ESIEE Paris
International Master of Computer Science IMC
(Diplôme d'Ingénieur)
Official program for years 2016 to 2018

Embedded Real-Time Computing

- ▷ Real-Time Systems
- ▷ Multiprocessor Scheduling
- ▷ Embedded Operating Systems

Controlability

- ▷ Signal and Systems
- ▷ Optimal Estimation and Control
- ▷ Distributed Control Systems

Connectivity

- ▷ Networking
- ▷ Wireless Networks for Intelligent Systems
- ▷ Localization workshop

Machine Learning

- ▷ Machine Learning 1 : Bayesian Theory
- ▷ Machine Learning 2 : Statistical Learning
- ▷ Machine Learning 3 : Structural Risk Minimization

Digital Imaging

- ▷ Mathematical Morphology
- ▷ Computational Geometry
- ▷ Computer Vision
- ▷ Computer Graphics on Android

Safety

- ▷ Model Checking
- ▷ Critical Application Development

ESIEE PARIS

une école de la CCI PARIS ILE-DE-FRANCE

IMC Programme

The **I**nternational **M**aster of **C**omputer Science recruits students holding a Bachelor's degree in Computer Science or an Engineering degree with a background in Computer Science. The program aims to provide the skills required in the design of "Intelligent Systems", i.e. systems that are able to interact with their operational environment. After a common core semester, graduates may broaden their knowledge in a chosen field of interest (e.g. machine vision, embedded real-time systems, machine learning). This core program is composed of courses guaranteed to be taught entirely in English, while flexibility is also incorporated by offering elective courses taught in either French or English.
A non exhaustive list of used programming languages and tools : C, python, C++, Matlab/Simulink, ANSYS SCADE, NuSMV model checker, Uppaal model checker, RTAI real-time operating system, OpenGL, Android, CAN bus, ns-network simulator, OpenCV, CUDA programming.

IMC Degree

Candidates succeeding the program are awarded a *Diplôme d'ingénieur* by ESIEE Paris, the highest degree awarded by France before a PhD. The Master degree has a double accreditation, an accreditation from the *Commission des titres d'ingénieur* (CTI) and one from the French ministry of Higher education.

IMC Faculty

The Master Faculty is composed of professors sharing their time between teaching and research. The research laboratories involved in the master are Laboratoire d'Informatique Gaspard-Monge (LIGM) and Laboratoire des Signaux et Systèmes (L2S).

First Year (E4)

Period 1 Sept. 2016 → Nov. 2016

Core Courses

- **Algorithms and Data Structures** (3 ECTS)
- **Networking** (3 ECTS)
- **Computer Architecture** (3 ECTS)
- **Signal and Systems** (3 ECTS)
- **Advanced Algorithms** (2.5 ECTS)

Management (3 ECTS)

- Communication 1 & Intercultural Management

Period 2 Nov. 2016 → Jan. 2017

Core Courses

- **Mathematical Morphology** (3 ECTS)
- **Graph and Algorithms** (3 ECTS)
- **Real-Time Systems** (3 ECTS)

Elective Courses

- **Optimal Estimation and Control** (3 ECTS)
  Alternative courses :
  - Development on Android Smartphones (Fr [a])
  - Introduction to Internet of Things (IoT) (Fr)
- **Operating Systems** (2.5 ECTS)
  Alternative courses :
  - Research workshop
  - Localization workshop (EF)
  - Workshop on Robotics (EF)
  - Image Analysis and Processing (Fr)
  - HCI Prototype and Ergonomics (Fr)

Management (3 ECTS)

- Introduction to Corporate Finance
- Introduction to International Marketing

Period 3 Jan. 2017 → May 2017

Core Courses

- **Machine Learning 1** (3 ECTS)
- **Optimization** (3 ECTS)

Elective Courses

- **Signal analysis with Python** (2.5 ECTS)
  Alternative courses :
  - Internet of Things (IoT) prototyping (Fr)
  - Workshop on Robotics (EF)
  - Web technology (JEE) workshop (Fr)
- **Computational Geometry** (3 ECTS)
  Alternative courses :
  - Language Theory and Compilers (Fr)
  - Distributed Real-time (Fr)
- **Model Checking** (3 ECTS)
  Alternative courses :
  - Wireless Networks for Intelligent Systems
  - Algorithmic Information Theory (Fr)
  - Multimedia Compression and Data Protection (Fr)

Management (3 ECTS)

- Communication 2
- French Business Culture

Period 4 May 2017 → Jul. 2017

Internship (3 months) or Project (2 months) (10 ECTS)

a. Fr=Taught in French, EF=Taught in English and French, Taught in en English when not mentioned

Second Year (E5)

Period 1 Sept. 2017 → Nov. 2018

Elective Courses

- **Distributed Control Systems** (3 ECTS)
  Alternative courses :
  - Parallelism and Distributed Computation (Fr)
  - Intelligent Vehicle (Fr)
- **Embedded Operating Systems** (3 ECTS)
  Alternative courses :
  - DotNet (Fr)
- **Machine Learning 2** (3 ECTS)
  Alternative courses :
  - Geometric Modeling and Computer graphics (Fr)
- **Critical Application Development** (3 ECTS)
  Alternative courses :
  - Information Systems (Fr)
  - Algorithms for Combinatorial Problems (Fr)

Management (3 ECTS)

- Communication 3

Period 2 Nov. 2018 → Jan. 2018

Elective Courses

- **Multiprocessor Scheduling** (3 ECTS)
  Alternative courses :
  - Architecture and Parallelism (EF)
  - Safe Development for Reactive Systems (Fr)
- **Machine Learning 3** (3 ECTS)
  Alternative courses :
  - Remote Embedded Control (Fr)
  - Multimedia and Vision (Fr)
- **Computer Vision** (3 ECTS)
  Alternative courses :
  - Software Reliability (Fr)
  - UML and Java Design (Fr)
- **Computer Graphics on Android** (3 ECTS)
  Alternative courses :
  - Cooperative Embedded Systems with Java (Fr)
  - Mobile Computer Science and Applications (Fr)

Management (3 ECTS)

- International Project Management
- Innovation Management

Period 3 and 4 Jan. 2018 → Jun. 2018

Internship (6 months) (30 ECTS)

French Language (E4, E5)

E4 : volume 80H, 7.5 ECTS, E5 : volume 50H, 4 ECTS

1 Technical Course details – First year ('E4')

Algorithms and data structures (Xavier Hilaire)

Content : Linked lists ; Binary search, red-black, and AVL trees ; Queues ; Heaps ; Complexity analysis ; Sorting and searching ; Algorithms on numbers ; Algorithms on strings ; Reminders on C language, compilation, Makefiles, and SVN.

Aim : The aim of this course is to provide the basics of programming to students having a limited background in computer science. We cover the essential and must-kwnown algorithms in the field, such as sorting, searching, or simple algorithms on arbitrary large numbers. Relevant data structures are presented within the context of the algorithm they are bound to. The course has a highly practical orientation, and involves many hours of concrete programming in C language, on UNIX platform.

Networking (Lynda Zitoune)

Content : OSI model, layer abstraction ; Paquet switching (OSI layer 2) ; Routing (CISCO routers, OSI layer 3) ; WAN (Wide Area Network) and VPN (Virtual Private Network).

Aim : This course provides attendees with the most essential concepts from low to mid-level networking. The most widespread networking technologies are first introduced. It is then shown how these different technologies may be abstracted in the first layers of the OSI model. Routing is considered, with concrete example given on CISCO routers. Large networks, and their administration, are explained at last.

Computer architecture (Mohamed Akil)

Content : Computer architectures and performances ; Architecture of RISC processors ; Memory hierarchy ; YoDSP Programming

Aim : This course is a first introduction to computer architecture and its impact on performances. More precisely, we study how the structure of a program, once implemented on a particular architecture, can impact on performances. We present a design methodology that permits to obtain an optimized implementation (on RISC or DSP) of a program given its algorithmic specification

Signal and Systems (Arben Cela)

Content : Fourier and Laplace transforms ; Theory of convolution ; Sampling and reconstruction ; Dynamical Systems ; Integration methods (Euler , Runge -Kutta , Adams) ; Stability ; Controllability, Observability.

Aim : Introduction to necessary mathematical tools for analysis of signal and dynamic systems. Different concepts such as stability, controllability and observability are introduced through different well known application examples. Mastering of the related Matlab/Simulink toolboxes, introduced through different applications, is another objectives of this course. The course also aims at introducing the basics of continuous-time and discrete-time signals and systems, such as impulse and frequency responses, the Fourier Transform, Fourier Series, Discrete Fourier Transform, Finite Fourier Transform, as well as the Z-Transform and the Function Transfer of a system. The course also involves, as part of it, the study and design of digital filters.

Advanced Algorithms (Xavier Hilaire)

Content : Dynamic programming ; Divide and conquer ; Greedy algorithms ; Introduction to NP-completeness and approximation algorithms.

Aim : This course is a continuation to "Algorithms and Dada structures", and pertains specifically on problems solving. We present resolution methods from three widespread families of algorithms, all providing either exact, or approximate but guaranteed solutions within a given tolerance. Typical examples are provided in the case of stock management, transportations, or resource assignment problems.

Mathematical morphology (Jean Cousty)

Content : Non-linear signal processing ; Erosion, dilation, closing, opening ; Skeletons ; Watershed segmentation, connected operators.

Aim : The aim of this course is to provide the fundamentals of mathematical morphology. We introduce new concepts in non-linear signal analysis, then explain the basic operators used in mathematical morphology and their main properties, and skeletonization. The problem of image segmentation is then considered, with the very popular watershed segmentation approach. Non-linear filtering and detection are illustrated on a wide variety of problems.

Graph Algorithms (Jean Cousty)

Content : Graph traversal, connected components ; Shortest path ; Minimum spanning tree ; Maximal flow.

Aim : This course is an introduction to the most popular algorithms produced by graph theory, and used in pattern recognition, combinatorics, AI, and problem resolution amongst others. It aims to provide attendees with the ability to : formalize a given problem in terms of graphs ; identify whether the problem has a known solution or not ; and in case not, suggest a new algorithm and evaluate its complexity.

Real-Time Systems (Yasmina Abdeddaïm)

Content : Real time scheduling algorithms ; Feasability analysis ; Optimality analysis ; Resource sharing ; Real-time Linux (RTAI).

Aim : This course is an introduction to scheduling for hard real-time systems. We introduce the task model, the classical scheduling algorithms, and feasibility analysis based on this model. Scheduling algorithms will be tested in practice on a real-time kernel.

Optimal Estimation and Control (Arben Cela)

Content : Nonlinear optimization ; Dynamic programming ; Optimal estimation/observation of linear (discrete/continuous) systems ; Robust control and controller construction.

Aim : Formulation and resolve a general non-linear constrained optimization problem and illustrate it through simple and relevant examples from different application domains. Design and implement an optimal estimator and controller for dynamic system as well as an optimal scheduler for real-time systems. Special emphases will be put on mastering Matlab optimization toolbox as well as on relevant application domain such as energy optimization.

Computational Geometry (Nabil Mustafa)

Content : The main topics covered will be : convex hulls, object intersections, point location, Delaunay triangulations and Voronoi diagrams.

Aim : This is an introductory course on computational geometry and its applications. The objective is to learn how to use the structure of geometric data to design better algorithms.

