cdot T_s \tag{13}$$

**Assignment 8: Sampling a sinusoidal signal**

In MATLAB, we can visualize $N$ samples of the above described discrete-time signal as follows:

```
n = 0:N-1;
theta = 2*pi*(f/fs);
x = A*sin(theta.*n+phi);
stem(n,x);
```

a. Enter the above piece of code in MATLAB with $f = 800$ [Hz], $A = 1$, $\phi = 0$, and $f_s = 4$ [kHz]. Plot in one and the same figure the samples $x[n]$ and two periods of the continuous-time function $x_c(t)$. Use in MATLAB a very fine time grid (e.g 10 times faster than $f_s$) to evaluate $x_c(t)$ and use the function `plot` for $x_c(t)$ and `stem` for $x[n]$. Ensure that the x-axis corresponds to the actual time $t$ in [msec].

b. Use the MATLAB function `sound` to play, for example, 100 periods of $x[n]$. Now increase the frequency $f$ in steps of 800 [Hz] to 4000 [Hz] and play the audio each new step. What do you notice? Provide a statement.

In the discrete-time domain there is no unique way to represent frequencies. This can easily be seen by the following relation:

$$x[n] = A\cos(\theta n + \phi) \equiv A\cos(\{\theta + 2\pi\} \cdot n + \phi) \equiv A\cos(\{-\theta + 2\pi\} \cdot n - \phi) \tag{14}$$

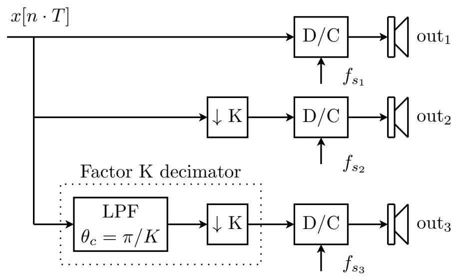
From this example it follows that the relative frequency $\theta$ results in the same sample values $x[n]$ as the relative frequency $\theta + 2\pi$ or $-\theta + 2\pi$ or .... Because of this ambiguity we have to make a choice which frequency we use if we convert the discrete-time signal samples back to a continuous-time signal. Usually we choose the frequencies in the Fundamental Interval (FI) $-\pi < \theta \leq \pi$.
So if we convert a continuous-time signal first to the discrete-time and then again back to a continuous-time signal, the result of the ambiguity is that first absolute frequencies $f$ of $x_c(t)$ are mapped, via equation (13), to relative frequencies $\theta$. When these relative frequencies are outside the FI, they will be mapped inside the FI. These new frequencies inside the FI are used when converting the discrete-time samples to a continuous-time signal. In other words, the original continuous-time frequency may have changed. This process of 'mapping into the FI' is called **aliasing**.

**Assignment 9: Visualization of 'aliasing' via time domain**

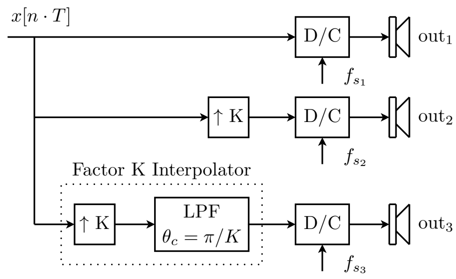
During the course the aliasing effect has been explained via the frequency domain. In this assignment you have to visualize the aliasing effect via time domain. For this purpose we use a fixed sampling frequency $f_s = 0.5$ Hz and a continuous-time signal $x_c(t) = \cos(2\pi f t + \phi)$, with $f = 0.2$ [Hz] and $\phi = \pi/4$. The sampled version of $x_c(t)$ is denoted as $x[n] = \cos(2\pi \frac{f}{f_s} n + \phi)$.
By using equation (14), derive two different sinusoidal signals $x_c(t)$, with different frequencies between 0 [Hz] and 0.75 [Hz], which result in the same sample values $x[n]$. Give this derivation and show in one MATLAB plot all these signals, thus 3 different versions of $x_c(t)$ together with the samples $x[n]$. Finally explain this result in frequency domain.

## 1.6 Up- and down sampling

In order to manipulate the sampling frequency, directly in the discrete-time domain, we can make use of the Sample Rate Increase (SRI) and Sample Rate Decrease (SRD) functions. From theory we have seen that the SRI introduce mirrors (replicas of the spectral content of the original signal into the FI), while the SRD introduces aliasing. To overcome these imperfections an SRI is usually followed by an LPF (to overcome the mirrors), which results in an interpolation scheme. An SRD is usually preceded by an LPF (to overcome aliasing), which results in an decimation scheme. A factor $K$ decimator is shown in Fig. 3 and a factor $K$ interpolator in Fig. 4.




