ISEP
Telecom & Images Laboratory

T&WN

DIGITAL SIGNAL PROCESSING
Case Study

Table of contents

This lab lasts 8 hours. You are expected to resolve a particular case study. You will use Matlab for all your simulations. *Sptool* may be an useful interactive tool to observe signals, their Fourier transform, and to design digital filters if needed.

At the end of your study, you have to present your results to the teachers, explaining your system and justifying it (method and parameters).

To prepare this lab, you should review some basic points related to signal processing: spectrum analysis, digital filters, power estimation, …

Using Sptool

In this example, we design a digital filter and we filter a chirp signal. We can see the signal spectrums before and after filtering.

Interactive FIR filter design

- Open Matlab.
- Open " *sptool* " from Matlab command window by typing `sptool`

To create a filter, press `New` in the `Filters` panel. A default filter is generated and displayed in the Filter Designer.
Enter a sampling frequency equal to 10 kHz and select `Equiripple FIR filter` as filter design algorithm . Then fill in the specification fields:

Minimum order checked
Type : lowpass

Fp:1000	(passband edge frequency)
Rp : 3	(stopband edge frequency)
Fs : 1500	(passband ripple)
Rs : 40	(stopband attenuation)

Click on the Frequency graph, then press `Apply` if `Auto Design` is disabled. Note that when `Auto Design` is enabled, the plot updates whenever you change a filter specification.

Sptool plots the magnitude response of the designed filter. In our example, we have designed a lowpass filter, with a bandpass 0-1KHz, a bandstop 1.5KHz-5KHz. The passband ripple is equal to 3dB (-1.5dB to 1.5dB) and the stopband attenuation is 40dB. Note that the definition of the passband ripple in Matlab is different from the common one. Generally, it represents the maximum ripple value (1.5dB in this example).

Now, activate again the `sptool` window and press `view` in the `Filters` panel. This opens a new window that allows you to analyse precisely the filter you have just designed. You can view the magnitude response, the phase response, the group delay response, the impulse response, the filter coefficients, the poles and zeros: press the corresponding icon or use the analysis menu. You can also change the display characteristics (zoom).

Importing and viewing signals

We are now going to generate a chirp signal and to pass this signal in the filter we have designed.

- Activate the Matlab command window.
- generates samples of a linear swept-frequency signal
  (see `help chirp` to understand these command lines):

```
t=0:0.0001:2;
y=chirp(t,0,1,2000);
```

Now we import this signal. Activate the `SPtool` window, click `Import` from the `File` menu to open the `Import` window. You can import a signal, filter, spectrum into SPTool from either the workspace or from a file.

In this example, we import the chirp signal from the workspace. Click the `From Workspace` radio button. The contents of the Matlab workspace are displayed. Select the signal data type from the `Import As` popup menu: `Import as Signal`. Select the variable y and press the → button: the selected variable is transferred to the `Data` field; finally, specify the sampling frequency and type a name into the `Name` field (for example sig1). Press OK. The SPTool windows is activated again.

We can now use the `View` button to make the Signal Browser active and view the imported signals. In the Signals panel:

- Select sig1
- Press the `View` button.

Filtering signals

We are now going to apply our filter to the chirp signal. Select sig1 in the `Signals` panel and filt1 in the `Filters` panel. Press the `Apply` button. Type the name of the output signal (sig2 for example) in the Output Signal field (`Apply Filter` window) and press OK.

Spectrum analysis

We can now analyse the spectrum of the chirp signal before filtering (sig1) and after filtering (sig2).

Activate the `SPTool` window, select sig1 from the `Signals` list and press `Create` in the `Spectra` panel. No spectrum is computed or displayed yet. Use the default parameters in the `Spectrum Viewer` or modify the parameters if necessary, and press `Apply` to compute the spectrum. The newly created spectrum has a default new nem (`spec`*n*).

Repeat this process to view the spectrum of the signal after filtering.

Exporting

It may be useful to export a filter or a signal to the Workspace, in order to use it in a command file. Activate the `Sptool` window, click `Export` from the `File` menu to open the `Export` window. You can export a signal, filter, spectrum from SPTool to either the workspace or a file. Select the variable you want to export, for example *filt1*, and press the `export to workspace` button.

Select again the Matlab command window and type the line command `who`. *filt1* is in the list of the current variable. Now type *filt1*. You can see:

```
filt1 =

    tf: [1x1 struct]
    ss: []
   zpk: []
   sos: []
   imp: []
  step: []
```

```
            t: []
            H: []
            G: []
            f: []
        specs: [1x1 struct]
           Fs: 1000
         type: 'design'
     lineinfo: []
SPTIdentifier: [1x1 struct]
        label: 'filt1'
```

`filt1` is a structure. For example, `filt1.tf.num` store the filter transfer function. Now type `filt1.tf.` This is also a structure :

`filt1.tf.num` is a vector storing the coefficient of the numerator of the transfer function.

`filt1.tf.den` is a vector storing the coefficient of the denominator, (1 for a FIR filter).

It may be also interesting to export a filter or a signal to the disk. You can do it in the same way, just press the `export to disk` button instead of the `export to workspace` button. Enter a name for the *.mat* file, for example filt1.mat. To import filter again to the workspace, select the Matlab command window and use the `load` command. For example: `load filt1`. You can also use Sptool: click `Import` from the `File` menu to open the Import window. Click the `From disk` radio button. You can either enter a *.mat* file name or click `Browse` to open a file listing and select a *.mat* file.

Case Study

1. DTMF specifications

DTMF (which stands for dual tone multifrequency) is the usual technique used for dialing on the PSTN. Each dialing digit is coded by the addition of two sinusoidal signals of different frequencies. The table below summarizes this coding:

	1209 Hz	**1336 Hz**	**1477 Hz**	**1637 Hz**
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

The duration of a dialed digit is always longer than 65 ms, and the delay between two dialed digits (no signal) is longer than 80ms.
The DTMF signals are sampled at 8KHz and encoded by 8 bits.
The frequency tolerance is equal to 1.5%. For example, the frequencies for a ‘1’ are:697 ± 10.455 Hz, 1209 ± 18.135 Hz.
The signal to noise ratio is more than 20dB.
The dial tone is a sinusoidal signal. Its frequency is equal to 440Hz.

2 Work to realize

- Use Matlab to realize a program detecting the dial tone and decoding the DTMF signals. In other words, your program takes as input a signal corresponding to a dialing, and gives as output the corresponding phone number. You can test your program on the .wav files from the c:\TP\TPTNS directory. Theses files are recordings of dialing signals.

- Justify your method and all the parameters used in your program.

- Show the results to the teachers (short explanation and demonstration)