



Politechnika Łódzka  
Instytut Elektroniki

# **SIGNAL PROCESSING**

Laboratory #2:

**Data visualization, array operations  
and defining functions in Python**

*M. Kociński, P. Strumiłło, K. Kudryński*

Medical Electronics Division  
Institute of Electronics

## PURPOSE:

To get acquainted with basic data visualization techniques and defining functions in Python.

## TASKS:

1. In this exercise we will use different **NumPy** and **PyLab** methods and functions:
2. In an interactive mode in the PyLab widow type the following series of commands:


```
dt=0.5
t=arange(0,10,dt)           # time scale
a=9.81                      # acceleration
v=a*t                       # velocity
s=(a*t**2)/2.              # distance

plot(t,v,'.')               #plot distance in discrete time instances

title('Velocity')
xlabel('t[s]')
ylabel('v[m/s]')

figure(2)
plot(t,s,'.r')              #plot distance in discrete time instances
title('Distance')
xlabel('t[s]')
ylabel('s[m]')

figure(3),plot(t,s,label='distance') # interpolated distance
legend()
grid()
```

By selecting  icon displayed at the bottom of the figure windows save (in your defined directory) the displayed plots in the jpg format files.

3. Enter the subplot commands (consult `subplot?` or `help(subplot)` commands)

```
figure()
subplot(2,1,1), plot (t,v)    # upper plot
subplot(2,1,2), plot (t,s)    # lower plot
```

4. Create a function `motion(a,t)` which returns the final velocity  $v$  and distance  $s$  of a body moving with acceleration  $a$ , after time  $t$ . Use Notepad++ to define and type in your function. Save the defined function under the name `my_functions.py` where all functions you write are stored. Now you can import your library of functions by using command:

```
from my_functions import *
```

and then run the motion function.

5. Write and test a new function `my_sign(x)` that checks, whether variable  $x$  is positive, negative or zero by printing appropriate texts, i.e. *'positive'*, *'negative'*, *'zero'*.

6. Write and test a new function `my_stat_1d(x)` which returns the minimum, average and maximum values of vector  $x$ .

7. You can also import a single function from `my_functions` in two possible methods:

- a. `from my_functions import my_stat_1d`
- b. `from my_functions import my_stat_1d as myst`

If you use method a. you invoke the `my_stat_1d` function e.g. by a command:  
`my_stat_1d(x)`

If you use method b. you invoke the `my_stat_1d` function e.g. by a command:  
`myst(x)`

□ 2012-10-20