HANDS-ON INTERVENTION:
Vehicle-Manipulator Systems

Lab1: Simulation in Python

Patryk Cieślak
patryk.cieslak@udg.edu

Exercise 1 (5 points)

The goal of the exercise is to build a simple simulation of a 2D harmonic oscillator. Physically the aforementioned oscillator is a point mass moving in the gravity field, attached to a fixed point, through a spring. Moreover, the motion of the object is influenced by a viscous damping force. The output of the simulation is displayed and updated in realtime using the animation capabilities of the *Matplotlib* library. A screen capture of the an exemplary simulation output is presented in Fig. 1.

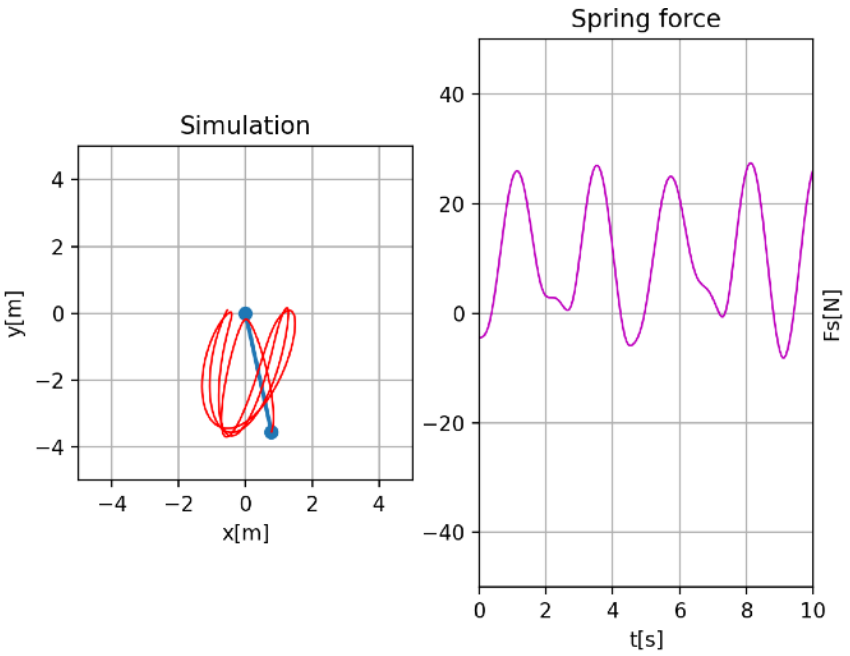

Fig 1. Simulation of a harmonic oscillator in Python

The following elements have to be included in the simulation program:

1) Import of necessary libraries.
2) Definition of states and parameters of the physical model.
3) Definition of parameters of the simulation.
4) Calculation of the net force acting on the object, according to the principles of physics.
5) Realtime update of position and velocity of the object, based on the state of the system, calculated force and sampling time (Euler integration).
6) Visualisation based on one figure with two subplots, updated at each simulation step.
7) First subplot: the oscillator structure and the path of the object in the XY plane.

8) Second subplot: the spring force versus the simulation time.
9) All plots with titles, labeled axes, proper axis limits, grid.
10) Infinitely repeating simulation. One simulation run lasts 10s.
11) Random initialisation of the body position at start of each simulation repeat.

The following elements have to be included in the report:

1) Drawing of the oscillator model, including the forces acting one the body.
2) Equations representing system dynamics.
3) Code of the simulation program (well formatted and commented in detail).
4) Screenshots of a few simulation runs (see Fig. 1).

Exercise 2 (3 points)

The goal of this exercise is to practice implementing classes in Python. Each student has to create a hierarchy of classes including one base class and three subclasses. What the classes will represent is left to the student's imagination (each particular student's imagination:-).

The following elements have to be included in the program:

1) Definition of the base class including: 1 constructor, min. 2 fields, min. 2 methods.
2) Definition of three subclasses including each: 1 constructor, min. 2 new fields, 2 overriden methods.
3) Instantiation of a list of objects based on the subclasses.
4) For loop printing information about all of the objects and results of calling methods on each of them.

The following elements have to be included in the report:

1) Chart that graphically presents the hierarchy of implemented classes.
2) Code of the program (well formatted and commented in detail).
3) Output of the program.

Questions (2 points)

1. How to change the colour of the plot (line) to a custom one? (outside of the standard colours defined through the formatting string)
2. How to add a legend to a plot?
3. What are the main differences between the object-oriented programming and the procedural programming?
4. What are abstract classes and do they exist in Python?