Speech signal processing using MATLAB

Assignment 1 (May 25, 2016)

Each task below requires you to write a small MATLAB script. Please create a separate script file for each of these and name it appropriately. Submit your results by uploading all files using the Moodle infrastructure until Tuesday, June 7.

MATLAB quick guide

- script files can be created inside MATLAB using the menu "File → New → Script"
- save a script file by using "File → Save as…" and run it in MATLAB's command line window by simply typing its name
- when computing values use unique variable names beginning with a letter
- the content of any variable can always be inspected by double-clicking its name in MATLAB's Workspace window ("Desktop → Workspace")
- code lines terminated by a semicolon (;) produce no output in command line window, whereas lines without terminating semicolon do:

```
>> a = sqrt( 2 ); % will give no output
>> b = sqrt( 2 ) % will give 1.4142 as output
```

- code lines starting with a percent sign (%) are comments and will be ignored by MATLAB (please try to comment your submissions well)
- plots can be generated using the following code snippet, e. g. plotting amplitude values `xi` over time values `ti` and labeling axes:

```
>> figure(); % create a figure
>> plot( ti, xi ); % plot data
>> xlabel( 'time' ); % set x-axis label
>> ylabel( 'amplitude' ); % set y-axis label
```

- plot range can be limited on both x- and y-axis:

```
>> xlim( [0, L] ); % limit x-range to 0...L
>> ylim( [0, 1000] ); % limit y-range to 0...1000
```

- you can search for MATLAB commands by running the command

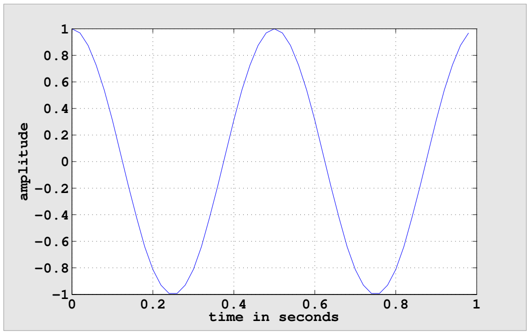
```
>> help SEARCHTERM
```

- also refer to the lecture's example scripts provided on Moodle
- an additional online resource is available on http://www.tutorialspoint.com/matlab/matlab_quick_guide.htm

Task 1, Sampling (5 points)

1. Create a continuous cosine signal with frequency $f = 2\,\text{Hz}$ and length $L = 1\,\text{s}$.
2. Discretize this signal (time and amplitude values) using a sampling rate of $f_S = 50\,\text{Hz}$ and $n_S = 8$ bits.
3. Plot the waveform of that discrete signal.

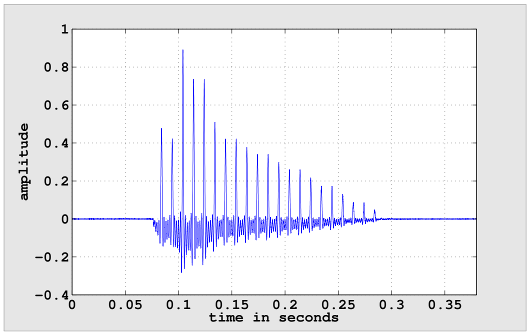
The result should look similiar to this:

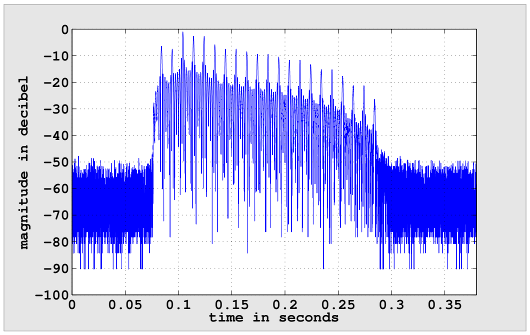


Task 2, Time domain (10 points)

1. Read the content of the wave file `https://github.com/murtex/spl/raw/master/matlab/sound.wav` (download it before).
2. Compute the discrete time values `ti` for this recording.
3. Compute the total energy and average power of this recording.
4. Convert recording's amplitude to magnitude values (using `abs( xi )`) and further to decibel values.
5. Plot the recording's waveform, both on linear and logarithmic (decibel) scale.

The result should look similiar to this:





Task 3, Frequency domain (10 points)

1. Read the content of the wave file `https://github.com/murtex/spl/raw/master/matlab/sound.wav` (download it before).
2. Fourier transform that signal.
3. Compute the discrete frequency values including negative ones.
4. Compute the power spectrum and restrict it to positive frequencies.
5. Convert linear power values to logarithmic (decibel) scale.
6. Plot the linear and logarithmic power spectrum. Limit the frequency range to $0 \ldots 1000$ Hz.

The result should look similiar to this: