الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne
Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement
Supérieur
et de la Recherche Scientifique

National Higher School of Autonomous Systems Technology

STATE ENGINEER TRAINING PROGRAM

Academic year: 2025/2026

Institution
National Higher School of Autonomous Systems Technology

Domain	Field	Specialty
Science and Technology	*Automatic*	*Robotics and Autonomous Systems Design*

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

اللجنة البيداغوجية الوطنية
لميدان العلوم و التكنولوجيا
Comité Pédagogique National du Domaine Sciences et Technologies

عرض تكوين
مهندس دولة

المؤسسة
المدرسة الوطنية العليا لتكنولوجيا الأنظمة المستقلة

الميدان	الفرع	التخصص
علوم وتكنولوجيا	الية	تصميم الروبوتات والأنظمة الذاتية المستقلة

Table of contents

Training Context and Objectives

1-Overview

The 'Robotics and Autonomous Systems Design' engineering degree is a comprehensive 5-year program designed to equip students with the technical expertise and interdisciplinary skills needed to design, develop, and deploy advanced robotic and autonomous systems. The program aims to provide a strong foundation in control, mechanical, electrical, and computer engineering, while emphasizing the integration of artificial intelligence, and machine learning into real-world applications. Students will gain hands-on experience in designing intelligent systems, together with programming autonomous functionalities, and integrating sensors and actuators for perception and action. The curriculum focuses on fostering innovation, problem-solving, and project management skills, preparing graduates to tackle complex challenges in industries such as manufacturing, transportation, and smart cities. Additionally, the program emphasizes ethical considerations, regulation, safety standards, and the societal impact of autonomous technologies, ensuring that graduates are not only technically proficient but also socially responsible. By combining theoretical knowledge with practical experience, the program aims to produce engineers capable of leading the development of cutting-edge robotic and autonomous systems that address global challenges and drive technological advancement.

2. Significance of the Proposed Program

Given the accelerating pace of technological innovation and the growing reliance on automation and robotics, there is a strong need for engineers with expertise in designing and optimizing robotic systems, autonomous machines, and integrated control systems. The proposed 5-year program in Robotics and Autonomous Systems Design will address this demand by equipping students with a comprehensive set of skills in automation, systems control, robotics, artificial intelligence, and the necessary foundational engineering knowledge.

3. Vision for the Training Program

3.1 Mission Statement

The mission of the Robotics and Autonomous Systems Design program is to cultivate a new generation of engineers proficient in designing, implementing, and optimizing automated and autonomous systems across diverse applications. The program will focus on the integration of systems engineering principles with advanced robotics technologies to create innovative solutions for real-world challenges.

3.2 Strategic Vision

3.2.1. Prepare Students for the new Industrial era

Industry 4.0, the current trend of automation and data exchange in manufacturing technologies, relies heavily on robots and autonomous systems. This program will position graduates as key players in shaping and driving forward Industry 4.0 technologies, ensuring they are well-versed in cutting-edge automation, robotics, and control systems.

3.2.2. Hands-On Learning and Practical Training

The program will emphasize practical experience through labs, projects, and internships, ensuring students not only learn theoretical concepts but also apply them to real-world scenarios. Collaboration with industry partners, research centers, and startups will be a core feature, allowing students to engage in cutting-edge RAS research and product development.

3.2.3. Multidisciplinary Approach

Robotics and autonomous systems design requires knowledge in multiple fields: electrical and mechanical engineering, computer science, machine learning, control theory, and sensor integration. This multidisciplinary approach will enable students to gain expertise in designing and optimizing complex systems that involve hardware and software components.

3.2.4. Promote Ethical and Sustainable Engineering

As robotics and autonomous systems have a direct impact on society, it is crucial that students understand the ethical and social implications of their designs. The program will incorporate courses on the societal impact of automation, safety, privacy concerns, and sustainable development, ensuring that graduates approach technological challenges with a responsible and human-centered mindset.

3.2.5. Innovation and Research

The program will foster innovation through research-oriented projects, encouraging students to explore emerging technologies like swarm robotics, bio-inspired robotics, and machine learning-based control systems. Students will be exposed to a variety of research methodologies to push the boundaries of existing technologies and develop the next generation of autonomous systems.

4. Admission requirements:

Candidates for this engineering program are students from the National Higher School of Autonomous Systems Technology who have successfully completed their preparatory cycle. Selection is based on merit ranking, according to the overall average marks of the two preparatory years subject to places availability. Admission to the Robotics and Autonomous Systems Design specialty follows the institution’s internal progression protocols.

5. Program objectives:

Upon Completion of the Program, the student should be able to:

- **Design and Development of Robotic Systems:** Ability to design, build, and integrate hardware and software for robotic and autonomous systems.
- **Programming and AI Expertise:** Proficiency in coding (Python, C++, ROS) and applying AI/ML techniques for perception, decision-making, and control.
- **Control Systems and Automation:** Mastery of control theory and algorithms for stable and efficient system operation.
- **Sensor Integration and Data Processing:** Skills in using sensors (e.g., LiDAR, cameras) and processing data for real-time decision-making.
- **Simulation and Modeling:** Experience with simulation tools to test, validate, and optimize robotic systems.
- **Interdisciplinary Problem-Solving:** Ability to combine mechanical, electrical, and software engineering principles to solve complex challenges.
- **Project Management and Teamwork:** Skills in managing projects, collaborating in teams, and delivering results in multidisciplinary environments.
- **Ethics and Safety Awareness:** Understanding of ethical considerations and safety standards in robotics and autonomous systems.

6. Targeted profiles and Competencies:

6.1 Targeted Profiles:

1. Robotics Engineers: Capable of designing and developing intelligent robotic and autonomous systems.

2. Industry-Ready Professionals: Aspiring engineers seeking to lead projects in sectors like manufacturing, healthcare, and smart technologies.

6.2 Targeted Competencies:

1. Robotics System Design: Skills in designing, integrating, and optimizing hardware and software for autonomous systems.

2. AI and Programming Expertise: Proficiency in coding (Python, C++, ROS) and applying AI/ML for perception, decision-making, and control.

3. Interdisciplinary Integration: Ability to combine mechanical, electrical, and software principles to create cohesive robotic solutions.

4. Problem-Solving and Innovation: Strong analytical and creative skills to develop cutting-edge solutions for real-world challenges.

5. Project Management: Experience in planning, executing, and managing multidisciplinary robotics projects.

6. Ethics, regulation and Safety: Understanding of ethical, regulation, safety, and societal implications of autonomous technologies.

7. Regional and national employability prospects

This engineering degree offers strong employability prospects across the entire national territory. Indeed, the field of robotics and artificial intelligence opens doors to research centers, academia and research (Ph.D.), and is at the heart of many PMEs, PMIs, and multinational corporations, whether in design offices or management roles.

Among others, opportunities can be found in:

- **Chemical and petrochemical industries.**
- **Steel and metallurgy industries.**
- **Mechanical engineering and automotive industries.**
- **Hydraulic industries and seawater desalination.**
- **Processing, textile, and manufacturing industries.**
- **Food processing industries.**
- **Pharmaceutical industries.**
- **Healthcare sector.**
- **Construction materials industries.**
- **Electrical energy production and distribution sector.**
- **Renewable energy sector.**
- **Agriculture.**

8.- Bridges to other specialties

Robotics and automation intersect with various other specialties in the field. By providing flexible pathways, the RASD program not only prepares students for careers in robotics but also equips them with specialized skills to work in related fields, driving innovation across industries like healthcare, manufacturing, and AI-driven automation. These bridges will empower graduates to become versatile engineers, ready to tackle complex challenges in the emerging world of intelligent systems.

The RASD program can offer bridges to the following disciplines:

1. Artificial Intelligence (AI) and Machine Learning (ML)
2. Electrical and Electronics Engineering
3. Mechanical Engineering
4. Computer Science and Software Engineering
5. Biomedical Engineering
6. Data Science and Big Data
7. Cybersecurity.

9. Program monitoring indicators

These points can be used to assess the success and adaptation of the program to the needs of the job market and students.

Professional integration rate

This rate measures the proportion of graduates who have found employment within six months of completing the program. It helps assess the program's effectiveness in preparing students for the job market.

Student preferences for the proposed program

This indicator measures the demand and interest in the program by tracking the number of applications or choices made by students to pursue the offered track.

Feedback from companies hosting interns

Feedback from companies that have hosted interns from the program is essential for evaluating the quality of students' skills and performance in a professional context.

Feedback from companies hiring graduates from the program

Companies that have hired graduates from the program provide valuable feedback on how well students are prepared to integrate into the workforce and meet industry needs.