Figuur 3: Factor $K$ SRD and decimator




Figuur 4: Factor $K$ SRI and interpolator

**Assignment 10: Up- and down-sampling**

In order to show the result of up- and down-sampling, use in this assignment the following:

- The up- and down-sample factor $K = 2$,
- The audio file `'scale.wav'`,
- To read in MATLAB an audio file `'in.wav'` as a vector `y` use:
  `[y, fsin]= audioread('in.wav')`,
- To write in MATLAB a file `y` to an audio file `'out.wav'`:
  `audiowrite('out.wav',y, fsout)`.
- Use the general expression for the coefficients of the LPF as given in equation (12). Find an appropriate value of $N$ which gives reasonable results.

a) Implement all 3 versions of the scheme as depicted in Fig. 3. Print the MATLAB code in the appendix. Upload in your group folder this MATLAB code and the resulting audio files. Give a short explanation of your results.

b) Implement all 3 versions of the scheme as depicted in Fig. 4. Print the MATLAB code in the appendix. Upload in your group folder this MATLAB code and the resulting audio files. Give a short explanation of your results.

# Hand in your report Lab1 of the course 5ESC0

# 2 Lab 5ESC3: DFT and Filter design

## 2.1 The Discrete Fourier Transform (DFT)

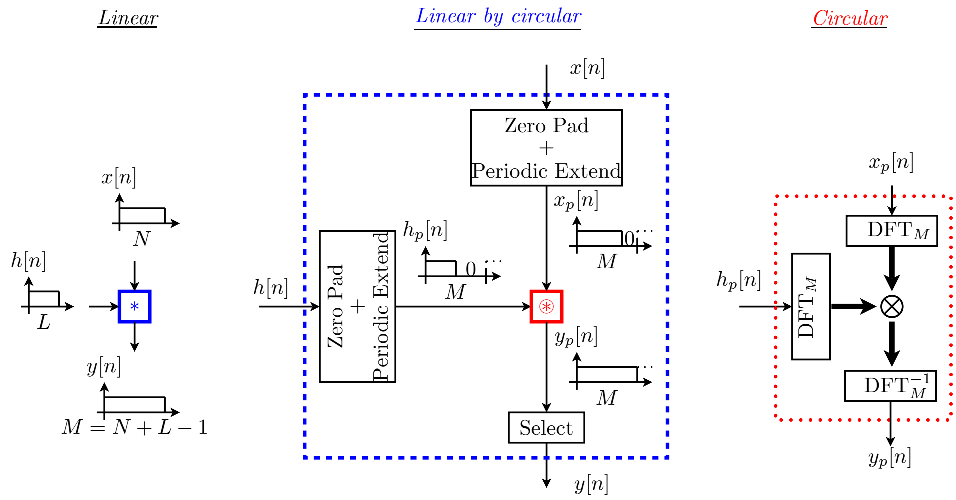
In practice we can evaluate and save only a finite number of sampled values in a digital system. Above that in most cases we can't find a mathematically expression of the FTD for all frequencies $\theta$, since $\theta$ is a continuous variable. For this reason the DFT has been invented and the length $N$ DFT of the signal samples $x[n]$, $n = 0, 1, \cdots, N-1$ is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad \text{with twiddle factor} \quad W_N = \mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}} \quad \text{and} \quad k = 0, 1, \cdots, N-1 \tag{15}$$

In general we can make the following remarks about the DFT:

- The DFT does not evaluate the spectral content of $x[n]$ for every relative frequency $\theta$. The spectral content is only evaluated for frequency indices $k$ in the range $k = 0, 1, \cdots, N-1$ and each index $k$ is related to the relative frequency $k \cdot \frac{2\pi}{N}$
- In MATLAB, we can use the command `fft`[1] to perform the DFT. The frequency index $k$ represent a relative frequency range: $0 \leq \theta < 2\pi$. In case a relative frequency range $-\pi \leq \theta < \pi$ is needed, the function `fftshift` can be used.
- From the DFT definition (15) it follows that the values of the frequency components $X[k]$ represent a periodic function with period $N$, since $X[k] = X[k + l \cdot N]$ with $l = 0, \pm1, \pm2, \cdots$. For this reason we usually calculate the values of $X[k]$ only in the range $k = 0, 1, \cdots, N-1$.
*Note: This periodic property follows the main concept: Sampling in one domain results in periodic extension in the other domain*
- The length $N$ DFT uses $N$ samples of the signal $x[n]$, even in case $x[n]$ contains more or even less then $N$ samples.
*Note: In case the signal length is less than $N$, the extra samples are assumed to be zero.*
- **Only** for a signal with **finite length** $N$, the values $X[k]$ of the length $N$ DFT represent the sampled values of the FTD. This can be proven as follows:
Assume that $x[n] \neq 0$ for $n = 0, 1, \cdots, N-1$ and $x[n] = 0$ elsewhere. Then the FTD can be evaluated by:
$$X(\mathrm{e}^{\mathrm{j}\theta}) = \sum_{n=-\infty}^{\infty} x[n]\mathrm{e}^{-\mathrm{j}\theta n} = \sum_{n=0}^{N-1} x[n]\mathrm{e}^{-\mathrm{j}\theta n}$$
Now, if we evaluate the FTD for the frequencies $\theta = k \cdot \frac{2\pi}{N}$ we obtain:
$$X(\mathrm{e}^{\mathrm{j}\theta})|_{\theta = k\cdot\frac{2\pi}{N}} = \sum_{n=0}^{N-1} x[n]\mathrm{e}^{-\mathrm{j}k\cdot\frac{2\pi}{N}\cdot n} \equiv X[k] = \mathrm{DFT}_N\{x[n]\}$$
- Circular convolution ($\circledast$) between two periodic extended (PE) sequences $x_p[n]$ and $h_p[n]$, can be performed in the DFT domain by a multiplication as depicted in the right hand side of Figure 5. When using FFTs such a circular convolution can be performed very efficiently. However, when filtering a signal $x[n]$ with a filter with an impulse response $h[n]$, we have to apply a linear convolution ($*$), as depicted in the left hand side of Figure 5. However linear and circular convolution are fundamentally different operations. The linear convolution of a length $N$ sequence $x[n]$ and a length $L$ sequence $h[n]$ results in a length $M = N + L - 1$ sequence $y[n]$. For the circular convolution to be equivalent to the linear convolution result, we have to pad both sequences with zeros to a length of at least $M$, as depicted in the middle figure of Figure 5.