Machine learning 1 (Xavier Hilaire)

Content : Reminders on probabilities (PDFs, types of convergence, central limit theorem, marginalization, ...) ; Basics of classification. Naïve Bayes. LQDA. ; Parametric estimation : MLE, EM, MAP, Bayesian inference, maximum entropy ; Non-parametric estimation : Parzen windows, k-NN ; PAC learning. Learning from uniform convergence. ; The bias-complexity trade off. VC dimension. ; Structural risk minimization, and model selection.

Aim : This course covers the fundamentals of pattern classification, and provides a first taste of machine learning by introducing some of the its most important and fundamental results – those pertaining on generalization issues. We first explain Bayesian decision theory, and show how a given pattern classification problem may be expressed in terms of probabilities and distributions. We then study various well-known techniques usable to solve the problem raised as the outcome of formalization. Essential results of machine learning are presented next. Those address the following questions : what is learning ? How fast can a given algorithm learn ? How well will it generalize on unseen data ? Which model should be selected for a given problem or data set ? Practical sessions (of 2 or 4 hours) are organized after each chapter. Those involve programming using either scikit-learn (Python) or the Matlab ML toolbox.

Optimization (Hugues Talbot)

Content : Linear programing ; integer programs and mixed integer-linear programs ; simplex algorithm ; branch and bound and cutting plane techniques.

Aim : This course provides an introduction to linear programming (LP), which is one of the easier and more versatile tool capable of modeling both objective functions and constraints. We study modeling ; the relationship between objectives and constraints ; the simplex algorithm ; duality and limit cases. In a second part, we study integer programming (IP), which is capable of modeling and solving most combinatorial problems, including NP-complete problems. This is essential in computer science because such problems are very common. Many examples are given. In a third part, we study transport problems, which is a class of IP problems that can be solved exactly and very efficiently. It is essential to be able to recognize such problems. In a last, short section, we provide an introduction to nonlinear, convex programming.

Model Checking (Yasmina Abdeddaïm)

Content : Reactive Systems Modeling ; Temporal Logics (CTL, LTL, CTL*) ; Model Checking Algorithms ; NuSMV Model Checker.

Aim : This course is an introduction to model checking, an automatic verification technique of concurrent and reactive systems. First we introduce the Kripke structure as a model of reactive concurrent systems, then the linear (LTL) and branching time (CTL) temporal logics used to model temporal specifications. Finally we present LTL and CTL model checking algorithms and the model checker tool NuSMV.

Signal analysis with Python (Jean-François Bercher)

Content : Representations of signals and systems ; filtering ; Fourier analysis ; Random Signals ; Adaptive filters ; Array Processing.

Aim : The objective of this formation is to review the main concepts of signal processing using the scientific Python toolchain. By the end of this course, students will be able to : describe and apply the main operators and methods of signal processing ; Design and implement simple signal processing solutions using the scientific Python toolchain ; Go on learning by themselves both the Signal Processing tools and improve their skills on Scientific Python computation and visualization.

2 Technical Course details – Second year ('E5')

Distributed Control Systems (Arben Cela)

Content : Introduction to Distributed Control Applications ; Modeling and analysis of Distributed Control System ; Model Based Design of distributed real time application in Automotive Industry ; Applications of Distributed Real Time Systems.

Aim : The main objective of this course is to give the necessary tools for modeling, analysis and design of Distributed Control Systems. Classification of difference communication and calculation architectures help to achieve the modeling/analysis objectives and allows to point out the advantages/disadvantages w.r.t performance specification.

Embedded Operating Systems (Yasmina Abdeddaïm)

Content : Booting Linux kernel using U-Boot, Root files system ; Busybox, Autotools ; Cross-compilation ; Minimal image for a Raspberry Pi ; Customization of an image.

Aim : This course aims to present the methods and tools needed to build a GNU/Linux operating system using the source code.

Machine Learning 2 (Xavier Hilaire)

Contents : Dimensionality reduction techniques : the Karhunen-Loève transform, PCA, LDA, ICA, SVD. ; Kernel methods ; Boosting ; Mutlticlass classification ; Online learning ; Clustering.