Feedback from students at the end of the program

Feedback from students at the end of their studies helps assess their overall satisfaction with the program, the relevance of the curriculum, and the practical and theoretical skills they have acquired.

Feedback from students at the beginning of the program

Feedback from students at the start of their studies helps measure their expectations of the program, their preparedness, and their motivation to engage in this educational path.

Graduate feedback

Long-term feedback from graduates provides insight into the impact of the program on their professional careers, their progression in the field, and the relevance of the skills they gained during their studies.

10 – Human resources available

A : Enrollment Capacity

150 Students (50 per promotion)

B-3 : Comprehensive Summary of Human Resources

Rank	Internal ressource	External ressource	Total
Professeur	**4**	**3**	**7**
MCA	**7**	**2**	**9**
MCB	**5**	**7**	**12**

B-4 : Permanent Support Staff

Rank	Staff
Laboratory Engineer	**1**
Computer Engineer	**1**
Administrator	**4**
Administrative Assistant	**2**
Total	**8**

11. Available Material Resources

A- Educational Facilities

The National Higher School of Autonomous Systems Technology provided facilities play a crucial role in supporting students' academic, personal, and professional development. These facilities include well-equipped classrooms, laboratories, and workshops designed to foster hands-on learning and collaboration. A Library with academic resources, study spaces, and digital tools are available to support research and self-directed learning. Additionally, dedicated spaces for practical training, such as labs or engineering workshops, enable students to apply theoretical knowledge in real-world contexts.

Location	Seatting Capacity	Number	Total Capacity
Lecture Hall	200	4	800
Tutorial Room	30	15	450
Practical Lab	25	15	375
Library	30	1	30
Reading Rooms	40	2	80
Workshop	10	1	10
Computer Center	40	1	40
Internet Room	40	2	80

B- Educational Laboratories and Equipment:

The National Higher School of Autonomous Systems Technology prioritizes hands-on learning experiences through well-equipped laboratory facilities. These educational environments are designed to support the practical application of theoretical concepts across all specializations offered by the institution.

As a newly established educational institution, the school is currently in the process of acquiring high-quality pedagogical equipment and materials to outfit its laboratories. This ongoing procurement initiative is strategically aligned with the specific requirements of each specialization and the overall educational objectives of the institution.

The equipment acquisition plan has been carefully developed to ensure that all laboratories will be furnished with state-of-the-art technology that meets international standards for engineering education. This deliberate approach to laboratory development reflects the institution's commitment to providing students with relevant, industry-aligned practical experiences that enhance their technical competencies and professional readiness.

Upon completion, these specialized laboratories will create comprehensive learning environments where students can effectively bridge theoretical knowledge with practical application, developing the hands-on skills essential for success in the autonomous systems sector.

- **Electricity Laboratory**

Number of students: 25

N°	Equipement	Quantity	Operational Status
1	Amperemeter	13	Excellent state
2	Galvanometer	02	Excellent state
3	Voltmeter	12	Excellent state
4	Multimeter	05	Excellent state
5	Wattmeter	07	Excellent state
6	Power supply	10	Excellent state
7	Power supply module	06	Excellent state
8	Low-Frequency Generator	06	Excellent state
9	Oscilloscope	10	Excellent state
10	Breadboards	20	Excellent state

- **Physics Laboratory 1**

Number of students: 25

N°	Equipment	Quantity	Operational Status
1	Force tables	01	Excellent state
2	Free Fall	04	Excellent state
3	Simple Pendulum	05	Excellent state
4	Air track	03	Excellent state
5	PasCars	02	Excellent state
6	Torsion pendulum	02	Excellent state
7	Balance of Coulomb	05	Excellent state
8	Electric field strengthmeter	05	Excellent state
9	Tank Rheographic	04	Excellent state
10	Analogic voltmeter	07	Excellent state
11	Analogic Ammeter	07	Excellent state

- **Physics Laboratory 2**

Number of students: 25

N°	Equipment	Quantity	Operational Status
1	Pohl's Pendulum	01	Excellent state
2	Wave Tank/Ripple Tank	01	Excellent state
3	Vibrating String/Cord	01	Excellent state
4	Kundt's Tube	01	Excellent state
5	Light Polarization (Equipment)	01	Excellent state
6	Light Diffraction (Equipment)	01	Excellent state

- **Chemistry Laboratory**

Number of students: 25

N°	Equipment	Quantity	Operational Status
1	Fume hood	01	Excellent state
2	Oven/Incubator	01	Excellent state
3	Distiller	01	Excellent state
4	Ice generator	01	Excellent state
5	Calorimeter	04	Excellent state
6	Heating plates	04	Excellent state
7	Heater with stirrer	01	Excellent state
8	Scale/Balance	02	Excellent state
9	pH meter	04	Excellent state
10	Dosing equipment with pH meter	02	Excellent state
11	Thermometer	15	Excellent state
12	Stopwatch/Timer	06	Excellent state
13	Power supplies for calorimetry	02	Excellent state
14	Voltmeter for calorimetry	02	Excellent state
15	Ammeter for calorimeter	02	Excellent state

- **Fluid Mechanics Laboratory**

Number of students: 25

N°	Equipment	Quantity	Operational Status
1	Digital Hydraulic Bench	01	Excellent state
2	Center of Pressure Apparatus	02	Excellent state
3	Falling Sphere Viscometers	02	Excellent state
4	Venturi Tube	01	Excellent state
5	Notch Discharge Apparatus	01	Excellent state

- **Mechanics Laboratory**

Number of students: 25

N°	Equipment	Quantity	Operational Status
1	Materials Testing Apparatus	01	Excellent state

C- Internship and Corporate Training Sites:

Training Sites

Institution	Student Number	Training Duration
SEAAL	**10**	15 days
SONATRACH	**20**	15 days
SONELGAZ	**20**	15 days
MOBILIS	**20**	15 days
ALGERIE-TELECOM	**20**	15 days
CDTA	**10**	15 days
CRTI	**10**	15 days

D- Personal Workspaces and Information Technology Resources

The institution provides students with a well-equipped library featuring a spacious reading room designed for academic study. High-speed internet connectivity is available throughout all campus facilities, ensuring continuous access to digital resources.

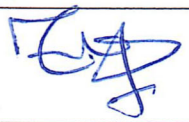
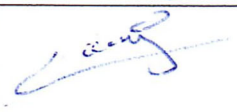
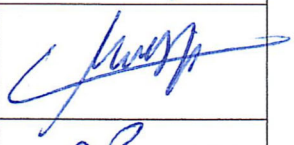
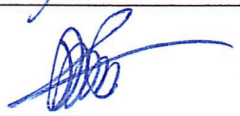
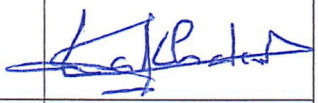
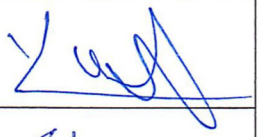
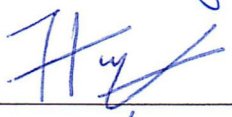
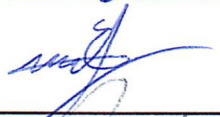
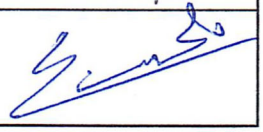
Signatures of teachers involved in the course

Internal teaching team for the specialty

Name	Speciality	Rank	Intervention Type	Signature
Mourad ADNANE	Instrumentation / Signal Processing	Prof	Course/Tutorials/Lab	
Faiza BOUMEDIENE	Mechanics / Robotics	Prof	Course/Tutorials/Lab	
Ziane KECHIDI	Physics	Prof	Course/Tutorials/Lab	
Ouassila HIOUAL	Computer Sciences	Prof	Course/Tutorials/Lab	
Kamel BOUDJIT	Electronics / Embedded Systems	MCA	Course/Tutorials/Lab	
Salaheddine AOUDJ	Chemistry / Electrochemistry	MCA	Course/Tutorials/Lab	
Tarek CHERIFI	Electronics / Embedded Systems	MCA	Course/Tutorials/Lab	
Abdelkader HAMTTAT	Mathematics	MCA	Course/Tutorials/Lab	
El mahdi MEDIA	Physics	MCA	Course/Tutorials/Lab	
Yasmine GUERBAI	Electronics / Artificial Intelligence	MCA	Course/Tutorials/Lab	
SAIB Abdessadek	Mathematics	MCA	Course/Tutorials/Lab	
Leila ABBAD	Electronics / Networking	MCB	Course/Tutorials/Lab	
Sarah CHENCHE	Computer Sciences/Artificial Intelligence	MCB	Course/Tutorials/Lab	
Mohamed LAIDI	Statistics / Stochastic processes	MCB	Course/Tutorials/Lab	
Halima LAMMARI	Mechanics / Robotics	MCB	Course/Tutorials/Lab	
Ahmed ZEGLAOUI	Mathematics / Optimization	MCB	Course/Tutorials/Lab	
Said REZIG	Mathematics	MAA	Course/Tutorials/Lab	
Fatiha KADIK	Mathematics	MAA	Course/Tutorials/Lab	
ZAIR Mustapha	Computer Sciences	MAA	Course/Tutorials/Lab	