[1]FFT = Fast Fourier Transform, which is an efficient implementation for the DFT




Figuur 5: Linear and circular convolution

The Inverse DFT (IDFT) is defined as:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn} \quad \text{with twiddle factor} \quad W_N = \mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}} \quad \text{and} \quad n = 0, 1, \cdots, N-1 \tag{16}$$

From this equation we can make the following remark:

- From the definition (16) it follows that the samples $x[n]$ are periodic with period $N$, since $x[n] = x[n + m \cdot N]$ with $m = 0, \pm 1, \pm 2, \cdots$. For this reason we usually calculate the values of $x[n]$ only in the range $n = 0, 1, \cdots, N-1$.
  *Note: Again we see here that the periodic property follows the main concept: Sampling in one domain results in periodic extension in the other domain*

**Assignment 11: The DFT of a finite length discrete-time signal**

Use a finite length discrete-time signal $x[n] = 1$ for $n = -\frac{K-1}{2}, \cdots, -1, 0, 1, \cdots, \frac{K-1}{2}$ and $x[n] = 0$ elsewhere (block-function). In this assignment you have to use $K = 11$.

a) Show (by hand) that the FTD $X(\mathrm{e}^{\mathrm{j}\theta})$ of the block function $x[n]$ can be written as follows:

$$X(\mathrm{e}^{\mathrm{j}\theta}) = A\frac{\sin(K_1\theta)}{\sin(K_2\theta)} \cdot \mathrm{e}^{-\mathrm{j}K_3\theta}$$

Evaluate the parameters $A, K_1, K_2$ and $K_3$.
*Note: Start by simplifying the FTD result as much as possible by using the geometric series* $\sum_{n=0}^{K-1}(a)^n = \frac{1-a^K}{1-a}$

b) Make a MATLAB function to plot the magnitude $|X(\mathrm{e}^{\mathrm{j}\theta})|$ as a function of $\theta$ in the FI. Make your MATLAB evaluations for `theta = [-pi:0.001:pi]`.

c) Verify if the DFT values of this finite length signal are equal to the sampled values of the FTD. Do this by using MATLAB to calculate the $N = K = 11$-point DFT of $x[n]$ and plot in the same figure as the previous one the absolute values $|X[k]|$ by using the command `stem`.

*Note: When you work with the MATLAB function* `fft`*, the index 1 indicates the DC component (frequency 0). The index $N$ does not reflect the frequency $2\pi$ but the frequency $(N-1)\cdot\frac{2\pi}{N}$. Remember this when displaying your data!*

**Assignment 12: Spectrum of a sine wave of different frequencies**

Make two sinusoidal signals in MATLAB $x_1[n] = \sin(\theta_1 \cdot n)$ and $x_2[n] = \sin(\theta_2 \cdot n)$, where $\theta_1 = 2\pi\frac{f_1}{f_s}$ en $\theta_2 = 2\pi\frac{f_2}{f_s}$. Use $f_1$, $f_2$ and $f_s$ as frequencies of 8 Hz, 9 Hz and 64 Hz respectively.

a) Calculate with MATLAB the $N = 32$-point DFTs of the signals $x_1[n]$ and $x_2[n]$. Display the resulting spectra (absolute value of DFT), by using the command `stem`, in two different figures.

b) The two spectra differ quite a lot, while $x_1[n]$ and $x_2[n]$ differ only slightly in frequency. What effect happens here? Why is this? How can this be prevented in this case?