Aim : This course is a continuation to Machine leaning 1, which is assumed fully absorbed by attendees as a prerequisite. We first return to the problem of bias-dimensionality trade off, and present classical techniques from the literature in conventional data analysis. The benefits of the "kernerlized" version of some of them, are studied next. We finally study advanced techniques known for their computational efficient in numerous cases, namely : tricks for multiclass problems, online (reinforcement) learning, clustering, and boosting. As for Machine learning 1, practical sessions (of 2 or 4 hours) are organized after each chapter. Those involve programming using either scikit-learn (Python) or the Matlab ML toolbox.

Critical application development (Yasmina Abdeddaïm)

Contents : Synchronous paradigm ; Esterel Synchronous Language ; Lustre Synchronous Language ; SCADE software.

Aim : Synchronous languages are used for the development of embedded reactive systems which are at the same time complex and critical systems. These languages are currently used successfully in industry, for example, EADS uses the software SCADE, based on the language Lustre, to program the embedded software flight control of Airbus. In this course, the fundamental principles of synchronous programming are presented then two languages, with different paradigm, are particularly studied : Lustre and Esterel, finally the software SCADE is presented.

Architecture and parallelism (Thierry Grandpierre)

Contents : Study of architecture classes of : RISC, CISC ; scalar ; superscalar ; vectorial ; VLIW ; DSP ; SoC.

Aim : Intraprocessor parallelism. ; Optimized programming of Pentium/iCore architectures, and their relevant units (SSE2.../Altivec) ; Initiation to CUDA programming, with applications to image transformations (histograms, color space transforms, rotations, a.s.o) ; CELL processor programming : 9-core based architecture, 8-core "DSP" programming, communicating using shared memory and message passing between processors ; Introduction to multicore programming (INTEL) with OpenMP.

Machine Learning 3 (Giovanni Chierchia)

Content : Linear regression, regularization, kernel trick, and linear model selection ; tree-based methods for regression and classification ; neural networks, and introduction to deep learning.

Aim : This course continues the study of machine learning concepts. We begin with linear regression, the fundamental starting point for modeling the relationship between a scalar dependent variable with one or more explanatory variables. We then move into the world of nonlinear modeling by considering tree-based methods for both regression and classification. Finally, we revisit neural networks in the context of deep learning. Each subject is illustrated in a laboratory session using simulated and real examples.

Computer Graphics on Android (Nabil Mustafa)

Contents : Android development environment ; getting and using sensor data in Android systems ; OpenGL ES 3.0 ; shading and texturing in OpenGL ES 3.0 using GLSL ES.

Aim : OpenGL Embedding Systems (ES) is an open source API that provides a strong and viable interface for the stimulation of software and graphics for Android games. It paves the way to making complete games for Android devices solely using the Android SDK. The aim of this course is to learn OpenGL ES 3.0, and its use in Android systems in development of graphics games, with the sensory inputs given by accelerometers, gravity sensors, gyroscopes, and rotational vector sensors present in mobile devices.

Computer Vision (Xavier Hilaire)

Contents : 2D and 3D projective transformations ; Camera calibration ; Epipolar geometry ; Dense and spare stereo vision algorithms ; Trifocal tensor.

Aim : This course introduces the fundamentals of computer vision : projective geometry, homographies, and single view cameras are first deeply studied to provide good mathematical grounds to attendees. Mathematical notions on eigenvalues, SVD, least squares, and constrained optimization will be recalled. Stereo vision and epipolar geometry are presented next, along with classical point matching algorithms and some efficient 3D object recognition algorithms. Extension to n cameras, and related open problems, are presented at last. The course includes a number of practical lab sessions on OpenCV, during which students will learn to : calibrate cameras, build a panoramic view from several overalaping shots, recognize and track a 3D object.