Last update: 01-06-2017

240AR064 - Scientific Python for Engineers

Coordinating unit: 240 - ETSEIB - Barcelona School of Industrial Engineering

Teaching unit: 707 - ESAII - Department of Automatic Control

Academic year: 2017

Degree: MASTER'S DEGREE IN AUTOMATIC CONTROL AND ROBOTICS (Syllabus 2012). (Teaching unit Optional)
MASTER'S DEGREE IN INDUSTRIAL ENGINEERING (Syllabus 2014). (Teaching unit Optional)

ECTS credits: 3 Teaching languages: English

Teaching staff

Coordinator: Perera Lluna, Alexandre

Others: Perera Lluna, Alexandre
Velasco Garcia, Manuel

Opening hours

Timetable: Fridays 15:00-16:00

Prior skills

Knowledge of a programming language

Degree competences to which the subject contributes

Transversal:

CT3. TEAMWORK: Being able to work in an interdisciplinary team, whether as a member or as a leader, with the aim of contributing to projects pragmatically and responsibly and making commitments in view of the resources that are available.

CT4. EFFECTIVE USE OF INFORMATION RESOURCES: Managing the acquisition, structuring, analysis and display of data and information in the chosen area of specialisation and critically assessing the results obtained.

Teaching methodology

This class will be structured in three main tasks:

Lectures: the teachers will expose theoretical and practical contentsr, with the active participation of students.
Challenges: Students are exposed to a problem to be solved in a limited time.
Competitive projects: Problem solving projects where students are placed on a simulated scenario. In this scenario students program a simulated bot employing machine learning algorithms in python.

Final project defense includes an oral exposition of the developed work jointly with a discussion on the related methodology.

Learning objectives of the subject

The goal of the class is to learn skills for scientific programming, focused on the application of advanced machine learning tools on robotics. Students will learn to develop structured and problem solving thinking in a competitive environment.

Last update: 01-06-2017

240AR064 - Scientific Python for Engineers

Study load

Total learning time: 75h	Hours medium group:	27h	36.00%
	Self study:	48h	64.00%

Last update: 01-06-2017

240AR064 - Scientific Python for Engineers

Content

Last update: 01-06-2017

240AR064 - Scientific Python for Engineers

Scientific Python for Engineering	Learning time: 30h Theory classes: 15h Laboratory classes: 3h Guided activities: 5h Self study : 7h
Description:	

Last update: 01-06-2017

240AR064 - Scientific Python for Engineers

Part I
1. Introduction
a. Why python?
b. Python History
c. Installing Python
d. Python resources
2. Working with Python
a. Workflow
b. ipython vs. CLI
c. Text Editors
d. IDEs
e. Notebook
3. Getting started with Python
a. Introduction
b. Getting Help
c. Basic types
d. Mutable and in-mutable
e. Assignment operator
f. Controlling execution flow
g. Exception handling
4. Functions and Object Oriented Programming
a. Defining Functions
b. Decorators
c. Writing Scripts and New Modules
d. Input and Output
e. Standard Library
f. Object-oriented programming
g. Magic Functions
5. Iterators and Generators
a. Iterators
b. Generators
6. Creating Graphic Interfaces (optional)
7. Debugging code
a. Avoiding bugs
b. Debugging workflow
c. Python's debugger
d. Debugging segfaults using gdb
Part II
1. Introduction to NumPy
a. Overview
b. Arrays
c. Operations on arrays
d. Advanced arrays (ndarrays)
e. Notes on Performance (\%timeit in ipython)
2. Matplotlib
a. Introduction
b. Figures and Subplots
c. Axes and Further Control of Figures
d. Other Plot Types
e. Animations
3. Plotting with Mayavi

240AR064 - Scientific Python for Engineers

a. Mlab: the scripting interface
b. Interactive work
4. Advanced Numpy
a. Life of ndarray
b. Universal functions
c. Interoperability features
d. Array siblings: chararray, maskedarray, matrix
e. Summary
f. Contributing to Numpy/Scipy

Part III
1. Scipy
a. Introduction
b. Input/Output
c. Statistics
d. Linear Algebra
e. Fast Fourier Transforms
f. Optimization
g. Interpolation
h. Numerical Integration
i. Signal Processing
j. Image Processing
k. Special Functions
2. Sparse Matrices in SciPy
a. Introduction
b. Storage Schemes
c. Linear System Solvers
d. Others
3. Optimizing code
a. Optimization workflow
b. Profiling your code
c. Speeding your code
4. Sympy
a. First Steps with SymPy
b. Algebraic manipulations
c. Calculus
d. Equation solving
e. Linear Algebra
Part IV
1. Python scikits
a. Introduction
b. scikit-timeseries
c. scikit-audiolab
2. scikit-learn
a. Datasets
b. Sample generators
c. Unsupervised Learning
i. Clustering
ii. Gaussian Mixture Models
iii. Novelty/Outliers Detection
d. Supervised Learning

240AR064 - Scientific Python for Engineers

i. Linear and Quadratic Discriminant Analysis
ii. Nearest Neighbors
iii. Support Vector Machines
iv. Partial Least Squeares
e. Feature Selection
3. Practical Introduction to Scikit-learn
a. Solving an eigenfaces problem
i. Goals
ii. Data description
iii. Initial Classes
iv. Importing data
b. Unsupervised analysis
i. Descriptive Statistics
ii. Principal Component Analysis
iii. Clustering
c. Supervised Analysis
i. k-Nearest Neighbors
ii. Support Vector Classification
iii. Cross validation

Qualification system

Class calification will be obtained a weighted mean comprising a evaluation of the challenges (50%) and the final project (50%).

Regulations for carrying out activities

Depending on the characteristics of the simlulation environment and the bot complexity, the students can do the competition individually or in teams. Students will prepare a project report describint mathematical strategy, code structure and performance metrics.

Bibliography

Basic:

Bressert, Eli. SciPy and NumPy: An Overview for Developers. New York: O'Reilly, 2012. ISBN 978-1449305468.

McKinney, Wes. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. Farnham: O'Reilly, 2013. ISBN 9781449319793.