## 2.2 Zero padding

The plotting resolution of a length $N$ DFT equals $\frac{2\pi}{N}$, which is the distance between two succeeding values of the spectrum of $x[n]$. We can improve the spectral plotting resolution by applying the so called **zero-padding**. With zero padding we append an amount of zeros to the $N$ samples up to a sequence of $R$ samples, with $R \geq N$. One of the applications of zero padding is to apply a DFT on a zero padded finite length signal in order to obtain a good approximation of the FTD of this finite length signal.
*Note: Zero padding is achieved automatically by applying the MATLAB function `fft(x,R)` to the $N$ data samples.*

**Assignment 13: Approximation of the FTD**

Use the same block function as in assignment 11, thus $x[n] = 1$ for $n = 0, 1, \cdots, K-1$ and zero elsewhere. Evaluate and plot the absolute value of the $N = 32$-point DFT of this block function, with $K = 11$. Use the `plot` function in stead of `stem`). Show (in one plot) that this result is a good approximation of the absolute value of the FTD $|X(\mathrm{e}^{\mathrm{j}\theta})|$ of assignment 11.

## 2.3 Resolution of the spectrum

In many practical cases the length of a signal $x[n]$ can be very large. As an example think of a recording of 5 minutes audio signal which has been sampled with a sample rate of 44 [kHz], which results in a signal with 13200000 samples. When applying a 'finite' length $N$ DFT we usually take 'a part of' the signal samples $x[n]$. By doing so we multiply the data samples of $x[n]$ with a finite length $N$ window function $w_R[n]$. Such a rectangular window function is defined as:

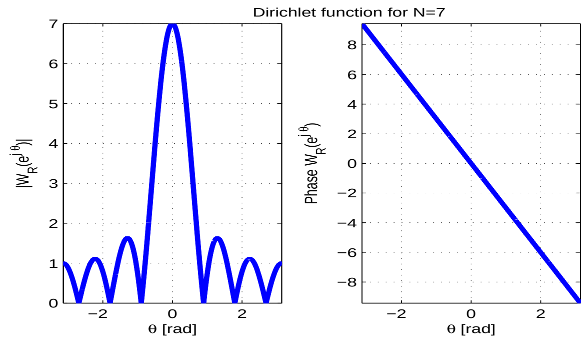
$$w_R[n] = \begin{cases} 1 & \text{for } n = 0, \ldots, N-1 \\ 0 & \text{elsewhere} \end{cases}$$

The FTD of this rectangular function is a so called Dirichlet function which can be calculated as follows:

$$W_R(e^{j\theta}) = \sum_{n=0}^{N-1} \mathrm{e}^{-\mathrm{j}\theta n} = \frac{1-\mathrm{e}^{-\mathrm{j}\theta N}}{1-\mathrm{e}^{-\mathrm{j}\theta}} = \frac{\mathrm{e}^{-\mathrm{j}\theta\frac{N}{2}}(\mathrm{e}^{\mathrm{j}\theta\frac{N}{2}} - \mathrm{e}^{-\mathrm{j}\theta\frac{N}{2}})}{\mathrm{e}^{-\mathrm{j}\theta\frac{1}{2}}(\mathrm{e}^{\mathrm{j}\theta\frac{1}{2}} - \mathrm{e}^{-\mathrm{j}\theta\frac{1}{2}})} = \frac{\sin(N\frac{\theta}{2})}{\sin(\frac{\theta}{2})}\mathrm{e}^{-\mathrm{j}\frac{(N-1)}{2}\theta}$$

An example of the amplitude and phase plot (in the FI) of the Dirichlet function for $N = 7$ is given in Fig. 6. In general the Dirichlet function has the following properties:

- Width of the main lobe: $2 \times \frac{2\pi}{N} = \frac{4\pi}{N}$
- 3dB bandwidth: $0.89\frac{2\pi}{N}$
  *Note: 3dB equals the value for which the amplitude has reduced by a factor of two.*




Figuur 6: Amplitude and phase plot in FI of Dirichlet function for $N$=7

The multiplication (in time domain) of the data samples $x[n]$ with the rectangular window $w_r[n]$, results in a convolution in the frequency domain. The spectrum of the original data $X(e^{j\theta})$ is thus convolved with $W_R(e^{j\theta})$ and we obtain:

$$\tilde{x}[n] = x[n] \cdot w_R[n] \quad \circ\!\!-\!\!\circ \quad \tilde{X}(e^{j\theta}) \quad = \quad X(e^{j\theta}) * W_R(e^{j\theta}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\vartheta}) W_R(e^{j(\theta-\vartheta)}) d\vartheta$$

From this result it follows that the original spectrum $X(e^{j\theta})$ is influenced by the FTD $W_R(e^{j\theta})$ of the window function $w_R[n]$ as follows:

- Sharp transitions (peaks) of $X(e^{j\theta})$ are 'smeared' out. This is caused by the width of the main lobe of $W_R(e^{j\theta})$. As a result, the 'spectral resolution' is reduced. Increasing the number of samples $N$ provides a narrower main lobe that will give a reduction of this effect.
- The side lobes of $W_R(e^{j\theta})$ cause 'spectral leakage'. Increasing the window length $N$ does not reduce this effect. The reason for this is that the hight of the first side lobe of a window does not depend on the window length $N$ (Gibbs phenomenon). The 'spectral leakage' can only be reduced by choosing a different window, such as Hamming, Hann, Blackman, Bartlett etc.