School director's stamp

External teaching team for the specialty

Name	Specialty	Institution	Rank	Intervention Type	Signature
Moufid MANSOUR	Control Eng	USTHB	Prof	Course Responsible	
Mohamed TADJINE	Control Eng / Robotics	ENP	Prof	Course Responsible	
Abdelhalim ZAOUI	Electrical Eng	ENSTA	Prof	Course Responsible	
Malika BOUTERFAS	Electronics / Sensors	ENSTA	MCA	Course Responsible	
Tarek CHERIFI	Electronics / Embedded systems	ENSTA	MCA	Course Responsible	
Souhila BOUTARFA	- Electronics/ Signal processing	ENSTA	MCB	Course Responsible	
Nesrine ISSAAD	Bioinformatics / Artificial intelligence	U. Algiers	MCB	Course Responsible	
kheira Lakhdari	Telecommunication / Artificial intelligence	ENSTA	MCB	Course Responsible	
Fouad YACEF	Control Eng / Drones	CDTA	Dr	Course Responsible	
Rabie Riadh BENREZKI	Navigation and Control	CDTA	Dr	Course Responsible	
Abdellah KHELLOUFI	Navigation and Control	CDTA	Dr	Course Responsible	
Mohamed ZAOUCHE	Control Eng / UAS	Expert	Dr	Course Responsible	
Abdelkader BELAHCENE	Computer Science / Open Source software	ENSTA	MAA	Course Responsible	

School director's stamp

Semester 1 :

Teaching Unit TU	Semester hourly volume (15 weeks / Semester)					Coefficients	Crédits	Assessment Method	
	Lectures (H)	Tutorials (H)	Laboratory Work (H)	Total (H)	Personal work			Tutorial/Lab work	Final Exam
Core Teaching Unit (UEF)									
UEF1.1.1	**4H30**	**4H30**	**3H**	**12H**	**7H30**	**12**	**12**		
Introduction to System Dynamics and Control	1H30	1H30	1H	4H	2h30	4	4	(20% Tut + 20% Lab)	60%
Digital Signal Processing	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Mathematics for Engineers	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
UEF1.1.2	**3H**	**3H**	**2H**	**8H**	**5H00**	**8**	**8**		
Advanced Analog Electronics	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Introduction to Electrical Engineering	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Methodology Teaching Unit (UEM)									
UEM1.1.1	**3H**	**0H**	**3H**	**6H**	**4H00**	**6**	**6**		
Computational Modeling & Simulation for Engineering Design	1H30	0H	1H30	3H	2h	3	3	40% Lab	60%
Advanced Technical Design	1H30	0H	1H30	3H	2h	3	3	40% Lab	60%
UEM1.1.2	**1H30**	**0H**	**1H**	**2H30**	**1H00**	**2**	**2**		
Sensors and Instrumentation	1H30	0H	1H	2H30	1H00	2	2	40% Lab	60%
Cross-disciplinary Teaching Unit (UET)									
UET1.1.1	**1H30**	**0H**	**0H**	**1H30**	**0H30**	**1**	**1**		
Introduction to Entrepreneurship	1H30	0H	0H	1H30	0H30	1	1		100%
Discovery Teaching Unit (UED)									
UED1.1.1	**1H30**	**0H**	**0H**	**1H30**	**0H30**	**1**	**1**		
Basic Principles of Robotics	1H30	0H	0H	1H30	0H30	1	1		100%
Total Semester S1	**15H00**	**06H00**	**10H30**	**31H30**	**18h30**	**30**	**30**		

Semester 2

Unité d'Enseignement UE	Semester hourly volume (15 weeks / Semester)					Coefficients	Crédits	Assessment Method	
	Lectures (H)	Tutorials (H)	Laboratory Work (H)	Total (H)	Personal work			Tutorial/Lab work	Final Exam
Core Teaching Unit (UEF)									
UEF1.2.1	**3H**	**3H**	**2H**	**8H30**	**4H30**	**8**	**8**		
Introduction to Machine Learning	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Optimization Techniques	1H30	1H30	1H30	4H30	2H00	4	4	(20% Tut + 20% Lab)	60%
UEF1.2.2	**4H30**	**4H30**	**2H**	**11H**	**7H00**	**11**	**11**		
Microprocessors and Microcontrollers	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Digital Control Systems	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Mechanical Systems	1H30	1H30	0H	3H	2H00	3	3	40% Tut	60%
Methodology Teaching Unit (UEM)									
UEM1.2.1	**4H30**	**0H**	**3H30**	**8H**	**4H00**	**7**	**7**		
Computer Aided Design (CAD)	1H30	0H	1H30	3H	2H00	3	3	40% Lab	60%
Engineering programming	1H30	0H	1H	2H30	1H00	2	2	40% Lab	60%
Electric Actuators for Robotics	1H30	0H	1H	2H30	1H00	2	2	40% Lab	60%
Cross-disciplinary Teaching Unit (UET)									
UET1.2.1	**1H**	**0H**	**1H30**	**2H30**	**1H00**	**2**	**2**		
Introduction to UAVs Operations and Regulations	1H	0H	0H	1H	0H30	1	1		100%
Capstone Project I	0H	0H	1H30	1H30	0H30	1	1	100%	
Discovery Teaching Unit (UED)									
UED1.2.1	**0H**	**0H**	**1H30**	**1H30**	**2H00**	**2**	**2**		
Linux Operating System	0H	0H	1H30	1H30	0h30	1	1	100% Lab	
Training Internship I	/	/	/	/	1H30	1	1	100%	
Total Semestre S2	**13H00**	**07H30**	**10H30**	**31H30**	**18H30**	**30**	**30**		

Semester 3

Unité d'Enseignement UE	Semester hourly volume (15 weeks / Semester)					Coefficients	Crédits	Assessment Method	
	Lectures (H)	Tutorials (H)	Laboratory Work (H)	Total (H)	Personal work			Tutorial/Lab work	Final Exam
Core Teaching Unit (UEF)									
UEF2.1.1	**3H**	**3H**	**2H**	**8H**	**5H**	**8**	**8**		
Nonlinear and optimal control design	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Multi-Physics Modeling of Technological Systems	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
UEF2.1.2	**4H30**	**4H30**	**2H**	**11H**	**8H00**	**12**	**12**		
Robot Motion Planning	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Kinematics, Dynamics and Control of Robot Manipulators	1H30	1H30	0H	3H	3H00	4	4	40% Tut	60%
Power Electronics	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Methodology Teaching Unit (UEM)									
UEM2.1.1	**4H30**	**0H**	**3H30**	**8H**	**4H00**	**7**	**7**		
Mobile Robotics	1H30	0H	1H	2H30	2H30	3	3	40% Lab	60%
Advanced and Current Trends in Artificial Intelligence,	1H30	0H	1H	2H30	1H	2	2	40% Lab	60%
Robot Operating System & Computer Tools for Robotics	1H30	0H	1H30	3H	0H30	2	2	40% Lab	60%
Cross-disciplinary Teaching Unit (UET)									
UET2.1.1	**1H30**	**0H**	**0H**	**1H30**	**0H30**	**1**	**1**		
Marketing and Economy	1H30	0H	0H	1H30	0H30	1	1		100%
Discovery Teaching Unit (UED)									
UED2.1.1	**1H**	**0H**	**1H30**	**2H30**	**1H00**	**2**	**2**		
Sensors for Autonomous Systems	1H	0H	0H	1H	0H30	1	1		100%
Capstone Project II	0H	0H	1H30	1H30	0H30	1	1	100% Lab	
Total Semestre S3	**14H30**	**07H30**	**09H00**	**31H00**	**18h30**	**30**	**30**		

