Ph.D. Program in Information Engineering

Course Catalogue

A.Y. 2020/2021

Rev. 1.5 – 12/3/2021

Revision History

Revisions with respect to the reference version: 1.0 – 15/10/2020

Rev. 1.1 – 22/10/2020

- Syllabus of course "*IE 5. Statistical Learning for Big Data in Medicine*" updated:
  - credits changed from 4 to 5;
  - new topic added: *Practical issues related to predictive model development: dealing with missing values, dealing with unbalanced datasets, ensure model generalization*;

Rev. 1.2 – 26/10/2020

- Syllabus of course "*IE 2. Statistics for Engineers*" updated

Rev. 1.3 – 3/11/2020

- Added course "*IE 25. Communicating Using Quantum Entanglement: Teleportation and the Quantum Internet*"

Rev. 1.4 – 17/11/2020

- Syllabus of course "*TSK 2. Python programming for Scientific Engineering*" updated

Rev. 1.5 – 12/3/2021

- Course "*IE 15. Modeling and Simulation of Complex & Multi-Disciplinary Dynamical Systems*" canceled
- Course "*IE 13. Introduction to Reinforcement Learning*" confirmed and lecture schedule fixed (see Class Schedule)
- Course "*IE 1. Statistical Methods*": lecture schedule fixed (see see Class Schedule)

[back to **Summary**]

Summary

General Information

Courses - Transversal Skills Area

Courses - Information Engineering Area

Mathematical and Statistical Methods

Bioengineering

Electronics

Telecommunications

[back to **Summary**]

Control Theory and Applications

Computer Science

Applied Optics

Course Index by Instructor

Coursework Requirements

The following requirements are valid for Ph.D. Students starting in October 2020 (36° cycle). In summary, Students shall **take courses for a minimum of 20 credits** and shall **attend the seminars proposed by the Ph.D. Program**, following the rules detailed below.

Important note: persisting of the COVID-related emergency may cause changes to the course and seminar schedule and consequent requirements adjustments (typically, deadline postponement). Should this happen, students will be informed in due course of the changes.

Definitions

A **course** is a series of lectures given by an instructor (professor or university researcher), possibly accompanied by laboratory sessions, that includes an assessment of the student knowledge (final exam, graded homework or project, etc.). A course gives credits.

A **seminar**, in the context of the Ph.D. Program, is typically a talk, or a series of talks, given by an academic researcher or an eminent professional, that does not include an assessment. A seminar does not give credits.

Course requirements

- Take Ph.D. courses for **a minimum of 20 credits** by the end of the second year.
- **At least 10** of the above 20 credits shall be earned by the end of the first year.

Specific constraints to earn the minimum of 20 credits of courses:

C.1 **Transversal Skills Area (TSK)**: at least 5 credits should come from courses belonging to the Transversal Skills area (labeled **TSK** in the course Summary).

C.2 **Information Engineering Area (IE)**: students shall earn at least 10 credits by taking courses belonging to the Information Engineering Area (labeled **IE** in the course Summary).

C.3 **External Courses**: up to a maximum of 5 credits may be earned by taking external courses (i.e. courses not included in this catalogue) falling in the following categories:

- Courses appearing in the list of external courses approved by the Executive Board. The list of credited external courses is available on the Ph.D. Program main website.
- Additional external courses might be included into the list after submission of a written request signed by the Student and his/her Supervisor. Only courses including an exam with grading are considered.
- Courses from other Ph.D. School catalogues (provided they include a final exam with grading).
- In order to get credit recognition for external courses, students shall obtain a certificate stating that the student attended the course and successfully passed the exam. Alternatively, the student may fill a Certification of Attendance with the course data and have it signed by the course instructor.

[back to **Summary**]

Seminar requirements

- Attend the **seminars** promoted by the Ph.D. Program (find the list on the website) during the three-year Ph.D. course.
- Attend all the lectures of the **Distinguished Lecturer Series** program offered by the Department during the three-year Ph.D. course.
- Attend at least two modules of the **PhD Educational Week on Transferable Skills 2021.**

Each first-year student must fill a tentative program of study form and upload it using the following link:

https://phd.dei.unipd.it/study-and-research-plan

within November 5th. The program of study may be subsequently modified by submitting a new form no later than June 30th of the second year. Seminars, Distinguished Lectures and PhD Educational Week modules should not be included in the program of study. Please, use the Seminar Certificate of Attendance to collect the signature of the speaker or of a member of the Executive Board attending the event.

Students are expected to attend classes regularly. Punctuality is expected both from instructors and students. Instructors have to report to the Coordinator of the Ph.D. Program students missing classes without proper excuse.

Instructors shall complete student grading within 30 days after the end of lectures and, anyway, no later than August 31st.

[back to **Summary**]

Class Schedule

The class schedule is embedded in the Ph.D. Program Calendar. If you have a Google account, you may visualize the class schedule through the following link:

Class Schedule of 2020/21 PhD Courses for Google Calendar

You may also visualize the class schedule using any browser through the following link:

Class Schedule of 2020/21 PhD Courses

Most classes meet in DEI/D meeting room, located at the 1st floor of DEI/D building, location (4) in the map below, or in Room 318 DEI/G, located at the 3rd floor of DEI/G Building, location (3) in the map below, at the Dept. of Information Engineering, via Gradenigo 6/A, Padova.

Please, always check the class schedule in the calendar to verify the class meeting room.

In order to locate the rooms, you may find helpful the map of the Department buildings:

Map of the Department of Information Engineering

[back to **Summary**]

TSK 1. Entrepreneurship and Technology-based Startups

Course Area: Transversal Skills

Credits: 5 (20 hours)

Instructors: Prof. Moreno Muffatto, Ing. Francesco Ferrati, Dipartimento di Ingegneria Industriale, Università di Padova

e-mail: moreno.muffatto@unipd.it, francesco.ferrati@unipd.it

Topics:

From the idea to the market

- Entrepreneurship attitudes
- What is a startup
- From a research project to an entrepreneurial project
- Market dimension, customers profiles and value proposition
- Development of the product/service concept

Intellectual Property Rights

- Types of IPR (patent, copyright, trademark)
- The structure of a patent application (description, claims, etc)
- Getting a patent: the patenting process (step by step)
- When to file a patent application: priority date, Patent Cooperation Treaty (PCT)
- Where to protect an invention
- Different IPR strategies

The team and the early decisions

- The creation of the founders' team
- Types and characteristics of founders' teams
- Founders' decisions and their consequences
- Frequent mistakes and suggestions deriving from experience

The economic and financial aspects of a startup

- The fundamental economic and financial operations of a technology-based startup
- The structures of the financial statements
- Income Statement, Balance Sheet, Cash Flow
- Evaluation of the value of the company
- Sources and cost of capital

[back to **Summary**]

Funding a startup

- Different sources of funds: Angel Investors and Venture Capital
- Investment companies and funds: how they work
- How and what investors evaluate
- The investment agreements between investors and startups
- New ventures' funding options

References:

- Noam Wasserman (2013) The Founder's Dilemmas: Anticipating and Avoiding the Pitfalls That Can Sink a Startup, Princeton University Press.
- Thomas R. Ittelson (2009), Financial Statements: A Step-by-Step Guide to Understanding and Creating Financial Reports, Career Press.
- Hall, J., & Hofer, C. W. (1993). Venture capitalists' decision criteria in new venture evaluation. Journal of Business Venturing, 8(1), 25-42.

Schedule and room: please, see Class Schedule

Enrollment:

To attend the course registration is compulsory by using the Moodle platform of the PhD Course in Industrial Engineering (in order to enter the Moodle platform click on "dettagli" of the course at the page http://www.cdii.dii.unipd.it/corsi). Once you are registered, if you cannot attend the course, please inform the lecturer.

Examination and grading: Attendance is required for at least 70% of the lecture hours (i.e. 14 hours). Final evaluation will be based on the discussion of a case study of a technology-based startup.