**Assignment 14: Calculating the minimum resolution of a spectrum**

In this assignment the spectrum is calculated by taking the absolute value of the DFT or FTD. Given an continuous-time signal $x(t)$ which is defined as follows:

$$x(t) = a_1 \sin(2\pi f_1 t) + a_2 \sin(2\pi f_2 t) \quad \text{with} \quad a_1 = 1 \text{ and } a_2 = 1$$

This signal is converted first to the discrete-time samples $x[n]$ with a sampling frequency of $f_s =$ 1000 [Hz]. After that we use for our calculations only a finite amount of these signal samples by multiplying with a window of finite length, thus $\tilde{x}[n] = x[n] \cdot w[n]$.
*Note: if not given explicitly, use a rectangular window $W_R[n]$.*

a) First use $f_1 = 175$Hz and $f_2 = 200$ [Hz]. Calculate the minimum length $N_a$ of data samples that we need in order to be able to distinguish the two spectral peaks of the windowed signal $\tilde{x}[n]$
*Use the 3 dB bandwidth of the rectangular window for your calculations.*

b) In order to show that you can distinguish the different spectral peaks of the signal, use $N_a$ data samples of the signal $x[n]$ and make a plot of the, approximated, FTD-spectrum of signal $\tilde{x}[n]$.
*Note: Use zero-padding by increasing the fft-length to e.g. $R = 5 \cdot N_a$ and use the command `plot`.*

c) Now use $f_1 = 185$Hz and $f_2 = 200$[Hz] and use again $N_a$ samples. Make a plot of the, approximated, FTD-spectrum of signal $\tilde{x}[n]$. From this plot it should follow that we can't distinguish the two spectral peaks.
Is it possible to distinguish the two peaks for this case by applying more zero-padding, e.g. by increasing the fft-length to $R = 6 \cdot N_a$ or even higher? Motivate shortly your answer.

d) Now use again $f_1 = 185$Hz and $f_2 = 200$ [Hz]. Calculate the minimum length $N_b$ of data samples that we need in order to be able to distinguish the two spectral peaks of the signal $x[n]$ and plot the resulting, approximated, FTD-spectrum of $\tilde{x}[n]$.

e) Use again $f_1 = 185$Hz and $f_2 = 200$ [Hz] but use a Hanning window. Calculate the minimum length $N_c$ of data samples that we need in order to be able to distinguish the two spectral peaks of the signal $x[n]$ for this case and plot the resulting, approximated, FTD-spectrum of $\tilde{x}[n]$. Compared this result with the result that has been obtain when using a rectangular window. Explain shortly the difference with respect to:

- Spectral leakage (more or less)
- Amplitudes of the spectral peaks (higher or lower)

*Note: The 3 dB bandwidth of a Hanning window equals $1.44|\frac{2\pi}{N}|$ .*

f) Use again $f_1 = 185$Hz, $f_2 = 200$ [Hz], but now with amplitudes $a_1 = 1$ and $a_2 = 0.35$. When using $N_c$ data samples and a Hanning window make a plot of the, approximated, FTD-spectrum of $\tilde{x}[n]$. Is it still possible to distinguish the spectral peaks? Give a short explanation of the result.

## 2.4 Circular vs linear convolution

Linear convolution ($*$) and circular convolution ($\circledast$) are fundamentally different operations. However, as depicted in Figure 5, there are conditions under which linear and circular convolution are equivalent. Knowing the conditions under which linear and circular convolution are equivalent allows you to use the DFT (or the FFT, its efficient equivalent) to efficiently compute linear convolutions.

**Assignment 15: Linear and circular convolution**

In this assignment you are given the following two finite length 4 sequences: $x[n] = 0$ except $x[0] = x[1] = x[2] = 1$ and $x[3] = \frac{1}{2}$ and $h[n] = 0$ except $h[0] = 1$, $h[1] = \frac{3}{4}$,$h[2] = \frac{1}{2}$,$h[3] = \frac{1}{4}$. The periodic extended sequences of length 4 are denoted by $x_p[n]$ and $h_p[n]$ respectively.

a) Evaluate by hand both the linear convolution result $y[n] = x[n]*h[n]$ and the length 4 circular convolution result $y_p[n] = x_p[n] \circledast h_p[n]$.

b) Implement and plot in MATLAB the length 4 circular convolution result $y_p[n] = x_p[n] \circledast h_p[n]$ by using DFTs.

c) Implement and plot in MATLAB the linear convolution result $y[n] = x[n] * h[n]$ by using DFTs.

