AGH	Faculty of Mechanical Engineering and Robotics Department of Robotics and Mechatronics	R&M
Mechatronic Systems Identification		
Lab 4: Matched filter		

1. Cross-correlation

Prepare a script that will create a signal **y** with the sampling frequency of 2000 Hz and the length of 2000 samples. Initially, create an empty signal vector (e.g. with the function zeros(s,1), where s is the length of the signal). In the second step, create a chirp signal **yp** (variable frequency) using Matlab's chirp function. The signal should be described as follows:

```
yp = chirp(tp,f1,tp(end),f2);
```

where **tp** is the time vector corresponding to the new signal, **f1** is the frequency at time **tp(1)** and **f2** is the frequency at time **tp(end).** Then multiply the chirp signal with a Hanning window of the same length. Based on this signal, do the following tasks:

Ex. 1.1

Give the signal yp a length of 200 samples, a start frequency of f1 = 10 Hz, and an end frequency of f2 = 300 Hz. Add the yp signal to the y signal starting from the location of the 500th sample. This can be done using the following syntax:

```
y(500:500+length(yp)-1) = yp;
```

Next, compute the correlation between the y signal and yp with the xcorr function using the following syntax:

```
[yc,lag] = xcorr(y,yp);
```

Display the received signal **yc** and the signal **y** on the time-domain graph (use the hold on function). The lag vector represents the lags given in the samples. So it can be converted to a time vector as follows:

```
tc = lag/fs;
```

Describe the difference between the two signals and what causes it. For better comparison, you can normalize the amplitude of both signals. Then try the same operation with convolution (conv function), taking into account the relationship between correlation and convolution. Compare waveforms.

Ex. 1.2

Calculate the correlation using the following parameter:

- Length of **yp** 200 samples, **f1 =** 30 Hz, and the following parameters **f2 =** 100 Hz, 200 Hz, 500 Hz.
- Length of **yp** 200 samples, **f1 =** 500 Hz, and **f2 =** 500 Hz.

Create a graph for the signal correlation in each of the combinations of the above parameters. Discuss the relationship between the length of the time base of the correlation result and the frequency band of the signal **yp.**

Ex. 1.3

To the signal **y**, add a **yp** signal consisting of 1000 samples, f1 = 50 Hz, f2 = 500 Hz at two time instances: starting from the 500th sample with an amplitude of 1 and from the 750th sample with an amplitude of 0.5. Then perform the correlation between the newly created y signal and the **yp** signal. Read out for which time instances the pulses occurred as a result of the correlation occur and what is the delay between them. How do the pulse locations relate to the location of the **yp** signals in the **y** signal? What are the amplitudes of the received pulses?

Ex. 1.4

To the y signal created in Problem 1.3, add the white noise of the standard deviation **amp** generated by the **randn** function**:**

```
szum = amp*randn(length(y),1);
```

Correlate the signal **y** with the **yp** signal assuming noise with amp values of 0.5, 1, and 2. Display the signals in the time domain for each case. Comment for which noise levels peaks are seen in the correlated signal.

The report should include the Matlab code used to perform the above tasks.

From Task 1.1, consider the graph showing the original signal and the signal correlated with the chirp signal. The effect of the correlation on the given signal and the differences in the implementation using the xcorr and conv functions should be discussed.

From task 1.2, graphs showing the correlation for signals with different frequency ranges should be taken into account. Comment what is the dependence of the width of the compressed signal on the changed parameters of the yp signal.

From task 1.3, attach the graph of the created signal y and the result of the correlation with yp. The comment should specify the distance between the two compressed pulses and describe how the matched filter helps in identifying the given signal.

From Task 1.4, attach the results of signal y correlation for different noise amplitudes. In the commentary, it is necessary to discuss the efficiency of filtration against a noisy signal.

2. MAtched filter

Task 2.1

In this task, you need to analyze the recording of a piece of music in the form of an **audio.wav** file. In the recording, synthetic signals in the form of sinusoids with increasing frequency were placed on both channels. The aim of the exercise is to locate the pulses in the recording. The standard pulse is in the mat file named Pulse.mat. Time waveforms from the recording should be read using the audioread function from Matlab:

```
[data,fs] = audioread(filename);
```

The variable **data** should be a matrix containing two channels of the audio recording, and **fs** corresponds to the sampling frequency of the signal. Play the recording and check if the pulses in the signal are audible. Then, using matched filtering, try to determine in which channel and at what time the given signal is located.

Task 2.2

Perform the above problem in the frequency domain. To do this, apply the complex coupling of the Fourier transform of the pulse **ypf**, which should be multiplied by the recording in the frequency domain **yf**:

```
yf_inv = yf.*conj(ypf.')
```

It is also necessary to use the so-called zero-padding to get the Fourier transform of a pulse signal with the same length as the audio signal:

```
y = fft(X,N);
```

where **X** is the time signal and **N** is the target length of the Fourier transform signal. In order to obtain the correlation result in the time domain, the inverse Fourier transformation of the **yf_inv** vector should be performed using the **ifft** function.

The report should include the Matlab code used to perform the above tasks. A graph showing the waveforms on both channels and the result of matched filtering should also be included. On the basis of the obtained graphs, it should be determined at which time moments in the recording there are impulses. Finally, apply matched filtering in the frequency domain and compare the results with the correlation in the time domain.