Semester 4

Unité d'Enseignement UE	Semester hourly volume (15 weeks / Semester)					Coefficients	Crédits	Assessment Method	
	Lectures (H)	Tutorials (H)	Laboratory Work (H)	Total (H)	Personal Work			Tutorial/Lab work	Final Exam
Core Teaching Unit (UEF)									
UEF2.2.1	**4H30**	**1H30**	**3H30**	**9H30**	**6H**	**10**	**10**		
Industrial Automation	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Industrial Robots Control	1H30	0H	1H	2H30	1H	2	2	40% Lab	60%
Internet Of Things (IOT)	1H30	0H	1H30	3H	2H30	4	4	40% Lab	60%
UEF2.2.2	**3H**	**1H30**	**3H**	**7H30**	**5H**	**8**	**8**		
UAV Flight Testing & Operations	1H30	1H30	1H30	4H30	2H	4	4	(20% Tut + 20% Lab)	60%
Introduction to Materials & Mechanical Manufacturing Processes	1H30	0H	1H30	3H	3H	4	4	40% Lab	60%
Methodology Teaching Unit (UEM)									
UEM2.2.1	**3H**	**1H30**	**1H**	**5H30**	**3H**	**5**	**5**		
Advanced Embeded Systems	1H30	0H	1H	2H30	1H	2	2	40% Lab	60%
Machine Dynamics	1H30	1H30	0H	3H	2H	3	3	40% Tut	60%
UEM2.2.2	**1H30**	**0H**	**1H30**	**3H**	**2H**	**3**	**3**		
Computational intelligence for Optimization	1H30	0H	1H30	3	2H	3	3	40% Lab	60%
Cross-disciplinary Teaching Unit (UET)									
UET2.2.1	**1H30**	**0H**	**0H**	**1H30**	**0H30**	**1**	**1**		
Project management	1H30	0H	0H	1H30	0H30	1	1		100%
Discovery Teaching Unit (UED)									
UED2.2.1	**3H**	**0H**	**0H**	**3H**	**2H30**	**3**	**3**		
Reverse Engineering and Rapid Prototyping	1H30	0H	0H	1H30	0H30	1	1		100%
Introduction to Industry 4.0 and Smart Manufacturing	1H30	0H	0H	1H30	0H30	1	1		100%
Training Internship II	/	/	/	/	1H30	1	1	100%	
Total Semestre S4	**16H30**	**04H30**	**09H00**	**30H00**	**19H00**	**30**	**30**		

Semester 5

Unité d'Enseignement UE	Volume horaire semestriel (15 semaines / semestre)					Coefficients	Crédits	Assessment Method	
	Lectures (H)	Tutorials (H)	Laboratory Work (H)	Total (H)	Personal work			Tutorial/Lab work	Final Exam
Core Teaching Unit (UEF)									
UEF3.1.1	**3H**	**3H**	**2H**	**8H**	**5**	**8**	**8**		
Advanced Machine Learning for Robotics	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Distributed Systems and Agent-Based Technologies	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
UEF3.1.2	**4H30**	**1H30**	**3H30**	**9H30**	**6H30**	**10**	**10**		
Human-machine, machine-machine Interaction/Collaboration	1H30	0H	1H	2H30	2H30	3	3	40% Lab	60%
Computer vision for robotics	1H30	1H30	1H	4H	2h30	4	4	(20% Tut + 20% Lab)	60%
Real-Time Operating Systems	1H30	0H	1H30	3H	1H30	3	3	40% Lab	60%
Methodology Teaching Unit (UEM)									
UEM3.1.1	**3H**	**1H30**	**2H**	**6H30**	**5**	**7**	**7**		
Marine Robotics	1H30	1H30	1H	4H	2H30	4	4	(20% Tut + 20% Lab)	60%
Data fusion	1H30	0H	1H	2H30	2H30	3	3	40% Lab	60%
Cross-disciplinary Teaching Unit (UET)									
UET3.1.1	**3H**	**0H**	**0H**	**3H**	**1H**	**2**	**2**		
Entrepreneurship & Startup Development	1H30	0H	0H	1H30	0H30	1	1		100%
Quality Control	1H30	0H	0H	1H30	0H30	1	1		100%
Discovery Teaching Unit (UED)									
UED3.1.1	**3H**	**0H**	**0H**	**3H**	**2H30**	**3**	**3**		
Service Robotics	1H30	0H	0H	1H30	0H30	1	1		100%
Cybersecurity	1H30	0H	0H	1H30	0H30	1	1		100%
Training Internship III	/	/	/	/	1H30	1	1	100%	
Total Semestre S5	**16H30**	**08H30**	**12H00**	**30H00**	**20H00**	**30**	**30**		

Semester 6

Internships in companies or research laboratories culminating in a thesis and an oral defense.

		Semester Hourly Volume	Coefficients
Internship and Final Year Project	14 Weeks	420 hours	30
Total	14 Weeks	420 hours	30

Detailed Syllabus by Subject

Detailed program by course subject

Semester 1

Course Subject: Introduction to System Dynamics and Control, **Code:** ISDC
Course Subject: Digital Signal Processing, **Code:** DSP
Course Subject: Mathematics for Engineers, **Code:** MAE
Course Subject: Advanced Analog Electronics, **Code:** AAE
Course Subject: Introduction to Electrical Engineering, **Code:** IEE
Course Subject: Computational Modeling & Simulation for Engineering Design, **Code:** CMSED
Course Subject: Advanced Technical Design, **Code:** ATD
Course Subject: Sensors and Instrumentation, **Code:** SNI
Course Subject: Introduction to Entrepreneurship, **Code:** INE
Course Subject: Basic Principles of Robotics, **Code:** BPR

Semester 2

Course Subject: Introduction to Machine Learning, **Code:** IML
Course Subject: Optimization Techniques, **Code:** OPT
Course Subject: Microprocessors and microcontrollers, **Code:** MPMC
Course Subject: Digital Control Systems, **Code:** DCS
Course Subject: Mechanical systems, **Code:** IMS
Course Subject: Computer Aided Design, **Code:** CAD
Course Subject: Engineering Programming, **Code:** ENGP
Course Subject: Electric Actuators for Robotics, **Code:** EAR
Course Subject: Introduction to UAVs Operations and Regulations , **Code:** UAVOR
Course Subject: Capstone Project I, **Code:** CAP-I
Course Subject: Linux Operating System, **Code:** LOS
Course Subject: Training Internship I, **Code:** TI-I

Semester 3

Course Subject: Nonlinear and Optimal Control Design, **Code:** NLOCD
Course Subject: Multi-Physics Modeling of Technological Systems, **Code:** MPMTS
Course Subject: Robot Motion Planning, **Code:** RMP
Course Subject: Kinematics, Dynamics and Control of Robot Manipulators, **Code:** KDCRM
Course Subject: Power Electronics, **Code:** POWE
Course Subject: Mobile Robotics, **Code:** MOR
Course Subject: Advanced and Current Trends in Artificial Intelligence, **Code:** ACTAI
Course Subject: Robot Operating System & Computer Tools for Robotics, **Code:** ROS-CTR
Course Subject: Marketing and Economy, **Code:** MARE
Course Subject: Sensors for Autonomous Systems, **Code:** SNAS
Course Subject: Capstone Project II, **Code:** CAP-II

Semester 4

Course Subject: Industrial Automation, **Code:** IAUT
Course Subject: Industrial Robots Control, **Code:** IRC
Course Subject: Internet Of Things (IOT), **Code:** IOT
Course Subject: UAV Flight Testing & Operations, **Code:** UAVFTO
Course Subject: Introduction to Materials & Mechanical Manufacturing Processes, **Code:** IMMMP
Course Subject: Advanced Embeded Systems **Code:** AES
Course Subject: Machine Dynamics, **Code:** MADY
Course Subject: Computational intelligence for Optimization, **Code:** CIO
Course Subject: Project management, **Code:** PROM
Course Subject: Reverse Engineering and Rapid Prototyping, **Code:** RERP
Course Subject: Introduction to Industry 4.0 and Smart Manufacturing, **Code:** IISM
Course Subject: Training Internship II, **Code:** TI-II

Semester 5

Course Subject: Advanced Machine Learning for Robotics, **Code:** AMLR
Course Subject: Distributed Systems and Agent-Based Technologies, **Code:** DSABT
Course Subject: Human-Machine, Machine-Machine Interaction/Collaboration, **Code:** MMI
Course Subject: Computer Vision for Robotics, **Code:** CVR
Course Subject: Real-Time Operating Systems, **Code:** RTOS
Course Subject: Marine Robotics, **Code:** MROB
Course Subject: Data fusion, **Code:** DFU
Course Subject: Entrepreneurship & Startup Development, **Code:** ESTART
Course Subject: Quality Control, **Code:** QCN
Course Subject: Service robotics, **Code:** SROB
Course Subject: Cybersecurity, **Code:** CYBER
Course Subject: Training Internship III, **Code:** TI-III

Detailed programs for the fifth semester

Teaching Unit	Subject Title	Code	Semester
UEF1.1.1	**Introduction to System Dynamics and Control**	**ISDC**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:

1. Understand the fundamental principles of linear control systems.
2. Analyze the stability, controllability, and observability of linear systems.
3. Design controllers (e.g., PID, state feedback) for linear systems.
4. Apply control theory to real-world engineering problems.