## 2.5 Frequency plots

The frequency response of an ideal LPF with cutoff frequency $\theta_c = \frac{\pi}{2}$ is as follows:

$$H(\mathrm{e}^{\mathrm{j}\theta}) = \begin{cases} 1 & \text{for } |\theta| <= \pi/2 \\ 0 & \text{for } \pi/2 < |\theta| \leq \pi \end{cases}$$

From this equation we can calculate the impulse response via the IFTD as follows (verify yourself this result):

$$h[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\mathrm{e}^{\mathrm{j}\theta}) \cdot \mathrm{e}^{\mathrm{j}\theta n} \mathrm{d}\theta = \frac{\sin(\frac{\pi}{2}n)}{\pi n} = \frac{1}{2} \cdot \mathrm{sinc}\left(\frac{\pi}{2}n\right).$$

In order to be able to realize this impulse response in practice we have to limit the infinite range of $h[n]$, which can be done by applying an appropriate length $N$ window $w[n]$, which results in:

$$\tilde{h}[n] \quad = \quad \begin{cases} w[n] \cdot h[n] & \text{as } -\frac{N}{2} \leq n \leq \frac{N}{2} \\ 0 & \text{others} \end{cases}$$

Furthermore we have to apply a shift operation in order to make the filter causal.

**Assignment 16: Frequency plots of LPF**

Implement in MATLAB the causal version of the impulse response of the LPF $\tilde{h}[n]$ for the following three different lengths $N = 5$, 11, and $N = 101$.
Choose a rectangular window $w_R[n]$. Plot the amplitude and phase spectrum of the three filters using the function `freqz` . What can you say about the influence of $N$ on the frequency response of the filter? Explain the behavior of the phase response (e.g. explain the angle and the different phase jumps).

**MATLAB tips:** Use functions `freqz` and `sinc`. Beware! The MATLAB `sinc(x)` function is defined as `sinc(π x)`

## 2.6 Using FDATool

By using the 'Filter Design and Analysis Tool' (FDATool) in MATLAB, it is possible to design and analyze both FIR and IIR filters. A major advantage of IIR filters is that they usually have a lot lower order.
With 'FDATool', it is possible to design different types of FIR filters. Some examples are:

- **Equiripple:** minimize the difference between the desired frequency response and the real frequency response

- **Least-squares:** linear phase FIR filter that minimizes the square error of a piecewise linear function with the actual frequency response
- **Window (with various types of windows)** instead of a block, for low pass filters, a window is used, which ensures a smooth transition. Examples of windows are:
  - Hamming
  - Hanning
  - Kaiser
  - Bartlett

Besides such filters, it is also possible to choose the structure, such as: 'Direct Form 1' of 'Direct Form 2'.

**Assignment 17: Filter design for audio signals 1**

Given the audio file 'audio_sin.wav'. This audio file contains a disturbing sinusoidal signal with frequency '$f_u$'.

a) Determine the absolute frequency '$f_u$ 'of the disturbing sinusoidal signal. To read the signal in MATLAB use: `[y, fs, nbits] = audioread('audio_sin.wav ')`. The samples of the sound are stored in a vector in MATLAB. You can check whether the file is read by the playback function: `audioplayer (y, fs)`.

b) Use the calculated frequency $f_u$ in order to design a filter in such a way that the output of the filter does not contain this disturbing frequency any more, while leaving all the other frequency components in tact as much as possible.
Apply in MATLAB this filter to the recorded audio signal and verify if the frequency $f_u$ has vanished from the output. Upload your resulting MATLAB code and audio file.

**MATLAB tips:** Use functions `audioread`, `audioplayer`, `play`, and `fdatool`. Beware! Function `audioplayer` creates an object. The object can be played back with function `play` (check the MATLAB help file). Use 'export' command to get a required filter from `fdatool`.

## Hand in your report Lab2 of the course 5ESC0

# 3 Lab 5ESC4: Stochastic Signal Processing

## 3.1 Samples of more variables

In the following experiment we construct a random variable $u_N[n]$ as the normalized sum of $N$ IID stationary random processes $x_i[n]$ as follows:

$$u_N[n] = \frac{1}{N}\sum_{i=1}^{N} x_i[n] \quad \text{with } N = 1, 2, \cdots \tag{17}$$

The random processes $x_i[n]$, for $i = 1, 2, \cdots, N$, are $N$ IID stationary Gaussian random processes with mean $E\{x_i[n]\} = \mu_{x_i} = \mu_x = 0$ and variance $E\{x_i^2[n]\} - E\{x_i[n]\}^2 = \sigma_{x_i}^2 = \sigma_x^2 = 1$ for $i = 1, 2, \cdots, N$.

**Assignment 18: Mean and variance**

a) Calculate an analytical expression for the mean $\mu_{u_N} = E\{u_N[n]\}$ and the variance $\sigma_{u_N}^2 = E\{u_N^2[n]\} - E\{u_N[n]\}^2$ of the random process $u_N[n]$ as a function of $N$.

b) Use MATLAB to generate the random processes $x_i[n]$, for $i = 1, 2, \cdots, 5$. Generate for each of these random processes 1000 IID samples with $E\{x_i[n]\} = \mu_{x_i} = \mu_x = 0$ and $E\{x_i^2[n]\} - E\{x_i[n]\}^2 = \sigma_{x_i}^2 = \sigma_x^2 = 1$. Construct $u_N[n]$ for $N = 1, 2$ and $N = 5$ and make a *subplot* of $u_1[n], u_2[n]$ and $u_5[n]$.

c) Does the plots of assignment b) follow the analytical results that you found in answer a)? Can you give a general statement about the mean and variance of a random process which consists of the normalized sum of $N$ IID random processes?