TSK 2. Python Programming for Scientific Engineering

Course Area: Transversal Skills

Credits: 5 (12+10 lecture hours; 12+10 hours for hands-on-sessions)

Instructors: Prof. Monica Reggiani, DTG, Univeristy of Padova, Dr. Luca Tagliapietra, DEI, University of Padova

e-mail: monica.reggiani@unipd.it, luca.tagliapietra@unipd.it

Topics – Part 1 "Python Programming for Scientific Engineering" (Dr. L. Tagliapietra):

Tired of banging your head against Excel and its mouse-based approach? Wondering about how to escape from such a nightmare? This course is for you: even basic Python skills can help you process and analyze your tons of data much more quickly and effectively.

This introduction to Python for Scientific Engineering will kickstart your learning of Python, as well as programming in general. The course is designed to be beginner-friendly, starting from the Python programming basics and focusing on how it can be used for scientific data manipulation and analysis.

Upon its completion, you will be able to write your own Python scripts and perform hands-on data analysis using NumPy and Pandas, the two most popular Python open-source libraries for data analysis.

Syllabus (tentative)

1. *Python Language Basics.*
   We will start from the very beginning, introducing the basic concepts of Python programming. On the first day of the course, we will dive into key concepts, including variables and operators, control flow, functions, and built-in data structures.
2. *Introduction to NumPy.*
   We will learn about NumPy, a fundamental Python package for mathematical and statistical operations. We will also get started with data exploration.
3. *Data Manipulation with Pandas*
   We will then jump into Pandas, a library created to facilitate working with data. We will use Pandas and NumPy to supercharge your analysis.
4. *Data Visualization with Matplotlib*
   We will not have a lot of time for this topic, but we must understand how to use data visualization to explore data, the core skill in data analysis.
5. *Guided project*
   In the latest hours of the course, we will apply the acquired skills to explore real data.

[back to **Summary**]

Topics –Part 2 "Getting Started with Scikit-Learn for Scientific Data Analysis" (Prof. M. Reggiani)

This course takes Python programming fundamentals for scientific engineering and moves towards their serious application for data analysis. Students will learn how to perform the most common supervised learning tasks of regression and classification. They will build predictive models, tune their parameters, and determine how well they perform with unseen data.

The course is intensely hands-on, thus focused on techniques and methods more than on these methods' statistics fundaments. Therefore, rather than implementing toy versions of the algorithms, the course use scikit-learn, one of the most popular and user-friendly machine learning libraries for Python. Attendance of the course "Introduction to Python for scientific engineering" is not a requirement, but it is strongly suggested since its contents (Python fundamentals, Pandas, Numpy, etc.) will be given-as-known in this course.

Syllabus (tentative)

1. *Introduction to data analysis*
   We will rapidly hover on fundamental concepts (and jargon) that every data scientist should know by heart. It will be a high-level overview, all relatively simple, but we should make sure everything is crystal clear before getting serious.
2. *End-to-end project*
   We will immediately try to solve a problem from the beginning to the end. We will understand that solving a problem is not just using a machine learning algorithm; we will learn how to handle, clean, and prepare data and how pipelines will tie together concepts to launch, monitor, and maintain your system.
3. *Classification*
   We will turn our attention to classification tasks and learn how to solve them using supervised learning techniques.
4. *Training models*
   While much can get done without knowing anything about what is under the hood, it is time to better understand how things work. We will learn about fundamental regression models.
5. *Decision Trees and Random Forests*
   Decision Trees and Random Forests are among the most powerful Machine Learning algorithms available today. We will solve a problem using the scikit-learn implementation of Decision Trees and discuss their limitations. We will then ensemble a group of Decision Tree to create a Random Forest that, despite its simplicity, is one of the most powerful algorithms available today.

Room: online lectures.

Timetable

Part 1 will be held on the days of 14, 15, and 16 December 2020 with the following timetable:

- 8:30-10:30 lecture
- 10:30-12:30 hands-on
- 14:00-16:00 lecture
- 16:00-18:00 hands-on

Part 2 will be held from February 1st through 5th, 2021 with the following timetable:

- 14:00-16:00 lectures
- 16:00-18:00 hands-on session

See also Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Examination and grading: To complete the course and get your grade, you will need to complete all the exercises proposed in the hands-on sessions. Although you are warmly invited to work on the exercises during the three days of the course, you can also hand them over in the following weeks but before the end of January (Part 1) and before the end of February (Part 2).

IE 1. Statistical Methods

Course Area: Information Engineering (may also be taken as Transversal Skills course)

Credits: 6 (24 hours)

Instructor: Dr. Lorenzo Finesso, CNR IEIIT Padova.

e-mail: lorenzo.finesso@unipd.it

Aim: The course will present a small selection of statistical techniques which are widespread in applications. The unifying power of the information theoretic point of view will be stressed.

Topics:

- *Background material.* The noiseless source coding theorem will be quickly reviewed in order to introduce the notions of entropy and informational divergence (relative entropy or Kullback-Leibler distance) between positive measures.
- *Divergence minimization problems.* Three I-divergence minimization problems will be posed and, via examples, connected with basic methods of statistical inference: ML (maximum likelihood), ME (maximum entropy), and EM (expectation-maximization).
- *Multivariate analysis methods.* The three standard multivariate methods, PCA (Principal Component Analysis), Factor Analysis, and CCA (Canonical Correlations Analysis) will be reviewed and their connection with divergence minimization discussed. Applications of PCA to least squares (PCR principal component regression, PLS Partial least squares). Approximate matrix factorization and PCA, with a brief detour on the approximate Nonnegative Matrix Factorization (NMF) problem.
- *EM methods.* The Expectation-Maximization method will be introduced in the context of Maximum Likelihood (ML) estimation with partial observations (incomplete data) and interpreted as an alternating divergence minimization algorithm à la Csiszár Tusnády.
- *Applications to stochastic processes.* Introduction to HMM (Hidden Markov Models). Maximum likelihood estimation for HMM via the EM method. If time allows: derivation of the Burg spectral estimation method as solution of a Maximum Entropy problem.

References: Lecture notes and a list of references will be posted on the course moodle site.

Schedule and room: Please, see Class Schedule.

Enrollment: Students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Familiarity with basic linear algebra and probability.

Examination and grading: Homework assignments.

IE 2. Statistics for Engineers

Course Area: Information Engineering (may also be taken as Transversal Skills course)

Credits: 5

Number of lecture: 6 lectures, 6 hours per day – only online, Zoom required for connection

Instructors: Prof. Luigi Salmaso, University of Padova, Prof. Rosa Arboretti, University of Padova, Dr. Marta Disegna, Bournemouth University.

e-mail: luigi.salmaso@unipd.it

Important note: registration mandatory; if you are interested to take this class (please see website https://phd.dei.unipd.it/courses/)

Aim: The course is an introduction to statistical methods most frequently used for experimentation in Engineering. Lectures are planned both in the classroom and in computer lab also for an introduction to the use of the following statistical software:

- R
- MINITAB (licensed to University of Padova)

Topics:

1. Elements of univariate statistical methods:
    a. Elements of descriptive statistics: frequency, indices of synthesis (position, variability and shape) and graphical representations (histogram, boxplot, scatterplot).
    b. Elements of probability theory: discrete and continuous probability distributions.
    c. Elements of statistical inference: sampling distributions, point and interval estimation, hypothesis testing, One-way ANOVA.

2. Linear and non-linear regression models:
    a.Simple and multiple linear regression model
    b.Logit model

3. Multivariate data analysis:
    a. Cluster Analysis: idea and steps
    b. Multidimensional data, matrix representation and data preparation.
    c. Distance and dissimilarity matrices.
    d. Hard clustering algorithms: hierarchical clustering algorithms, non-hierarchical clustering algorithms and Bagged clustering algorithm.
    e. Fuzzy clustering algorithms: fuzzy *C*-means and fuzzy *C*-medoids.
    f. Validity indices and optimal number of clusters.