Recommended Prerequisites:

Course Outline

Chapter 1: Introduction to Control Systems (1 week)

- Definition of control systems
- Open-loop vs. closed-loop systems
- Examples of control systems in engineering

Chapter 2: Mathematical Modeling of Systems (3 weeks)

- Differential equations
- Laplace Transform and its inverse
- Transfer functions
- State-space representation
- Linearization of nonlinear systems
- Examples: Mechanical, electrical, and electromechanical systems

Chapter 3: Time-Domain Analysis (2 weeks)

- Transient and steady-state response
- First-order and second-order systems
- Performance metrics: Rise time, settling time, overshoot, and steady-state error

Chapter 4: Stability Analysis (2 week)

- Concept of stability
- Routh-Hurwitz criterion
- Root locus method

Chapter 5: Frequency-Domain Analysis (2 weeks)

- Bode plots
- Nyquist plots
- Gain and phase margins

Chapter 6: Controller Design (2 weeks)

- Proportional-Integral-Derivative (PID) control
- Ziegler-Nichols tuning method
- Lead-lag compensators

Chapter 7: State-Space Representation (3 weeks)

- Controllability and observability
- State feedback control
- Pole placement

Laboratory work:

- **Lab 1:** Time-domaine analysis (step/impulse responses) in MATLAB.
- **Lab 2:** Frequency- domaine analysis (Bode/Nyquist diagrams) in MATLAB.
- **Lab 3:** PID controller design and tuning in MATLAB/Simulink.
- **Lab 4:** State-space modeling and control in MATLAB.

Assessment Methods:
- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources

- **Textbooks:**
  1. **"Modern Control Systems"** by Richard C. Dorf and Robert H. Bishop.
  2. **"Modern Control Engineering"** by Katsuhiko Ogata.
  3. **"Automatic Control Systems"** by Benjamin C. Kuo and Farid Golnaraghi.
  4. **"Linear System Theory and Design"** by Chi-Tsong Chen.
  5. **"Feedback Control of Dynamic Systems"** by Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini.
  6. Online resources: MIT OpenCourseWare, Khan Academy, and YouTube tutorials.
- **On-line Resources:**
  1. **MIT OpenCourseWare (OCW) - Linear Control Systems**
     - Link: https://ocw.mit.edu
  2. **Control Tutorials for MATLAB and Simulink**
     - Link: https://ctms.engin.umich.edu

Teaching Unit	Subject Title	Code	Semester
UEF1.1.1	**Digital Signal Processing**	**DSP**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:

The objective of this course is to present tools to analyze the properties of a signal and examine what happens to it when it passes through a system. The last part concerns the acquisition of the concepts and notions of time-frequency analysis of non-stationary signals.

Targeted Skills

- Understanding and applying Fourier and Hilbert transforms.
- Designing and analyzing FIR and IIR filters.
- Performing correlation and convolution operations.
- Analyzing stochastic signals and their statistical properties.
- Applying time-frequency analysis techniques to real-world signals.

Recommended prerequisites

- Basic Mathematics : fonctions, integrals, etc.
- Probability and Statistics
- Basic programming skills (MATLAB/Python)

Course Outline :

Chapter 1: Introduction (1 week)

1. Signal Definition
2. Signal processing and its applications

Chapter 2: Deterministic signals (2 weeks)

1. Continuous Time and Frequency Signals (Fourier Transform)
2. Discrete-Time and Continuous Frequency Signals (Sampling Theorem)
3. Discrete Time and Frequency Signals (Discrete Fourier Transform 'DFT', FFT)

Chapter 3: Signals and systems (2 weeks)

1. Z Transform
2. Hilbert transform
3. Linear and stationary systems

Chapter 4: Synthesis of digital filters (2 week)

1. FIR filter design methods
2. IIR filter design methods

Chapter 5: Random signals (2 weeks)

1. Random Process
2. Stationary Signals

3. Ergodic Signals

Chapter 6: Introduction to Estimation and Spectral Analysis (3 week)

1. Estimation elements: Bias, variance.
2. Correlation estimators
3. Periodogram
4. Welch
5. AR Modeling: Levinson's Algorithm

Chapter 7: Introduction to Time-Frequency Analysis (3 week)

1. Concept of time-frequency representation
2. Short-Time Fourier Transform (STFT), Wavelet Transform, Time-frequency trade-offs
3. Instantaneous Frequency, Spectrogram, Wigner-Ville Distribution
4. Quadratic Time-Frequency Distributions
5. Cross-term reduction
6. Application examples

Laboratory work

Lab 1: MATLAB/Octave Initiation/Signal Generation
Lab 2: Sampling, DFT/FFT
Lab 3: Digital Filtering
Lab 4: Spectral estimation
Lab 5: Time-frequency analysis

Assessment Methods:

- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended textbooks and resources:

1. Proakis, J.G. &Manolakis, D.G. "Digital Signal Processing: Principles, Algorithms, and Applications." Pearson, 2007.
2. Oppenheim, A.V. &Schafer, R.W. "Discrete-Time Signal Processing." Prentice Hall, 2010.
3. Hayes, M.H. "Statistical Digital Signal Processing and Modeling." John Wiley& Sons, 1996.
4. Cohen, L. "Time-Frequency Analysis." Prentice Hall, 1995.
5. Mitra, S.K. "Digital Signal Processing: A Computer-Based Approach." McGraw-Hill, 2010.

Teaching Unit	Subject Title	Code	Semester
UEF1.1.1	**Mathematics for Engineers**	**MAE**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:

1. **Master Advanced ODEs and Systems:**
   Solve higher-order linear ODEs, analyze systems of ODEs using Wronskian, phase plane methods, and Laplace transforms, and apply these to engineering models.
2. **Explore Series Solutions and Special Functions:**
   Use power series, Frobenius methods, and special functions (Legendre, Bessel, hypergeometric) to solve ODEs and understand their engineering applications.
3. **Solve PDEs Using Analytical and Transform Methods:**
   Apply separation of variables, D'Alembert's solution, and integral transforms (Fourier and Laplace) to solve partial differential equations (PDEs).
4. **Understand Sturm-Liouville Problems and Orthogonal Functions:**
   Solve Sturm-Liouville problems, classify orthogonal functions, and apply them in engineering contexts.
5. **Apply Numerical Methods for ODEs and PDEs:**
   Use multistep methods, finite differences, and time-scale analysis to numerically solve ODEs and PDEs, including elliptic, parabolic, and hyperbolic types.
6. **Model and Solve Real-World Engineering Problems:**
   Develop the ability to model, analyze, and solve complex engineering systems using advanced mathematical tools and techniques.

By the end of the course, students will be equipped with the mathematical skills necessary to tackle advanced engineering challenges.

Course outline:

Chapter 1: Special Functions and ODEs (5 weeks)

1. Higher Order Linear ODEs.
2. Systems of ODEs:
   - Basic Theory: Wronskian, Phase Plane Method, Critical Points, Qualitative Methods, Laplace Transform for Sytems.
   - Conversion of an nth-Order ODE to a System.
   - Models in Engineering.
3. Series Solutions of ODEs and Special Functions.
   - Power Series Method, Frobenius Method
   - Special Equations and Functions: Legendre, Bessel, Hypergeometric …
4. Fourier Analysis, and Polynomial and Trigonometric Approximations.
5. Sturm-Liouville Problem:
   - Orthogonal Functions
   - Classification of Orthogonal Functions: Recurrence, Differential Equations, Generating functions.
6. Applications in Engineering.

Chapter 2: Explicit Solution of PDEs. (5 weeks)

1. Basic Concepts of PDEs.
2. Solving PDEs:
    - Direct Methods, Working Procedures: Particular Integral, Complimentary Function.
    - Solution by Separating Variables.
    - D'Alembert's Solution of the Wave Equation.
3. Integral Transforms:
    - Solution of PDEs using Fourier Integrals and Transforms.
    - Solution of PDEs using Laplace Transforms.
4. Polar, Cylindrical and Spherical Coordinates: Orthogonal Series.

Chapter 3: Time-Scale and Numerical Solutions of ODEs and PDEs (4 weeks)

1.Time-Scale Analysis:
    - Discrete-Time and q-Analogue Version.
    - Some Equations, Functions and Transforms.
2. Multistep Method, Finite Difference, Casorati Determinant.
3. Methods for Systems and Higher order ODEs.
4. Methods for (Elliptic, Parabolic, Hyperbolic) PDEs.
5. Application: PDEs of Engineering.

Chapter 4: (4 weeks)

Laboratory Work:

These labs will reinforce theoretical concepts, develop computational skills, and provide practical insights into solving real-world engineering problems using advanced mathematics.

