



Politechnika Łódzka  
Instytut Elektroniki

# **SIGNAL PROCESSING**

Laboratory #3:

**Importing Python packages;  
storing/loading: text, binary, Matlab  
and wave files**

*M. Kociński, P. Strumiłło*

Medical Electronics Division  
Institute of Electronics

## PURPOSE:

To learn how to import Python packages and how to load and save text, binary, Matlab and wave files files.

## TASKS:

1. Learn four ways of importing Python packages inside scripts. Please create and run the following scripts in your Python folder: *sinus1.py*, *sinus2.py*, *sinus3.py*, *sinus4.py*

```
#-----  
# sinus1  
import pylab  
import numpy  
  
x=numpy.arange(100)  
y=numpy.sin(x*numpy.pi/50.)  
  
pylab.plot(x,y)  
pylab.show()
```

```
#-----  
# sinus2  
import pylab as pl  
import numpy as np  
  
x = np.arange(100)  
y = np.sin(x*np.pi/50.)  
  
pl.plot(x,y)  
pl.show()
```

```
#-----  
#sinus3  
from pylab import plot, show, title  
from numpy import arange, sin, pi  
  
x=arange(100)  
y=sin(x*pi/50)  
  
plot(x,y)  
show()
```

```
#-----not recommended -----  
#sinus4
```

```
from pylab import *  
from numpy import *
```

```
x=arange(100)  
y=sin(x*pi/50)
```

```
plot(x,y)  
show()
```

## 2. Loading and saving files in ASCII format (the so called text files)

```
#-----  
#load/read data sets to/from file in ASCII (American Standard Code for Information  
Interchange) format  
#-----
```

```
#This script can be used to save several (the same sized) arrays to one file  
import numpy as np  
x = np.arange(100)
```

```
y1 = np.sin(x*np.pi/50)  
y2 = np.cos(x*np.pi/50)
```

```
np.savetxt('my_sin_cos.txt', (x,y1,y2),fmt='%0.2f')
```

```
#-----  
#This script can be used to load into Python workspace several (the same sized) arrays from one  
file  
import numpy as np
```

```
# load from file - case 1  
xx, yy1, yy2 = np.loadtxt('my_sin_cos.txt')
```

```
#draw loaded data  
#we can import some modules (modules or function from modules) also in the middle of the file  
from pylab import plot, show, title, figure  
figure(1)  
plot(xx,yy1,xx,yy2)  
title('Loaded data - case 1')
```

Please note that saving and loading files takes place from the current directory.  
For loading/saving files from other directories you need to specify a path to a file, e.g.  
`xx, yy1, yy2 = np.loadtxt('d:/Biomed2012_1/my_sin_cos.txt')`

3. Loading and saving binary files in Python native formats *\*.npy* and *\*.npz*

*#The simplest possibility is to use numpy's binary file format.  
#See numpy.save, numpy.savez and numpy.load.*

*#-----  
# save a single array  
#-----  
import numpy as np*

*b = np.arange(20.)  
np.save('b.npy', b)  
print b*

*b2 = np.load('b.npy')  
print b2*

*#-----  
# several arrays in a single .npz file  
#-----*

*cc = np.arange(5)  
dd = np.arange(4,15,2.5)  
ee = np.arange(100,110,2)  
print cc,dd,ee*

*# names: a,b,c can be any valid key names  
np.savez('multi\_array.npz',a=cc, b=dd, c=ee)*

*#load from file  
multi = np.load('multi\_array.npz')*

*# for a list of arrays type in:  
multi.files*

*# to select and print individual arrays type in:  
print multi['a']*

*# to assign to a new array  
zz = multi['b']*

#### 4. Loading and saving binary files Matlab files \*.mat

```
#####  
#### Matlab files - loadmat, savemat ####  
#####  
# from documentation: www.scipy.org/Cookbook/Reading\_mat\_files  
# and http://docs.scipy.org/doc/scipy/reference/tutorial/io.html  
  
#loading .mat files method#1 good for loading many variables stored in one mat file  
  
from scipy.io import loadmat, savemat  
from numpy import reshape  
from pylab import plot  
  
    ecg=loadmat('ecg_all.mat')['ecg_s']  
    ecg=reshape(ecg,len(ecg))  
    plot(ecg)  
    show()  
  
#saving data in .mat files  
savemat('ecg_py.mat',{'ecg_new':ecg})
```

#### 5. Loading, saving and playing wave files \*.wav

```
#####  
#### Play audio files (wav) ####  
#####  
  
#playing wave file from Python in Windows OS  
import winsound  
winsound.PlaySound("scale.wav", winsound.SND_ALIAS)
```

```
#####  
####   Loading wave sound files   ####  
#####  
  
#loading wave files into Python  
#-----  
from scipy.io.wavfile import read as read_wav  
  
sampling_rate, data = read_wav('scale.wav')  
print sampling_rate  
print data  
  
#saving wave files from Python  
#-----  
from scipy.io.wavfile import write as write_wav  
  
write_wav('inv_scale.wav',sampling_rate,data[::-1])  
  
winsound.PlaySound("inv_scale.wav", winsound.SND_ALIAS)
```

□ 2012-10-29