[back to **Summary**]

g. Labelling and profiling the clusters: an application of suitable tests and regression models.

4. DOE: Introduction to Factorial Designs, Two level and general factorial designs. Tutorials in MINITAB.

Bibliography

1. Stark, P.B., 1997. SticiGui: Statistics Tools for Internet and Classroom Instruction with a Graphical User Interface.
2. Montgomery DC, Design and Analysis of Experiments, 2010, Wiley.
3. Lattin J, Carroll JD, Green PE, Analyzing Multivariate Data, 2003, Duxbury Applied Series.
4. Johnson RA, Wichern DW, Applied Multivariate Statistical Analysis,1998, Prentice Hall; 4th edition.
5. Hollander and Wolfe, Nonparametric Statistical Methods, 2nd edition, 1999, Wiley Series in Probability and Statistics.
6. Shumway RH, Stoffer DS, Time Series Analysis and Its Applications (With R Examples), 2nd Edition, 1998, Springer Texts in Statistics, NewYork.
7. Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). Cluster analysis (Fifth ed.). Wiley series in probability and statistics: John Wiley & Sons, Ltd.
8. Adhoc material by Lecturer.

Useful Web-based Resources:

Stat tutor: http://www.statstutor.ac.uk/

HyperStat Online Stat Textbook: http://davidmlane.com/hyperstat/

Stats tools: http://www.imathas.com/stattools/

Datacamp: https://www.datacamp.com/

Regarding R

Stat with R: https://www.r-tutor.com/elementary-statistics
R Installation and Administration:
http://www.statistik.tuwien.ac.at/lv-guide/software/MANUALS/R-admin.pdf
Paradis, E. (2005), R for Beginners:
https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf
Online R resources for Beginners:

[back to **Summary**]

http://www.introductoryr.co.uk/R_Resources_for_Beginners.html

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Examination and grading: Attendance is required for at least 2/3 of the lecture hours. Final evaluation will be based on the discussion of a case study within the individual PhD project.

OUTLINE OF LECTURE AND LAB PROGRAMME

Date	Lecture (morning)	Tutorial/Lab (afternoon)
2/02/21	Elements of univariate statistical methods	Introduction to MINITAB, MINITAB for univariate statistical methods
4/02/21	Introduction to R, R for univariate statistical methods	R for univariate statistical methods, linear and non-linear regression models
9/02/21	Multivariate data analysis	R for Multivariate data analysis
11/02/21	Multivariate data analysis	R for Multivariate data analysis
23/02/21	DOE	MINITAB for DOE
25/02/21	DOE	MINITAB for DOE

IE 3. Computational Inverse Problems

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Fabio Marcuzzi, Dept. of Mathematics, University of Padova.

e-mail: marcuzzi@math.unipd.it

Aim: We study numerical methods that are of fundamental importance in computational inverse problems. Real application examples will be given for distributed parameter systems in continuum mechanics. Computer implementation performance issues will be considered as well.

Topics:

- definition of inverse problems, basic examples and numerical difficulties.
- numerical methods for QR and SVD and their application to the square-root implementation in PCA, least-squares, model reduction and Kalman filtering; recursive least-squares; High Performance Computing (HPC) implementation of numerical linear algebra algorithms.
- regularization methods;
- underdetermined linear estimation problems and sparse recovery;
- numerical algorithms for nonlinear parameter estimation: nonlinear least-squares (Levenberg-Marquardt), back-propagation learning;
- underdetermined nonlinear estimation problems and deep learning;
- examples with distributed parameter systems in continuum mechanics: reconstruction of forcing terms and parameters estimation;

References:

[1]F.Marcuzzi "Computational Inverse Problems", lecture notes (will be posted on the moodle page of the course)

[2]G. Strang, "Linear Algebra and Learning From Data", Wellesley - Cambridge Press, 2019

[3]L. Trefethen and J. Bau, “Numerical Linear Algebra”, SIAM, 1997

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements:

[back to **Summary**]

- basic notions of linear algebra and, possibly, numerical linear algebra.
- the examples and homework will be in Python (the transition from Matlab to Python is effortless).

Examination and grading: Homework assignments and final test.

IE 4. Heuristics for Mathematical Optimization

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Domenico Salvagnin

e-mail: dominiqs@gmail.com, domenico.salvagnin@unipd.it

Aim: Make the students familiar with the most common mathematical heuristic approaches to solve mathematical/combinatorial optimization problems. This includes general strategies like local search, genetic algorithms and heuristics based on mathematical models.

Topics:

- Mathematical optimization problems (intro).
- Heuristics vs exact methods for optimization (intro).
- General principle of heuristic design (diversification, intensification, randomization).
- Local search-based approaches.
- Genetic/population based approaches.
- The subMIP paradigm.
- Applications to selected combinatorial optimization problems: TSP, QAP, facility location, scheduling.

References:

[1] Gendreau, Potvin "Handbook of Metaheuristics", 2010

[2] Marti, Pardalos, Resende "Handbook of Heuristics", 2018

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements:

- Moderate programming skills (on a language of choice)
- Basics in linear/integer programming.

Examination and grading: Final programming project.

IE 5. Statistical Learning for Big Data in Medicine

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Andrea Facchinetti (Department of Information Engineering, University of Padova), Dr. Martina Vettoretti (Department of Information Engineering, University of Padova)

e-mail: facchine@dei.unipd.it

Aim: The course is intended to provide a better understanding of the methodologies used in the analysis of big data in medical applications and epidemiology.

Topics:

- Types of clinical studies (randomized clinical trials, retrospective studies, longitudinal studies), definition of exposures and main outcomes (incidence, prevalence, risk ratio, odds ratio);
- Logistic regression to link covariates to the main outcome: definition and properties, parameter estimation via maximum likelihood, coefficient interpretation, goodness of fit tests, covariate selection;
- Survival analysis: definition of lifetime, survival, and hazard functions; univariate nonparametric and parametric survival analysis; multivariate survival analysis with parametric and semiparametric (Cox) proportional hazard models;
- Practical issues related to predictive model development: dealing with missing values, dealing with unbalanced datasets, ensure model generalization;
- Hands-on: analysis of big data collected in medical research and implementation of logistic / survival models for the prediction of clinical outcomes

References:

- A set of lecture notes and a complete list of references will be made available by the Lecturer
- The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Second Edition, 2009) by Trevor Hastie, Robert Tibshirani, Jerome Friedman

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Basics of probability theory and statistics; basics of Matlab programming.

Examination and grading: Final project consisting in the development of logistic / survival models on a given dataset.

IE 6. Model Predictive Control with Case Studies in Automotive and Biomedicine

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructors: Prof. Ruggero Carli, Dr. Mattia Bruschetta, Dr. Simone Del Favero, Department of Information Engineering, University of Padova

e-mail : carlirug@dei.unipd.it, mattia.bruschetta@dei.unipd.it, simone.delfavero@unipd.it

Aim: To provide the methodological tools needed to understand model based control and basic knowledge of Model Predictive Control. The course is tailored to students who have not received an extensive training on control theory. As case studies, the course focus on Automotive and Bioengineering applications.

Topics:

1. Introduction to model based control.
2. State Space Models: driving the state with inputs
3. State Space Model: estimating the state form the output;
4. Learning Linear models form the Data
5. Linear Quadratic Problem, the Infinite Horizon LQ Problem, Convergence of the Linear Quadratic Regulator.
6. Model Predictive Control - Regulation: Formulation, Dynamic Programming Solution, Stability properties, MPC for Unconstrained Systems, MPC for Systems with Control Constraints, MPC for Systems with Control and State Constraints, Suboptimal MPC, Tracking.
7. Elements of Nonlinear MPC: Direct and Indirect methods for NLP, Condensing, Sequential Quadratic Programming, Real Time Iteration Scheme.
8. Automotive case studies: Motion Cueing Algorithms, Virtual Rider, Autonomous Driver.
9. Biomedicine case study: the Artificial Pancreas. The Blood Glucose Regulation Problem, possible MPC Approaches (modular MPC, zone MPC, non-linear MPC), Clinical Testing.