**MATLAB tips:** Use functions `normrnd` and `subplot`.

In the following experiment we will examine the relationship between so called scatter plots for pairs of random processes and their correlation coefficient. We will see that the correlation coefficient determines the shape of the scatter plots.
Let $x_i[n]$, for $i = 1, 2$, be two IID stationary Gaussian random processes with zero mean and unit variance, thus $E\{x_i[n]\} = \mu_{x_i} = \mu_x = 0$ and $E\{x_i^2[n]\} = \sigma_{x_i}^2 = \sigma_x^2 = 1$ for $i = 1, 2$. The random process $y_N[n]$ is defined as follows:

$$y_N[n] = \frac{x_1[n] + (N-1)x_2[n]}{N} \quad \text{with } N = 1, 2, \cdots \tag{18}$$

Notice that since $y_N[n]$ is a linear combination of two Gaussian processes, $y_N[n]$ will also be Gaussian.

**Assignment 19: Mathematical expressions for variance and correlation coefficient**

a) Calculate, and make a MATLAB plot as function of $N$, of the mathematical expression for the variance $E\{y_N^2[n]\} - E\{y_N[n]\}^2 = \sigma_{y_N}^2$.
*Note:* In order to calculate the variance $\sigma_{y_N}^2$, you first have to calculate the mean $E\{y_N[n]\} = \mu_{y_N}$. Furthermore the x-axis of the plots should represent the value of $N$ in the range $1, 2, \cdots, 100$.

b) Calculate, and make a MATLAB plot as function of $N$, of the mathematical expression for the normalized cross covariance coefficient

$$\rho_{x_2,y_N}[0] = \frac{E\{(x_2[n] - \mu_x)\cdot(y_N[n] - \mu_{y_N})\}}{\sigma_{x_2}\cdot\sigma_{y_N}}$$

Now use MATLAB to generate 1000 IID samples of $x_1[n]$ and $x_2[n]$ and construct with these samples $y_N[n]$ for $N = 1, 2, 5$ and $N = 100$.

**Assignment 20: Scatter plots and empirically evaluate correlation coefficient**

a) Generate a scatter plot of the ordered pairs of samples $(x_2[n], y_N[n])$, for $N = 1, 2, 5$ and $N = 100$. Do this by plotting, for each new value of $N$, the points $(x_2[1], y_N[1]), (x_2[2], y_N[2]), \cdots, (x_2[1000], y_N[1000])$. In order to plots points without connecting them with lines, use the plot command with the '.' format. Use the command subplot(2,2,p) (with p=1,2,3,4) to plot the four cases for $y_N[n]$ in the same figure. Be sure to label each plot using the title command.

b) Empirically compute an estimate of the normalized cross correlation coefficient $\hat{\rho}_{x_2,y_N}[0]$ using the samples $x_2[n]$ and $y_N[n]$ with the following formula:

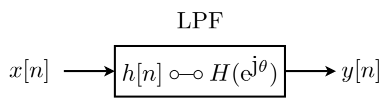
$$\hat{\rho}_{x_2,y_N}[0] = \frac{\sum_{n=1}^{1000} (x_2[n] - \hat{\mu}_{x_2})(y_N[n] - \hat{\mu}_{y_N})}{\sqrt{\sum_{n=1}^{1000} (x_2[n] - \hat{\mu}_{x_2})^2 \sum_{n=1}^{1000} (y_N[n] - \hat{\mu}_{y_N})^2}}$$

Compare these numerical estimates $\hat{\rho}_{x_2,y}[0]$ with the theoretical results $\rho_{x_2,y_N}[0]$ of the previous assignment.

c) Explain how the scatter plots are related to normalized cross correlation coefficient.

## 3.2 Autocorrelation and power spectral density

In this section, we will filter an input signal, which is a discrete random process, to generate an output signal, which is a new discrete random processes. We will study the statistical properties of the output process and its relation with the input process and the filtering operation. Fig. 7




Figuur 7: LTI filter

shows a system in which the input signal $x[n]$ is a white Gaussian random process with zero mean and unit variance. This input signal $x[n]$ is used as the input to a Low Pass Filter (LPF), from which the impulse response is denoted by $h[n]$ and the frequency response by $H(e^{j\theta})$.

### 3.2.1 Analysis and empirical results for simple LPF case

In first instance we use the following very simple LPF

$$h[n] = \frac{1}{3}(\delta[n] + \delta[n-1] + \delta[n-2]) \tag{19}$$

Thus, in this case the output random process $y[n]$ can be described by the following difference equation:

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2]) \tag{20}$$

which implies that the output samples are averaged values of 3 succeeding input samples.

**Assignment 21: Correlation function and power spectral density (PSD) function**

a) Derive, by hand, the autocorrelation $r_y[l] = E\{y[n]y[n-l]\}$ of the random process $y[n]$.

b) Derive, by hand, the PSD $P_y(e^{j\theta})$ of the random process $y[n]$ in two different ways, namely first as the FTD of $r_y[l]$ and second as $P_y(e^{j\theta}) = P_x(e^{j\theta}) \cdot |H(e^{j\theta})|^2$.