Lab 1: Solving Systems of ODEs and Phase Plane Analysis

- **Objective:** Analyze and solve systems of ordinary differential equations (ODEs) and visualize their behavior using phase plane methods.
- **Tasks:**
    1. Solve a system of linear ODEs (e.g., a mechanical or electrical system) using matrix methods.
    2. Compute the Wronskian to check for linear independence of solutions.
    3. Plot phase portraits and identify critical points (e.g., nodes, saddles, spirals).
    4. Use Laplace transforms to solve a system of ODEs and interpret the results.

Lab 2: Series Solutions and Special Functions

- **Objective:** Use power series and special functions to solve ODEs and analyze their properties.
- **Tasks:**
    1. Solve a second-order ODE using the power series method and Frobenius method.
    2. Generate and plot Legendre polynomials and Bessel functions.
    3. Explore the orthogonality properties of these functions and their applications in engineering problems (e.g., heat conduction, vibration analysis).

Lab 3: Fourier Analysis and Approximations

- **Objective:** Perform Fourier analysis to approximate functions and solve engineering problems.
- **Tasks:**
    1. Compute Fourier series for periodic functions (e.g., square waves, sawtooth waves).
    2. Analyze the convergence of Fourier series and visualize the approximations.

3. Apply Fourier transforms to solve a heat conduction or wave propagation problem.

Lab 4: Solving PDEs Using Analytical and Numerical Methods

- **Objective:** Solve partial differential equations (PDEs) using separation of variables and numerical techniques.
- **Tasks:**
    1. Solve the heat equation or wave equation using separation of variables.
    2. Implement finite difference methods to solve a parabolic PDE (e.g., heat equation) or hyperbolic PDE (e.g., wave equation).
    3. Visualize the solutions and compare analytical and numerical results.

Lab 5: Time-Scale Analysis and Numerical Solutions of ODEs

- **Objective:** Explore time-scale analysis and numerical methods for solving ODEs and PDEs.
- **Tasks:**
    1. Implement a multistep method (e.g., Runge-Kutta) to solve a higher-order ODE.
    2. Analyze a discrete-time system using time-scale calculus.
    3. Solve an engineering problem (e.g., damped oscillator, electrical circuit) numerically and interpret the results.

Tools and Software:

- MATLAB/Python/Mathematica: For numerical computations, plotting, and simulations.
- Engineering Applications: Problems can be drawn from mechanical vibrations, heat transfer, fluid dynamics, electrical circuits, etc.

Assessment Methods:

- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources

1. C. Chicone, Invitation to Applied Mathematics: Differential Equations, Modeling, and Compuation, Academic Press publications, (2017).
2. H. F. Davis, *Fourier Series and Orthogonal Functions,* Dover Publivations, Inc., (1st Edition 1989).
3. D. G. Duffy, *Advanced Engineering Mathematics*, CRC Press, Taylor & Francis Group, (4th 2017).
4. U. Graf, *Applied Laplace Transforms and z-Transforms for Scientists and Engineers,* Springer Basel AG, (1st Edition 2004).
5. G. A. Korn, T. M. Korn, *Mathematical Handbook for Scientists and Engineers*, Dover Publivations, Inc., (2000).
6. W. R. LePage, *Complex Variables and the Laplace Transforms for Engineers,* Dover Publivations, Inc., (1961).

Teaching Unit	Subject Title	Code	Semester
UEF1.1.2	**Advanced Analog Electronics**	**AAE**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:
1. To understand the fundamental principles of analog electronic circuits.
2. To analyze and design basic analog circuits using diodes, transistors, and operational amplifiers.
3. To develop practical skills in building and testing analog circuits.
4. To apply analog electronics concepts to real-world applications.

Learning Outcomes:
By the end of the module, students will be able to:
1. Analyze and design basic analog electronic circuits.
2. Understand the operation and applications of diodes, transistors, and op-amps.
3. Build and test analog circuits in a laboratory setting.
4. Apply analog electronics principles to solve real-world engineering problems.

Recommended prerequisites

Module Outline:
Chapter 1: Introduction to Analog Electronics (1 Week)
- Basic concepts: voltage, current, resistance, power
- Introduction to circuit components: resistors, capacitors, inductors
- Ohm's Law, Kirchhoff's Laws, and basic circuit analysis techniques

Chapter 2: Semiconductor Fundamentals (2 Weeks)
- Introduction to semiconductors
- Intrinsic and extrinsic semiconductors
- PN junction and its characteristics

Chapter 3: Diode Circuits (3 Weeks)
- Diode operation and models (ideal, practical, and small-signal)
- Applications of diodes: rectifiers, clippers, clampers
- Half-wave and full-wave rectifiers
- Bridge rectifier
- Zener diodes and voltage regulation
- LED and photodiode applications

Chapter 4: Bipolar Junction Transistors (BJTs) (3 Weeks)
- BJT structure and operation (NPN and PNP)
- BJT configurations: common emitter, common base, common collector
- DC biasing of BJTs
- Small-signal analysis and hybrid-π model
- BJT amplifiers: single-stage and multi-stage

Chapter 5: Field-Effect Transistors (FETs) (1 Week)
- Introduction to FETs: JFETs and MOSFETs
- FET characteristics and operation
- FET biasing techniques
- FET amplifiers: common source, common gate, common drain

Chapter 6: Operational Amplifiers (Op-Amps) (2 Weeks)
- Introduction to op-amps: ideal and practical characteristics
- Op-amp configurations: inverting, non-inverting, differential
- Applications of op-amps: summing amplifier, integrator, differentiator
- Active filters: low-pass, high-pass, band-pass, and band-stop
- Comparators and waveform generators

Chapter 7: Feedback and Oscillators (2 Weeks)
- Concept of feedback: positive and negative feedback
- Effects of feedback on gain, bandwidth, and impedance
- Oscillator principles: Barkhausen criterion
- Types of oscillators: RC, LC, and crystal oscillators

Chapter 8: Power Amplifiers and Voltage Regulators (1 Week)
- Class A, Class B, Class AB, and Class C power amplifiers
- Efficiency and distortion in power amplifiers
- Voltage regulators: linear and switching regulators
- Heat dissipation and thermal management

CHAPTER9: Multivibrators (2.5 weeks)
- Introduction to Multivibrators (Definition and classification).
- A stable Multivibrator (Operating Principle, Transistor-Based Astable Circuit, Oscillation Frequency Calculation, Astable Multivibrator with Integrated Circuit (555)).
- Monostable Multivibrator (Operating Principle, Transistor Monostable Circuit, Pulse Duration Calculation, Monostable with NE555).
- Bistable Multivibrator Applications (Operating Principle, Types of Bistables).
- Practical application.

Laboratory Work:

LW 1: Diode Circuits
- Half-wave and full-wave rectifiers
- Zener diode voltage regulator

LW 2: Transistor Circuits
- BJT biasing and amplifier circuits
- FET-based amplifier design

LW 3: Op-Amp Circuits
- Inverting and non-inverting amplifiers
- Active filters and waveform generators

LW 4: Oscillators and Power Amplifiers
- RC oscillator design
- Class B push-pull amplifier

Assessment Methods:
- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources
1. "Electronic Devices and Circuit Theory" by Robert L. Boylestad and Louis Nashelsky.
2. "Microelectronic Circuits" by Adel S. Sedra and Kenneth C. Smith.
3. "Operational Amplifiers and Linear Integrated Circuits" by Robert F. Coughlin and Frederick F. Driscoll.
4. Online resources: MIT OpenCourseWare, All About Circuits, and Khan Academy.

Teaching Unit	Subject Title	Code	Semester
UEF1.1.2	**Introduction to Electrical Engineering**	**IEE**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:

By the end of this module, students will:

1. **Understand** the fundamental laws of electricity and magnetism, with a focus on their practical applications in electrotechnics.
2. **Analyze** and design simple electrical circuits for power distribution, energy conversion, and control.
3. **Operate** and characterize common electrical machines (motors, generators, transformers) under different load conditions.
4. **Evaluate** energy efficiency and apply safety standards in designing and troubleshooting electrical systems.
5. **Gain** practical experience through laboratory experiments in measurement, circuit assembly, and fault diagnosis.

Recommended Prerequisites:

- Basic knowledge in Mathematics & Physics: Algebra, basic calculus, and introductory concepts of electricity (Ohm's law, simple circuit analysis).
- Introductory Electronics: Familiarity with electronic components (resistors, capacitors, inductors) is helpful.