References:

[1] J. B. Rawlings and D. Q. Mayne. Model predictive control : Theory and design. Nob Hill Publisher.

Other material and research papers will be available online for download.

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Basic Calculus and Linear Algebra.

[back to **Summary**]

Examination and grading: Homework and take home exam.

IE 7. Fluid mechanics for the functional assessment of cardiovascular devices

Course Area: Information Engineering

Credits: 5 (18 hours)

Instructor: Prof. Francesca Maria Susin, Dept. of Civil, Environmental and Architectural Engineering (DICEA)

e-mail: francescamaria.susin@unipd.it

Aim:

The course is intended to give a survey of research approaches for the assessment of cardiovascular medical devices. Emphasis will be given to methods and techniques adopted for in vitro analysis of hemodynamic performance of prosthetic heart valves.

Topics:

Review of basic fluid mechanics concepts. Fluid mechanics of prosthetic heart valves (PHVs). Pulse duplicators for in vitro testing of PHVs and mock circulation loops for pre-clinical evaluation of VADs. Experimental techniques for the assessment of PHVs. CFD for functional assessment of PHVs.

References:

[1] M. Grigioni, C. Daniele, G. D'Avenio, U. Morbiducci, C. Del Gaudio, M. Abbate and D. Di Meo. Innovative technologies for the assessment of cardiovascular medical devices: state of the art techniques for artificial heart valve testing. Expert Rev. Medical Devices, 1(1) : 81-93, 2004.

[2] K.B. Chandran, A.P. Yoganathan and S.E. Rittgers. Bio fluid Mechanics: the human circulation. CRC Press, Boca Raton, FL, 2007.

[3] A.P. Yoganathan, K.B. Chandran and F. Sotiropoulos. Flow in prosthetic heart valves: state of the heart and future directions. Annals of Biomedical Engineering, 33(12) : 1689-1694, 2005.

[4] A.P. Yoganathan, Z. He and S. Casey Jones. Fluid mechanics of heart valves.

[5] A.P. Yoganathan and F. Sotiropoulos. Using computational fluid dynamics to examine the hemodynamics of artificial heart valves. Business briefing: US cardiology 2004 : 1-5, 2004.

[6] V. Barbaro, C. Daniele and M. Grigioni. Descrizione di un sistema a flusso pulsatile per la valutazione delle protesi valvolari cardiache. ISTI-SAN Report 91/7, Rome, Italy, 1991 (in Italian).

[7] M. Grigioni, C. Daniele, C. Romanelli and V. Barbaro. Banco di prova per la caratterizzazione di dispositivi di assistenza meccanica al circolo. ISTISAN Report 03/21, Rome, Italy, 2003 (in Italian).

[8] M.J. Slepian, Y. Alemu, J.S. Soares. R.G. Smith, S. Einav and D. Bluestein. The Syncardia total arti cial heart: in vivo, in vitro, and computational modeling studies. Journal of Biomechanics,46 (2013): 266-27, 2013.

[back to **Summary**]

Further references will be given during the course.

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Fundamentals of Fluid Dynamics.

Examination and grading: Homework assignment with final discussion.

IE 8. Diagnostics of Electron Devices

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Giovanna Mura, Department of Electrical and Electronic Engineering (DIEE), University of Cagliari.

e-mail: gmura@diee.unica.it

Aim: this course provides an overview of the Failure Analysis techniques for the diagnostics of electron devices. Failure analysis is the process of analyzing the failed electron devices to determine the reason for degraded performance or catastrophic failure and to provide corrective actions able to x the problem. It is a proactive tool with three fundamental tasks: 1) Technical/scientific: 2) Technological 3) Economical. The purpose of this course is to teach what Failure Analysis should be and should do, to show how and why it often does not, to state that F.A. has Logics and has Rules.

Microscopy, in its several forms (optical, electron, scanning, transmission, emission, ionic) and tools is the playground for practical FA, and its fundamentals will be described. Device basic technology, working principle and failure physics are the other pillars for a successful study.

Several case studies will be proposed with the aim to demonstrate that if sometimes Failure Analysis looks unclear or not problem solving is merely because it was badly conducted.

Topics:

1. Reverse engineering
2. Failure modes and failure mechanisms
3. Principles and fundamental methods in Electron Microscopy
4. Methodology for the Failure Analysis

References: Failure Analysis of Integrated Circuits - Tools and Techniques, Springer International Series - Lawrence C. Wagner.

Slides

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Electron Devices, Microelectronics, Optoelectronics devices.

Examination and grading: Written test/ presentation of a report at the end of the course.

[back to **Summary**]

IE 9. Physics and Operation of Heterostructure-Based Electronic and Optoelectronic Devices

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructors: Prof Enrico Zanoni, Prof. Matteo Meneghini, Dr. Carlo De Santi, DEI, University of Padova.

e-mail: zanoni@dei.unipd.it, menego@dei.unipd.it, desantic@dei.unipd.it

Aim: This course provides an introduction to the physics and operating principles of advanced electronic and optoelectronic devices based on compound semiconductors. These devices are particularly important for several applications: high electron mobility transistors (HEMTs) represent excellent devices for the realization of high frequency communication systems, radars, satellite applications, and high efficiency power converters. On the other hand, LEDs and lasers are high-efficiency monochromatic light sources, that can be used both for lighting applications (with a considerable energy saving), in the biomedical field, and in in photochemistry. Special focus will be given to Gallium Nitride (GaN) based devices, that represent the most promising devices for future power electronics applications. This course will focus on the main aspects related to the physics of heterostructures, on the recombination processes in semiconductors, on carrier transport in heterostructures, on the structure and operating principles of MESFET, HEMTs, GITs, on the trapping and reliability in compound semiconductor devices, on the operating principles of LEDs and lasers, and on parasitics and reliability in LEDs and lasers. An overview of real applications highlighting the capabilities of these devices will also be given.

Topics:

- physics of heterostructures, band diagrams, carrier transport in heterostructures;
- recombination processes in semiconductors; properties of compound semiconductors;
- basic structure of heterojunction transistors, MESFET, HEMT, GIT; parasitics and reliability in HEMTs, LEDs and lasers;
- operating principles of LEDs and lasers;
- methods for advanced characterization of heterojunction based devices; applications of GaN based HEMTs, LEDs and lasers;
- modeling of semiconductor-based devices

References:

Umesh Mishra, Jasprit Singh, Semiconductor Device Physics and Design, Springer, 2008

Ruediguer Quay, Gallium Nitride Electronics, Springer 2008.

Tae-Yeon Seong, Jung Han, Hiroshi Amano, Hadis Morko, III-Nitride Based Light Emitting Diodes and Applications, Springer 2013

[back to **Summary**]

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Introductory course of device physics: Microelectronics, Optoelectronic and Photovoltaic Devices.

Examination and grading: Written test at the end of the course.

IE 10. Embedded Design with FPGA

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructors: Dr. Andrea Stanco and Prof. Daniele Vogrig, Department of Information Engineering, University of Padova

e-mail: andrea.stanco@dei.unipd.it, vogrig@dei.unipd.it

Aim: The course aims at teaching how to practically use System-on-a-Chip (FPGA+CPU) as a potential application to academic research topics. 75% of the course will be held in a dedicated laboratory to deal with the programming of a Pynq-Z1 board.

Topics:

- Recap on basic of Digital Design. Digital Design Flow (HDL language and HLS). Introduction to VHDL program language.
- Introduction to FPGA and Zynq SoC.
- Introduction to Vivado System Design environment. Time domains, time violations, metastability, system constraints.
- Introduction to SDK environment
- Information exchange between processor and programmable logic. Hardware and Software interrupts.
- Communication between SoC and the outside world.
- PYNQ (Python on Zinq) project as example of how to make easier the design embedded systems
- Case studies

References:

[1] Xilinx, Vivado Design Suite User Guide, UG893 (v2019.1), https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug893-vivado-ide.pdf

[2] Xilinx, Xilinx Software Development Kit (SDK) User Guide, https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_1/SDK_Doc/index.html

Other material will be pointed out in class and available online for download

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

[back to **Summary**]

Course requirements: Basic knowledge of digital electronics. Knowledge of program language (e.g. C/C++). No VHDL knowledge or experience on FPGAs is required.