Now use MATLAB to generate 1000 IID samples of $x[n]$ and use these samples to generate the samples of $y[n]$.

**Assignment 22: Scatter plots**

Generate 4 scatter plots using subplot(2,2,p) (with p=1,2,3,4). The first scatter plot should consist of points $(y[n], y[n+1])$ $(n = 1, 2, \cdots, 900)$. Notice that this correlates samples that are separated by a 'lag' of 1 sample. The other 3 scatter plots should consist of the points $(y[n], y[n+2])$, $(y[n], y[n+3])$, $(y[n], y[n+4])$ (all for $n = 1, 2, \cdots, 900$). What can you deduce about the random process $y[n]$ from these scatter plots?

In most real applications the following expression to estimate the autocorrelation is used:

$$\hat{r}_y[l] = \frac{1}{M} \sum_{n=0}^{M-|l|-1} y[n]y[n+|l|] \quad \text{for } -(L-1) \leq l \leq L-1 \tag{21}$$

where $M$ is the number of used samples of the sequence $y[n]$ and $L \ll M$. In the next assignment we take $L = 11$.

**Assignment 23: Empirical correlation and PSD function**

a) Use MATLAB to calculate $\hat{r}_y[l]$. Make one stem plot with the theoretical values $r_y[l]$ (use the result from assignment 21) and the estimated values $\hat{r}_y[l]$ versus $l$ for $-10 \leq l \leq 10$.

b) For what value of lag $l$ does $r_y[l]$ reach its maximum and for what value of lag $l$ does $\hat{r}_y[l]$ reach its maximum?

c) Think of a procedure to obtain an estimate of the PSD that can be derived from the calculated values $\hat{r}_y[l]$. Apply this procedure in MATLAB to calculate an estimate of the PSD $\hat{P}_y(e^{j\theta})$ for 100 values of $\theta$ in the range $-\pi < \theta \leq \pi$. Finally make one plot with the theoretical (use the result from assignment 21) and estimated values of the PSD.

d) Give a short reasoning of possible differences between the theoretical and estimated values of the PSD.

## 3.3 Cross correlation of two random processes

Cross-correlation of signals is often used in applications where we need to estimate the distance to a target (e.g. in sonar and radar). In a basic set-up, a zero-mean signal $x[n]$ is transmitted, which then reflects off a target after traveling for $\tau/2$ seconds. The reflected signal is received, amplified, and then digitized to form $y[n]$. If we summarize the attenuation and amplification of the received signal by the constant $\alpha$, then we can model the received signal as follows:

$$y[n] = \alpha x[n-\tau] + w[n] \tag{22}$$

where $w[n]$ represents additive noise from the environment and receiver electronics. Furthermore, we assume that $x[n]$ and $w[n]$ are uncorrelated zero-mean random variables. In order to compute the distance we need to compute the delay $\tau$. We can do this by using the cross-correlation $r_{xy}[l] = E\{x[n]y[n+l]\}$.

**Assignment 24: Expression for cross-correlation**

a) Give a short derivation in which you show the following relation:

$$r_{xy}[l] = \alpha r_x[l-\tau] \tag{23}$$

b) From this result, describe shortly a procedure to estimate the delay $\tau$ based on measurements of the cross-correlation.

Write a MATLAB function r=crosscor(x,y,l) to compute the lag $l$ sample cross-correlation between two discrete-time random processes, $x[n]$ and $y[n]$. In order to test your function, generate two length 1000 sequences of zero-mean Gaussian random variables, denoted as $x[n]$ and $z[n]$. Then compute a new sequence $y[n] = x[n] + z[n]$.

**Assignment 25: Test cross-correlation function**

a) Use your MATLAB function crosscor(x,y,l) to calculate the sample cross-correlation for lags $-10 \leq l \leq 10$ and plot the result.

b) Which value of $l$ produces the largest cross-correlation? Why?

c) Is the cross-correlation function an even function? Why or why not?

Next we will do an experiment to illustrate how cross-correlation can be used to measure time-delay in radar applications. Download the MAT file radar.mat and load it into MATLAB by using the load command. The vectors trans and received contain two signals corresponding to the transmitted and received signals for a radar system.

**Assignment 26: Estimate delay for radar data**

a) Plot the transmitted signal and the received signal on a single figure using subplot. Can you estimate the delay $\tau$ by visual inspection of the received signal?

b) Use your MATLAB function crosscor(x,y,l) to calculate the autocorrelation of the signal trans for the lags $-100 \leq l \leq 100$. Next, compute the sample cross-correlation between the signal trans and received for the range of lag values $-100 \leq l \leq 100$, using again your MATLAB function crosscor(x,y,l). Plot both autocorrelation and cross-correlation for the lags $-100 \leq l \leq 100$.

c) Based on your proposal to estimate the delay , determine the delay $\tau$.

## Hand in your report Lab3 of the course 5ESC0