Course Outline:

Chapter 1: Introduction to Electrotechnics (1 weeks)

- Evolution of electrical engineering
- Overview of applications (power systems, electronics, robotics)
- Fundamental constants and units

Chapter 2: Basic Electrical Concepts (1.5 weeks)

- Voltage, current, resistance (Ohm's law)
- Power and energy calculations (P=VI, etc.)
- AC vs. DC, single-phase vs. three-phase
- Basic waveforms and phasor representation

Chapter 3: Magnetism & Electromagnetic Induction (3 weeks)

- Magnetic fields and flux
- Faraday's law, Lenz's law
- Practical induction devices (coils, solenoids)
- Inductive reactance and transformers preview

Chapter 4: Electrical Components & Circuits (1.5 weeks)

- Resistors, capacitors, inductors
- Series, parallel, and resonance circuits
- Impedance, reactive power, power factor

Chapter 5: Transformers & Power Distribution (2 weeks)

- Transformer operation, turns ratio
- Power distribution (transmission lines, high voltage)

- Efficiency and losses (copper loss, core loss)
- Three-phase systems basics

Chapter 6: Electric Machines (4.5 weeks)

- DC motors (shunt, series, compound)
- Induction motors (single-phase, three-phase)
- Synchronous machines (generators, motors)
- Control and speed regulation.

Chapter 7: Electric Machines (1.5 weeks)

- Safety standards (IEC, IEEE, local codes)
- Protection devices (fuses, circuit breakers, RCDs)
- Grounding/earthing, insulation, PPE
- Energy-saving measures (power factor correction, improved design)

Laboratory Work:

- **Lab 1:** Basic Measurements
- **Lab 2:** Induction Experiment
- **Lab 3:** RLC Circuit Analysis
- **Lab 4:** Transformer Testing
- **Lab 5:** DC Motor Performance
- **Lab 6:** AC Motor Performance

Assessment Methods:

- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources:

- **Hughes, E.** *Electrical and Electronic Technology*. Pearson Education.
- **Wildi, T.** *Electrical Machines, Drives, and Power Systems*. Pearson Education.
- **Fitzgerald, A.E., Kingsley, C., & Umans, S.D.** *Electric Machinery*. McGraw-Hill.
- **IEEE/IEC** standards for safety, grounding, and power quality.
- **Roth, A.** *The Circuit Designer's Companion*. Newnes (for practical design tips).

Teaching Unit	Subject Title	Code	Semester
UEM1.1.1	**Computational Modeling & Simulation for Engineering Design**	**CMSED**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h30	**3/3**

Course Objectives:

- Understand the fundamental principles of the Finite Element Method (FEM).
- Formulate and solve engineering problems using FEM.
- Implement FEM techniques in robotics and autonomous systems.
- Use commercial FEM software for structural, thermal, and dynamic analysis.
- Develop numerical models and validate results through case studies.

Recommended Prerequisites:

- Engineering Mathematics (Linear Algebra, Differential Equations)
- Mechanics of Materials
- Numerical Methods
- Programming

Course Outline

Chapter 1 : Introduction to FEM and its Mathematical Foundations (2 weeks)

- Overview of FEM and its applications in robotics.
- Comparison with analytical and other numerical methods.
- Discretization and element types.
- Approximation techniques and interpolation functions.
- Weighted residual methods and Galerkin's method.

Chapter 2: 1D Finite Elements & Structural Applications (2 weeks)

- Bar and truss elements.
- Stiffness matrix derivation.

Chapter 3: FEM for Beam and Frame Structures (3 weeks)

- Beam element formulation (Euler-Bernoulli and Timoshenko).
- Applications in robotic arms and mechanical linkages.

Chapter 4: 2D Elements: Triangular and Quadrilateral Elements (3 weeks)

- Plane stress and plane strain conditions.
- Isoparametric formulation.

Chapter 5: 3D Elements & Applications in Robotics (2 weeks)

- 3D element formulation and applications.

- Tetrahedral and hexahedral elements.

Chapter 6: Dynamic Analysis Using FEM (2 weeks)

- Equations of motion for FEM models.
- Modal and transient dynamic analysis.

Laboratory work:
Lab 1: Introduction to an FEM software.
Lab 2: Implementation of 1D FEM.
Lab 3: Beam analysis using FEM software.
Lab 4: 2D meshing techniques and analysis.
Lab 5: 3D structural analysis of robotic components.
Lab 6: Vibration analysis of robotic components

Assessment Methods:
- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources

1. Zienkiewicz, O.C., Taylor, R.L. (2000). *The Finite Element Method. Volume 1: The basis*. MPG books Ltd
2. Bathe, K.J. (1996). *Finite Element Procedures*. Prentice Hall Inc
3. Rao, S.S. (2004). The Finite Element Method in Engineering. Elsevier Science & Technology Books

Teaching Unit	Subject Title	Code	Semester
UEM1.1.1	**Advanced Technical Design**	**ATD**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h30	**3/3**

Course Objectives:
This course complements the technical drawing course and will enable students to acquire the principles of standardized representation of mechanical parts, also known as industrial drawing. Furthermore, this subject will allow students to represent and interpret plans of mechanisms and machines. It also aims to enhance students' graphic imagination so that they can better master the universal language of communication among technicians and prepare for the proper use of CAD tools.

Learning Outcomes:
By the end of this course, students will be able to:

- Understand the role of standardization
- Read an assembly drawing and understand its operation
- Understand different mechanical connections and create the kinematic diagram of a mechanism
- Find mechanical solutions for different connections
- Effectively apply CAD tools

Course outline:
Chapter 1: Basic Mechanical Functions (3 weeks)

- Principle of isostatic
- Mechanical connections (elementary connections, connection characteristics, connection modes, realization of connections)
- Centring and orientation functions (rotational guidance, translational guidance, functional dimensioning, fits, technical specifications, and symbolization).

Chapter 2: Drawing Interpretation and Kinematic Diagrams (3 weeks)

- Functional dimensioning, fits, kinematic diagrams, assembly drawing, definition drawing, exploded view representation

Chapter 3: Assembly Analysis (5 weeks)

- Study of articulated connections, bearing assemblies, thrust bearings, plain bearings, obstacles, gears, sealing, and lubrication function

Chapter 4: Application – CAD of a Mechanical System (4 weeks)

- Creation of various parts and assemblies, including the use of element libraries (bearings, screws, etc.)
- Drawing preparation (tolerances, functional clearances, fits, etc.)
- Use of SolidWorks for practical work

Assessment Methods:
- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources

1. Chevalier A. "Guide du dessinateur industriel", Editions Hachette Technique,
2. French, M. J. (1999). "Conceptual Design for Engineers". Springer.
3. Giesecke, F. E., Mitchell, A., Spencer, H. C., & Hill, I. L. (2011). "Technical Drawing with Engineering Graphics". Pearson.
4. Lamit, L. B. (2002). "Engineering Design Graphics". Prentice Hall.
5. Bertoline, G. R., Wiebe, E. N., Hartman, N., & Ross, W. A. (2018). "Fundamentals of Graphics Communication". McGraw-Hill.
6. Zeid, I. (2014). "Mastering CAD/CAM". McGraw-Hill Education.

Teaching Unit	Subject Title	Code	Semester
UEM1.1.2	**Sensors and Instrumentation**	**SNI**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h0	**2/2**

Course Objectives:
The objective of this course is to provide engineering students with a practical introduction to the fundamental technologies of sensors and their real-world applications. Through the study of various types of sensors, such as capacitive, piezoelectric, inductive, ultrasonic, accelerometers, and image sensors, students will explore their operating principles and their applications. Instrumentation techniques will be covered, incorporating key concepts such as program control, sampling, data collection, and analysis, with practical examples to illustrate their use.
Weekly laboratory sessions will give students hands-on experience in working with different sensors, enabling a deeper understanding of physical phenomena and measurement techniques.

Targeted Skills
- Selection and integration of sensors for various applications.
- Implementation of sensor calibration and interfacing techniques.
- Signal processing and noise filtering in sensor systems.
- Practical understanding of sensors used in autonomous systems.
- Ability to design and analyze sensor-based measurement systems.

Recommended prerequisites
- Basic electronics and circuit theory.
- Fundamentals of physics (mechanics, fluid mechanics, optics, and electromagnetism).
- Basics of signal processing and data acquisition.
- Introduction to programming.