Examination and grading: Homework assignments and final project.

IE 11. Introduction to Information Theory

Course Area: Information Engineering

Credits: 4

Instructor: Prof. Deniz Gunduz

e-mail: d.gunduz@imperial.ac.uk

Important note: Not offered in a.a. 2020/21

IE 12. Machine Learning for Wireless Communication Systems

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Dr. Paolo Dini, Senior Researcher, Centre Tecnologic de Telecomunicacions de Catalunya

e-mail: paolo.dini@cttc.es

Aim: The course will introduce the main architectures used in the design of next-generation mobile systems, together with their challenges and open issues. In particular, we focus on data-centric network scenarios and network control systems to provide broadband communications and support vertical markets. The core of the course is the application of Machine Learning (ML) tools to solve the identified networking problems. It will explained what the usage models are and what they imply in terms of stability, convergence and optimality guarantees. For this, fundamentals of Reinforcement Learning and Artificial Neural Networks will be given. Finally, several ML algorithms will be tailored for specific case studies, such as the Energy - Quality of Service trade-off and the analysis of context information (traffic demand, user mobility). The course covers Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning applications to mobile networking.

Topics:

- Introduction of next-generation mobile network scenarios and architectures
    - network softwarization paradigm
    - data-centric network scenario
    - vertical markets
- Identification of machine learning tools for mobile networking issues
- Fundamentals of Reinforcement Learning
    - approximated Dynamic Programming
    - Temporal-Difference methods
    - Deep-Reinforcement Learning
- Fundamentals of Artificial Neural Network architectures
    - Multi-layer perceptrons
    - Recurrent neural networks
    - Convolutional neural networks
    - Auto-encoders
- Mobile network on-line optimization methods
    - Applications of Reinforcement Learning
- Mobile traffic characterization and modeling
    - Applications of Artificial Neural Networks

References:

[back to **Summary**]

[1] Anwer Al-Dulaimi, Xianbin Wang, Chih-Lin I, 5G Networks: Fundamental Requirements, Enabling Technologies, and Operations Management, October 2018, Wiley-IEEE Press

[2] R. Boutaba, M.A. Salahuddin, N. Limam, et al., A comprehensive survey on machine learning for networking: evolution, applications and research opportunities, Elsevier J. Internet Serv Appl (2018) 9: 16

[3] Richard Sutton, Andrew G. Barto, Reinforcement Learning: An Introduction, MIT press, 2017

[4] Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT press

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Basic knowledge of probability theory, random processes, python scripting.

Examination and grading: Each student will develop a final project, possibly related to his/her research activity, addressing some topic presented in the Course.

[back to **Summary**]

IE 13. Introduction to Reinforcement Learning

Course Area: Information Engineering

Credits: 4 (15 hours)

Instructor: Dr. Juan José Alcaraz Espín, Associate Professor, Technical University of Cartagena, Spain.

e-mail: juan.alcaraz@upct.es

Aim: The course will provide an introduction to the field of reinforcement learning, covering its mathematical foundations and the description of the most relevant algorithms. The main concepts and techniques will be illustrated with Python code and application examples in telecommunications and other related areas. The students will acquire hands-on experience with the proposed assignments in which they will have to implement Python code for solving several challenges and exercises. The course will start with the basic concepts of learning in sequential decision problems, formalized in the multi-armed bandit (MAB) problem and its variants. Then, the Markov decision processes (MDPs), which generalize the MAB problem, will be introduced. The objective of reinforcement learning (RL) is to find approximate solutions to MDPs. The main RL approaches will be presented incrementally: 1) tabular methods, which are capable of addressing relatively small problems, 2) value function approximation, which allows scaling up previous algorithms to larger problems, and 3) policy gradient algorithms which follow a different scaling approach and can be used in combination with value function approximation (Actor-Critic methods).

Topics:

Unit 1. Introduction to Reinforcement Learning

Unit 2. Multi-Armed Bandits: Stochastic Bandits, Boltzmann Exploration, UCB algorithms, Thompsom Sampling, Contextual Bandits.

Unit 3. Markov Decision Processes: Stochastic Shortest Path problems. Policy Iteration. Value Iteration. MDPs with discount.

Unit 4. Tabular Methods: Monte Carlo Method, Temporal Difference, Off-policy algorithms, Planning at decision time.

Unit 5. Value Function Approximation (VFA) Methods: Linear VFA, Monte Carlo with VFA, TD methods with VFA.

Unit 6. Policy Gradient Algorithms: Score functions, Policy Gradient Theorem, Monte Carlo Policy Gradient, Actor-Critic Policy Gradient.

Unit 7 (Optional) Evolutionary Algorithms

References:

[1] Reinforcement Learning: An Introduction, Second Edition, Richard S. Sutton and Andrew G. Barto, MIT Press, Cambridge, MA, 2018.

[back to **Summary**]

[2] Approximate Dynamic Programming: Solving the Curses of Dimensionality, Second Edition, Warren B. Powell, Wiley, 2011.

[3] Dynamic Programming and Optimal Control Vol I and Vol II, 4th Edition, Dimitri P. Bertsekas, Athena Scientific, 2012.

[4] Algorithms for Reinforcement Learning, Csaba Szepesvári, Morgan and Claypool, 2010.

[5] Reinforcement Learning and Optimal Control, Dimitri P. Bertsekas, Athenea Scientific, 2019.

[6] Markov Decision Processes: Discrete Stochastic Dynamic Programming, Martin L. Puterman, Wiley, 2006.

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course Requirements: Basics of linear algebra, probability theory, Python scripting

Examination and Grading: The grading will be based on the students' solutions to the proposed assignments.

IE 14. Information Theoretic Models in Security

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Nicola Laurenti, Department of Information Engineering

e-mail: nil@dei.unipd.it

Aim: The class aims at providing the students with an information theoretic framework that will allow formal modeling, understanding of the fundamental performance limits, and derivation of unconditionally secure mechanisms for several security-related problems.

Topics:

Lectures 1-5 will explore the following fixed topics:

- *All or nothing: security without compromise.* Quantitative definition of security. Unconditional security. Distinguishability. Composable security
- *The Holy Grail of perfect secrecy.* Shannon's cipher system. Perfect secrecy. Ideal secrecy. Practical secrecy. The guessing attack.
- *Secrecy without cryptography.* The wiretap channel model. Rate-equivocation pairs. Secrecy rates. Secrecy capacity for binary, Gaussian and fading channel models.
- *Security from uncertainty.* Secret key agreement from common randomness on noisy channels. Information theoretic models and performance limits of quantum key distribution.

Lectures 6-10 will introduce a few topics chosen by the students and the instructor among the following:

- *Who's who?* An information theoretic model for authentication in noisy channels. Signatures and fingerprinting.
- *The gossip game.* Broadcast and secrecy models in multiple access channels. The role of trusted and untrusted relays.
- *Secrets in a crowd.* Information theoretic secrecy in a random network with random eavesdroppers. Secrecy graphs and large networks secrecy rates.
- *A cipher for free?* Information theoretic security of random network coding.
- *The jamming game.* Optimal strategies for transmitters, receivers and jammers in Gaussian, fading and MIMO channels.
- *Alea iacta est.* Secure and true random number generation. Randomness extractors and smooth guessing entropy
- *Writing in sympathetic ink*. Information theoretic models of steganography, watermarking and other information hiding techniques.
- *Leaky buckets and pipes.* Information leaking and covert channels. Timing channels.
- *The Big Brother.* Privacy and anonymity measures. Differential privacy. The privacy vs utility database tradeoff.

[back to **Summary**]

- *The dining cryptographers.* Unconditional secret sharing and secure multiparty computation.
- *Information theoretic democracy.* Privacy, reliability and verifiability in electronic voting systems.

References:

- Y. Liang, H.V. Poor, and S. Shamai (Shitz), *Information Theoretic Security*, Now, 2007.
- M. Bloch, J. Barros, *Physical-Layer Security: from Information Theory to Security Engineering*, Cambridge University Press, 2011.

A short list of reference papers for each lecture will be provided during class meetings.

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Basic notions of Information Theory (e.g., those from the *Telecomunicazioni* class in the *Corso di Laurea in Ingegneria dell'Informazione*).

Examination and grading: Each student (or small group of students) must submit a project, and grading will be based on its evaluation. Students are encouraged to work from an information theoretic point of view on a security problem related to their research activities.

IE 15. Modeling and Simulation of Complex & Multi-Disciplinary Dynamical Systems

Course Area: Information Engineering

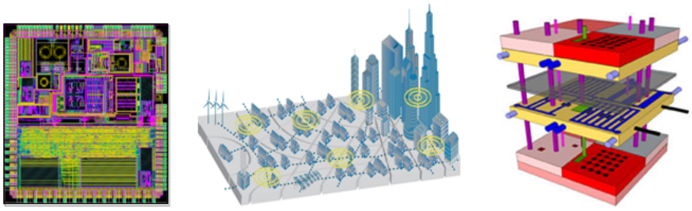
Credits: 4 (16 hours)

Instructor: Luca Daniel, Professor of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

e-mail: luca@mit.edu

URL: projects abstracts and video presentations with demos by students that took this class at MIT

Aim: Many complex systems found in nature/society and studied by social/economical/physical scientists (e.g. the human cardiovascular system, the brain neural network, biological systems, the geophysical network of oil/water/gas reservoirs, social networks), or developed by engineers (e.g. labs on chips, iPads, magnetic resonance scanners, nationwide electrical/gas/oil transportation network, supply chains, currency/stock markets, buildings/automotive/aircraft frames) can be viewed as large collections of interconnected dynamical system components. The performance of such complex systems typically depends on complicated constitutive and conservation relations between components, as well as on random uncertainties.



Power delivery network for Integrated Circuit (IC) or city/state   Heat sink for 3D IC

Blood/Oil delivery network   Building/Mechanical frames

The **goal** of this course is to provide students with a working hands-on knowledge of the state of the art in modeling, simulation, model order reduction and uncertainty quantification techniques. Examples will be drawn from a large variety of complex and multi-disciplinary dynamical systems, as well as from student proposed applications, helping them with their own research projects in different engineering and science disciplines that deal with complex systems.

Topics:

Upon completion of this course students will be able to:

- Recognize and formulate mathematical frameworks (e.g. conservation laws and constitutive equations) common to a lot of multi-disciplinary complex dynamical systems.
- Select, modify and implement an appropriate steady state solver (e.g. sparse LU vs. iterative methods) for their linear or linearized complex system description.

[back to **Summary**]

- Select, implement and modify an appropriate strategy to facilitate initialization and convergence of a Newton solver for their nonlinear complex system.
- Select and implement an appropriate technique (e.g. explicit fix time step vs. implicit adaptive time steps) for the time domain simulation of their complex system.
- Select and implement an appropriate strategy (e.g. Shooting Newton or Harmonic Balance) for period state analysis of their complex system (e.g. vibrations in mechanical/structural frames, radio frequency circuits, heart beat cycles).
- Select and implement an appropriate strategy to "reduce" automatically models of system components generated for instance by large PDE solvers' discretizations, while preserving input/output accuracy for a range of parameter values, as well as important physical properties. Select and implement an appropriate strategy to "generate" automatically stable parameterized reduced models of system components from input/output measurements.
- Use parameterized reduced order models of system components in order to accelerate optimization, inverse problems in complex systems.
- Select and implement uncertainty quantification techniques for stochastic simulation of complex systems affected by random variations in geometries and material properties.

Schedule and room: late spring 2021. Exact lecture days and time will be reported in Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Calculus, differential equations, linear algebra as well as some basic programming experience in Matlab, or other programming languages for scientific computing.

Examination and grading: Students will be working in small teams on a course-long project involving modeling and simulation of a complex system either self-proposed from their own field of research, or chosen from a few examples developed in class. Time in class will include short lectures interleaved by numerous interactive and hands-on activities coordinated by the instructor and supporting the self-proposed projects. Final evaluations will be based on in-class work and interaction with the staff during the course as well as on a final live project demo presentation and report. The focus of the course will not be on mathematical formalism and rigorous theorem proving, but rather on developing general intuition, creativity, practical implementation and model debugging skills.

[back to **Summary**]

IE 16. Elements of Deep Learning

Course Area: Information Engineering

Credits: 6 (24 hours)

Instructor: Dr. Gian Antonio Susto

e-mail: gianantonio.susto@dei.unipd.it,

Aim: The course will serve as an introduction to Deep Learning (DL) for students who already have a basic knowledge of Machine Learning. The course will move from the fundamental architectures (e.g. CNN and RNN) to hot topics in Deep Learning research.

Topics:

- Introduction to Deep Learning: context, historical perspective, differences with respect to classic Machine Learning.
- Feedforward Neural Networks (stochastic gradient descent and optimization).
- Convolutional Neural Networks.
- Neural Networks for Sequence Learning.
- Elements of Deep Natural Language Processing.
- Elements of Deep Reinforcement Learning.
- Unsupervised Learning: Generative Adversarial Neural Networks and Autoencoders.
- Laboratory sessions in Colab.
- Hot topics in current research.

References:

[1] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. CoRR, abs/1701.07875.

[2] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. CoRR, abs/1409.0473.

[3] I. Goodfellow, Y. Bengio, A. Courville 'Deep Learning', MIT Press, 2016

[4] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., & Bengio, Y. (2014). Generative Adversarial Nets. NIPS.

[5] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural computation, 9 8, 1735-80.

[6] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. ACL.

[7] Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Commun. ACM, 60, 84-90.

[8] LeCun, Y. (1998). Gradient-based Learning Applied to Document Recognition.

[back to **Summary**]

[9] Mikolov, T., Sutskever, I., & Chen, K. (2013). Representations of Words and Phrases and their Compositionality.

[10] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. Journal of Machine Learning Research, 11, 3371-3408.

[11] Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent Neural Network Regularization. CoRR, abs/1409.2329.

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Basics of Machine Learning and Python Programming.

Examination and grading: Final project.

IE 17. Applied Functional Analysis and Machine Learning

Course Area: Information Engineering

Credits: 7 (28 hours)

Instructor: prof. Gianluigi Pillonetto

e-mail: giapi@dei.unipd.it

Aim: The course is intended to give a survey of the basic aspects of functional analysis, machine learning, regularization theory and inverse problems.

Topics: Review of some notions on metric spaces and Lebesgue integration: Metric spaces. Open sets, closed sets, neighborhoods. Convergence, Cauchy sequences, completeness. Completion of metric spaces. Review of the Lebesgue integration theory. Lebesgue spaces.

Banach and Hilbert spaces: Finite dimensional normed spaces and subspaces. Compactness and finite dimension. Bounded linear operators. Linear functionals. The finite dimensional case. Normed spaces of operators and the dual space. Weak topologies. Inner product spaces and Hilbert spaces. Orthogonal complements and direct sums. Orthonormal sets and sequences. Representation of functionals on Hilbert spaces.

Compact linear operators on normed spaces and their spectrum: Spectral properties of bounded linear operators. Compact linear operators on normed spaces. Spectral properties of compact linear operators. Spectral properties of bounded self-adjoint operators, positive operators, opera- tors defined by a kernel. Mercer Kernels and Mercer theorem.

Reproducing kernel Hilbert spaces, inverse problems and regularization theory: Representer theorem. Reproducing Kernel Hilbert Spaces (RKHS): definition and basic properties. Examples of RKHS. Function estimation problems in RKHS. Tikhonov regularization. Primal and dual formulation of loss functions. Regularization networks. Consistency/generalization and relationship with Vapnik's theory and the concept of V-gamma dimension. Support vector regression and classification.

References:

[1] W. Rudin. Real and Complex Analysis, McGraw Hill, 2006

[2] C.E. Rasmussen and C.K.I. Williams. Gaussian Processes for Machine Learning. The MIT Press, 2006

[3] H. Brezis, Functional analysis, Sobolev spaces and partial differential equations, Springer 2010

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

[back to **Summary**]

Course requirements: The classical theory of functions of real variable: limits and continuity, differentiation and Riemann integration, infinite series and uniform convergence. The arithmetic of complex numbers and the basic properties of the complex exponential function. Some elementary set theory. A bit of linear algebra.

Examination and grading: Homework assignments and final test.

[back to **Summary**]

IE 18. Applied Linear Algebra

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructors: Prof. Luca Schenato, Dipartimento di Ingegneria dell'informazione, Università di Padova (http://automatica.dei.unipd.it/people/schenato.html)

e-mail: schenato@dei.unipd.it

Aim: We study concepts and techniques of linear algebra that are important for applications with special emphasis on the topics: solution of systems of linear equations with particular attention to the analysis of the backward error and computational cost of the basic algorithms and matrix equation. A wide range of exercises and problems will be an essential part of the course and constitute homework required to the student.

Topics:

1. Vectors: inner products, norms, main operations (average, standard deviation, ...)
2. Matrices: matrix-vector and matrix-matrix multiplication, Frobenius norm,
3. Complexity, sparsity
4. Special matrices: Diagonal, Upper Triangular, Lower triangular, Permutation (general pair), inverse and orthogonal
5. A square and invertible: LU decomposition (aka gaussian elimination), LU-P decomposition, Cholesky decomposition
6. Ax=b via LU-P decomposition: forward and backward substitution
7. (sub)Vector spaces: definitions, span, bases (standard, orthogonal, orthonormal), dimension, direct sum, orthogonal complement, null space, orthogonal complement theorem
8. Gram-Smith orthogonalization and QR decomposition (square and invertible A, general non-square)
9. Ax=b via QR decomposition. LU-P vs QR
10. Linear maps: image space, kernel, column and row rank
11. Fundamental Theorem of Linear Algebra (Part I): rank-nullity Theorem, the 4 fundamental subspace
12. Eigenvalues/eigenvector and Shur decomposition
13. Projection matrices: oblique and orthogonal, properties
14. Positive semidefinite matrices: properties and quadratic functions square root matrix
15. Properties of A'A and AA' and Polar decomposition
16. Singular Value Decomposition: proofs and properties
17. Pseudo-inverse: definition and relation to SVD
18. Fundamental Theorem of Linear Algebra (Part II): special orthogonal basis for diagonalization
19. Least-Squares: definition, solution and algorithms

[back to **Summary**]

20. Ill-conditioned problems vs stability of algorithms, numerical conditioning of algorithms, numerical conditionings

Objectives:

- Theory: formal proofs of many results (theorem-proof type problems)
- Algorithms: understanding of most commonly used algortithm used in MATLAB and Python for Linear Algebra
- Implementation: MATLAB implementation of algorithms and performance evaluation on Big Data

References:

Textbooks and Internet Notes:

1. S. Boyd, L. Vanderberghe, "**Introduction to Applied Linear Algebra**", Cambridge University Press, 2018
2. G. Strang, "**The Fundamental Theorem of Linear Algebra**", *The American Mathematical Monthly*, vol. 100(9), pp. 848-855, 1993
3. G. Strang, "**Linear Algebra and Learning From Data",** Wellesley - Cambridge Press, 2019

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: A good working knowledge of basic notions of linear algebra as for example in [1]. Some proficiency in MATLAB.

Examination and grading: Grading is based on Homeworks, Written final exam, Short presentation based on a recent paper of Linear Algebra Algorithms for Big Data.

IE 19. Control of Multivariable Systems: A Geometric Approach

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Andrea Serrani, The Ohio State University

e-mail: serrani.1@osu.edu

Aim: The goal of the course is to introduce the geometric theory of linear multivariable systems as a fundamental tool for the solution of relevant control problems, including disturbance rejection, non-interaction, fault detection and isolation, and tracking and regulation. Attention will be devoted to routines available in MATLAB for numerical implementation of the control algorithms presented in class. Design examples on a realistic model of an aerospace system will be introduced.

Topics:

1. **Background:** Subspaces, maps, factor spaces, projections.
2. **Systems Theory:** Controllability, observability, compensator design.
3. **Disturbance Decoupling:** Controlled invariance, controllability subspaces. Duality: Conditioned invariance, unknown-input observers.
4. **Eigenvalue Assignment under Invariance Constraints:** Multivariable zeros. Zero dynamics.
5. **Non-interacting Control:** Synthesis via dynamic extension. Duality: Fault detection and isolation.
6. **Tracking and Regulation:** Right-inversion. The regulator problem.

References: W.M. Wonham, "Linear Multivariable Control: A Geometric Approach," Springer-Verlag; Supplementary notes.

Schedule and room: late spring/summer 2021, lectures exact days and time will be published in Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: A basic course in linear system theory and proficiency in linear algebra are required. Working knowledge in MATLAB/SIMULINK is needed for the design examples.

Examination and grading: Homework assignments and/or take-home final examination.

IE 20. Causal Inference for Complex Networks

Course Area: Information Engineering

Credits: 4 (16 hours)

Instructor: Prof. Reza Arghandeh, Department of Electrical and Computer Engineering, Florida State University

e-mail: r.arghandeh@fsu.edu

Aim:

One of the notable analytical challenges of our century is the intricate complexity of systems that shape our civilization ranging from electricity networks to computer networks to biological networks and social networks due to all interdependency and interconnectivity among them. It is near impossible to understand complex network systems behavior unless we go beyond the classic machine learning and network science and develop a casual insight into the machinery behind different networks. Nevertheless, the notable differences in forms, scopes, components, and nature of different networks, most networks follow common cause and effect principles. This course provides a selection of concepts from information theory and causality inference domains to analyze complex networks considering their inherent interdependencies. During the course, students will be familiar with use cases from electric grids, roadways, and social networks.

Topics:

1. Motivating problems in complex networked systems. i) some analytical problems in smart grids. ii) some analytical problems in smart cities.
2. Elements of Graph Theory: i) overview of graphs. ii) path, connectivity, and weighted graphs. iii) metrics for graphs.
3. Causality Inference: i) causality language ii) theory of causation and intervention iii) state- of-the-art causality inference methods
4. Causality for Complex networks i) causality methods for large scale networks ii) example applications in smart grids

References:

[1] J. Pearl, Causality, Cambridge University Press, 2009.

[2] A. Barabasi, Network Science, Cambridge University Press , 2016.

[3] F. Bullo, Lectures on Network Systems, CreateSpase, 2018. Class lectures and other material and research papers will be available online for download.

Schedule and room: late spring/summer 2021, lectures exact days and time will be published in Class Schedule

[back to **Summary**]

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: familiarity with basic probability. Knowledge of network theory also helps, but it is not a requirement.

Examination and grading: a final project or a take-home exam.

IE 21. Real-Time Systems and Applications

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Dr. Gabriele Manduchi, Consiglio Nazionale delle Ricerche

e-mail: gabriele.manduchi@igi.cnr.it

Aim: The course will provide an insight in the realm of real-time system. Knowledge in this field is normally fragmented and scattered among different engineering disciplines and computing sciences, and the aim of the course is present aspects related to theory and practice in a way which is holistic enough to prepare graduates to embark on the development of real-time systems, frequently complex and imposing safety requirements

Topics:

- Introduction via a case study: a system tracking circular objects;
- Operating Systems review;
- Another case study: the Spectre vulnerability
- Tasks, threads and interprocess communication;
- Real-time scheduling: definitions, cyclic executive, utilization based scheduling, response time analysis, priority inheritance;
- Data Acquisition techniques;
- GPU and FPGA and ZINQ architectures in real-time applications

References:

[1]I C Bertolotti, G Manduchi. Real-Time Embedded Systems. Open Source Operating Systems Perspective. CRC Press, 2012

[2]G C Buttazzo. Hard Real-Time Computing Systems. Predictable Scheduling Algorithms and Applications. Springer 2005.

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Basic knowledge of Operating System concepts.

Examination and grading: Each student will develop a survey report based on one or several articles related to the material covered in class and referring to some field of application for real-time systems.

IE 22. Bayesian Machine Learning

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Prof. Giorgio Maria Di Nunzio

e-mail: dinunzio@dei.unipd.it

Aim: The course will introduce fundamental topics in Bayesian reasoning and how they apply to machine learning problems. In this course, we will present pros and cons of Bayesian approaches and we will develop a graphical tool to analyse the assumptions of these approaches in classical machine learning problems such as classification and regression.

Topics:

- Introduction of classical machine learning problems.
  - Mathematical framework
  - Supervised and unsupervised learning
- Bayesian decision theory
  - Two-category classification
  - Minimum-error-rate classification
  - Bayes decision theory
  - Decision surfaces
- Estimation
  - Maximum Likelihood Estimation
  - Expectation Maximization
  - Maximum A Posteriori
  - Bayesian approach
- Graphical models
  - Bayesian networks
  - Two-dimensional visualization
- Evaluation
  - Measures of accuracy

References:

[1] J. Kruschke, Doing Bayesian Data Analysis: A Tutorial Introduction With R and Bugs, Academic Press 2010

[2] Christopher M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer 2007

[3] Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification (2nd Edition), Wiley-Interscience, 2000

[back to **Summary**]

[4] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin, Learning from Data, AMLBook, 2012 (supporting material available at http://amlbook.com/support.html)

[5] David J. C. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003 (freely available and supporting material at http://www.inference.phy.cam.ac.uk/mackay/

[6] David Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012 (freely available at http://web4.cs.ucl.ac.uk/staff/D.Barber/pmwiki/pmwiki.php?n=

[7] Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012 (supporting material http://www.cs.ubc.ca/ murphyk/MLbook/)

[8] Richard McElreath, Statistical Rethinking, CRC Presso, 2015 (supporting material https://xcelab.net/rm/statistical-rethinking/)

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements: Basics of Probability Theory. Basics of R Programming.

Examination and grading: Homework assignments and final project.

IE 23. Learning from Networks

Course Area: Information Engineering

Credits: 5 (10 lectures, 2 hours each)

Instructor: Prof. Fabio Vandin

e-mail: fabio.vandin@unipd.it

Important note: Not offered in a.a. 2020/21

[back to **Summary**]

IE 24. Advanced topics in scientific and parallel programming with practical application to the CAPRI HPC infrastructure

Course Area: Information Engineering

Credits: 5 (20 hours)

Instructor: Giacomo Baruzzo, Department of Information Engineering, University of Padova

e-mail: giacomo.baruzzo@unipd.it

Aim: Provide basic skills for working on remote servers, using/developing parallel software and deploying it on a containerized computer server. The course gives basic introduction to modern computer architecture and to the most important parallel programming paradigms: Multi-threading, OpenMP, MPI and CUDA with examples (mostly Python). The course covers basic tools to access and to interact with remote servers, to manage remote resources, and to manage jobs. The course introduces principles of software containerization from the perspective of users, providing practical examples of Docker and Singularity. The concepts discussed are applied to simple case of studies involving writing and/or running parallel programs using the CAPRI HPC infrastructure (256 cores, 6TB shared RAM and 2 GPU Nvidia P100) recently acquired by the University of Padova for research activities.

Topics:

1. How to use a computing server (application to CAPRI)
    a. Introduction to High Performance Computing (HPC hardware and architectures, HPC software, supercomputers)
    b. Job scheduling (slurm; writing a job; running, stopping and querying status of a job)
    c. The CAPRI queuing system and policy (CAPRI hardware and architecture; access to CAPRI and projects; execution queue; how to choose queue)
2. Containerization (singularity)
    a. Overview of containerization (definition of containers and container daemon; singularity and docker software; containers vs virtual machines; advantages: re-usability and reproducibility, flexibility, efficiency; disadvantages: learning curve)
    b. Using container that have already been defined (running, stopping, and resuming containers; containers options and flags)
    c. Defining new containers (new containers from scratch; extending existing containers)
    d. Sharing containers and the container repository (browsing and adding a container to the repository; guidelines for creating and documenting containers to be shared)
3. Version control (git)
    a. Basic operations (create a git repository, staging and committing changes, repository status and history, work with branches)
    b. Advanced operations and remote repository (clone a remote repository, work with a remote repository, GUI for git, git web-based hosting services)
4. Parallel architectures and multi-process/parallel programming

a. Introduction to parallel programming models and architectures (basic definitions; shared vs distributed memory architectures; threading: CPU and GPU; shared memory programming; message passing programming; performance metrics)
b. Parallel programming languages and frameworks (multi-threading; OpenMP; MPI; CUDA)

5. Hands on example (a simple parallel software for data analysis / machine learning; a simple parallel software for numerical analysis; students' proposals)

References:

- Eijkhout, V. (2013). Introduction to High Performance Scientific Computing. Lulu. com.
- Grama, A., Kumar, V., Gupta, A., & Karypis, G. (2003). Introduction to parallel computing. Pearson Education.
- Parhami, B. (2006). Introduction to parallel processing: algorithms and architectures. Springer Science & Business Media.
- Ad-hoc material by Lecturer

Schedule and room: please, see Class Schedule

Course requirements: Basics usage of tools for run/develop of scientific software (preferable unix platforms)

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Examination and grading: Each student must produce a small parallel and containerized software (either predefined or custom built container) related to her/his research field. Each student can either a) write a simple parallel software with one of the programming paradigm presented during the course using a language of choice or b) choose a (possibly parallel) software typically used in the research activity. Containerized software must run on the CAPRI server.

IE 25. Communicating using quantum entanglement: teleportation and the Quantum Internet

Course Area: Information Engineering

Credits: 4 (16 hours)

Instructor: Prof. Paolo Villoresi, DEI and Padua Quantum Technologies Research Center, University of Padova.

e-mail: paolo.villoresi@dei.unipd.it

Aim: Quantum communication protocols address at the most fundamental level of resources for communications. Although involving a more challenging implementation technology then for classical systems, they offer wider and unprecedented opportunities in term of novel schemes. These leverage of the fundamental principles of Quantum Mechanics, with a characteristic trait in the quantum entanglement between communication agents – photons. The exploitation of this resource is elaborated in this Course, consisting of an introduction to the concepts and formalism, to the generation and measure of entangled states, of a Lab session on teleportation and of perspectives on the so-called Quantum Internet under development worldwide – and for which UniPD is playing a leading role.

Topics:

- Recall of quantum description of radiation.
- Entangled systems with discrete variables.
- Generation of correlated and entangled states.
- Quantum measurements on entangled states.
- Teleportation protocols.
- Modeling of a quantum channel for entangled pairs.
- Lab session on teleportation.
- Perspective on the development of the Quantum Internet in Space and on ground.

References:

[1]P. Villoresi, “Communicating using quantum entanglement: teleportation and the Quantum Internet", lecture notes (will be posted on the e-learning page of the course)

[2]John Garrison and Raymond Chiao, Quantum Optics, Oxford University Press, 2008

Schedule and room: please, see Class Schedule

Enrollment: students must enroll in the course using the Enrollment Form on the PhD Program eLearning platform (requires SSO authentication).

Course requirements:

[back to **Summary**]

- basic notions of Quantum Information.

Examination and grading: Homework assignments and final test.

Alphabetical List of Course Instructors

[back to **Summary**]