Module outline:
Chapter 1: Introduction to Sensor Technologies and their application (2 weeks)
- Definition and importance of sensors
- Classification of sensors

Chapter 2: Sensor Types and their Application in different fields (6 weeks)
2.1 Optical Encoders
- Incremental Encoders
- Absolute Encoders
- Calibration

2.2 Accelerometers and Gyroscopes
- Working principles (MEMS, piezoelectric, capacitive sensors)
- Applications in motion tracking and robotics
- Calibration and error sources

2.3 GNSS , LiDAR and Radar

- GNSS definitions and types
- GPS fundamentals, accuracy, and error correction
- LiDAR working principles and applications
- Comparison of GPS and LiDAR for positioning
- RADAR working principles and applications

2.4 Environmental Sensors

- Temperature sensors (Thermocouples, RTDs, Thermistors)
- Pressure sensors (Barometric, Absolute, Gauge)
- Humidity sensors and their applications

2.5 Motion and Navigation Sensors

- IMUs and their role in UAVs and ground robots
- Altitude sensors (Barometric, Ultrasonic, LiDAR-based)
- Applications of navigation sensors in autonomous vehicles

2.6 Proximity and Obstacle Detection Sensors

- Ultrasonic sensors
- Infrared sensors
- LiDAR and Radar in obstacle detection

Chapter 3: Sensor Interfacing and Calibration (3 weeks)

3.1 Basics of Sensor Interfacing

- Analog vs. digital sensors
- Communication protocols (I2C, SPI, UART, CAN)
- Data acquisition and synchronization

3.2 Sensor Calibration Techniques

- Importance of calibration
- Methods (Static, Dynamic, Multi-point Calibration)
- Software tools for calibration

3.3 Error Sources and Compensation

- Sources of errors (Drift, Bias, Noise, Temperature effects)
- Error correction techniques (Filtering, Sensor fusion)

Chapter 4: Noise Filtering and Signal Conditioning (2 weeks)

3.1 Introduction to Signal Conditioning

- Signal amplification and attenuation
- Filtering techniques (Low-pass, High-pass, Band-pass, Notch filters)
- Signal isolation and impedance matching

3.2 Noise Sources and Reduction Methods

- Types of noise (thermal, shot, flicker, EM interference)
- Shielding and grounding techniques
- Software-based noise reduction

Laboratory Work

LW 1: Temperature sensors and Data Acquisition
LW 2: Accelerometer and magnetometer sensors
LW 3: Ultrasonic sensors.
LW 4: Capacitive sensors.
LW 5: Pressure sensors and Error compensation

Assessment Methods:

- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended textbooks and resources

1. Fraden, J., *Handbook of Modern Sensors: Physics, Designs, and Applications*. Springer, 2016.
2. Webster, J. G., *The Measurement, Instrumentation, and Sensors Handbook*. CRC Press, 2000.
3. Wilson, J., *Sensor Technology Handbook*. Elsevier, 2004.
4. Groves, P. D., *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.
5. Carr, J. J., *Sensors and Circuits: Sensors, Transducers, Signal Conditioning, and Wireless Sensors*. Newnes, 2017.

Teaching Unit	Subject Title	Code	Semester
UET1.1.1	**Introduction to Entrepreneurship**	**INE**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30			**1/1**

Course Objectives:

1. To introduce students to the fundamentals of entrepreneurship and its role in driving innovation in renewable energy and sustainable development.
2. To develop critical thinking and problem-solving skills for identifying market opportunities and creating viable business models.
3. To familiarize students with the process of starting and managing a business, including legal, financial, and operational aspects.
4. To encourage students to think creatively and develop entrepreneurial mindsets that align with sustainable development goals (SDGs).
5. To provide hands-on experience in developing and pitching a business idea related to renewable energy or sustainability.

Course Outline:

Chapter 1: Introduction to Entrepreneurship (1 week)

- Definition and importance of entrepreneurship
- Types of entrepreneurship: Social, Green, and Technology-driven
- Role of entrepreneurship in renewable energy and sustainable development
- Case studies: Successful entrepreneurs in the RASD field

Chapter 2: Entrepreneurial Mindset and Creativity (2 weeks)

- Characteristics of successful entrepreneurs
- Developing a growth mindset and resilience
- Techniques for creative problem-solving and innovation
- Brainstorming and ideation techniques for sustainable solutions

Chapter 3: Identifying Opportunities in RASD (2 weeks)

- Market research and analysis
- Identifying gaps and opportunities in renewable energy and sustainability sectors
- Understanding customer needs and pain points
- Trends and challenges in the RASD industry

Chapter 4: Business Model Development (2 weeks)

- Introduction to business models and their components
- The Business Model Canvas (BMC)
- Designing sustainable and scalable business models
- Case studies: Business models in renewable energy startups

Chapter 5: Legal and Regulatory Framework (2 weeks)

- Basics of business registration and intellectual property (IP)
- Compliance with environmental and energy regulations
- Understanding subsidies, incentives, and policies for renewable energy projects

Chapter 6: Financial Planning and Funding (2 weeks)

- Basics of financial management for startups
- Preparing a budget and financial projections
- Sources of funding: Grants, venture capital, crowdfunding, and angel investors
- Pitching to investors and stakeholders

Chapter 7: Marketing and Sales Strategies (2 weeks)

- Developing a marketing plan for RASD products/services
- Digital marketing and social media strategies
- Building a brand around sustainability
- Sales techniques and customer relationship management

Chapter 8: Operations and Project Management (1 week)

- Basics of operations management for startups
- Lean startup methodology
- Project planning and execution in renewable energy projects
- Risk management and contingency planning

Chapter 9: Pitching and Presentation Skills (1 week)

- Crafting a compelling pitch deck
- Effective communication and storytelling techniques
- Practicing elevator pitches and investor presentations
- Feedback and iteration

Assessment Methods:

- Final exam (100%).

Recommended Textbooks and Resources:

1. **"Entrepreneurship: The Practice and Mindset"** by Heidi M. Neck, Christopher P. Neck, and Emma L. Murray
2. **"Business Model Generation"** by Alexander Osterwalder and Yves Pigneur
3. **"The Lean Startup"** by Eric Ries
4. **"Sustainable Energy - Without the Hot Air"** by David J.C. MacKay
5. Online resources: Coursera, edX, and Udemy courses on entrepreneurship and renewable energy

Teaching Unit	Subject Title	Code	Semester
UED1.1.1	**Basic Principles of Robotics**	**BPR**	**S1**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30			**1/1**

Course Overview:
This course introduces students to the fundamental principles of robotics, covering the essential concepts, theories, and practices that form the foundation of robotics and autonomous systems design. The course is designed to provide students with a strong theoretical background while also emphasizing practical applications through hands-on projects and laboratory work.

Course Objectives:
By the end of this course, students will be able to:

1. Understand the basic components and architecture of robotic systems.
2. Analyze and apply kinematic and dynamic principles to robotic systems.
3. Design and implement control strategies for robotic systems.
4. Understand and apply sensor integration and data processing techniques.
5. Develop and program basic robotic systems using industry-standard tools and software.
6. Evaluate the ethical and societal implications of robotics and autonomous systems.

Recommended Prerequisites:

- **Mathematics:** Linear Algebra, Calculus, Differential Equations
- **Programming:** Basic knowledge of Python or C++
- **Physics:** Mechanics and Dynamics

Course Outline:

Chapter 1: Introduction - What is a Robot? (1 week)

- Definition and classification of robots.
- Differences between robotics and automation.

Chapter 2: General Concepts, Terminology, and Standards (1 week)

- Key terms in robotics.
- Robotics standards and regulations (ISO 10218, ISO/TS 15066, ISO 10218-1:2025,etc.).

Chapter 3: History of Robotics (1 week)

- Origins and early automated machines.
- Evolution of robotic technologies.

Chapter 4: The Global Robotics Market (1 week)

- Overview of key industry players.
- Current trends and future prospects.

Chapter 5: What benefits does the robotization bring? (1 week)

- Advantages and disadvantages of robots.
- Economic and social impact.

Chapter 6: Different Types of Robotics (1 week)

- Industrial, service, medical, military robotics, etc.
- Real-world applications.

Chapter 7: Components and Characteristics of a Robot (1 week)

- Actuators, sensors, controllers, and interfaces.
- Embedded intelligence and robot autonomy.

Chapter 8: Introduction to a Robotic Cell (1 week)

- Components of a robotic work cell.
- Integration process in production lines.

Chapter 9: Mathematical Foundations for robot modelling (1 week)

- Position and orientation of a body in space
- Coordinate transformations and homogeneous transformations

Chapter 10: Basics of Industrial Robot Programming (1 week)

- Robotics programming languages.
- Introduction to simulation environments.

Chapter 11: Safety in Robotics & Collaborative Robotics (1 week)

- Safety measures for industrial robots.
- Standards and challenges in collaborative robotics.

Chapter 12: Peri-Robotic Equipment & Robotics and Industry 4.0 (1 week)

- Advanced sensors, computer vision, and human-machine interfaces.
- Digitalization and digital twins in robotics.

Chapter 13: Case Studies (1 week)

- Presentation of real-world robotics projects.
- collaborative robots
- Analysis and discussion of challenges and solutions.

Assessment Methods:

- Final exam (100%).

Recommended Textbooks and Resources:

- **Textbooks**

1. **"Robotics: Modelling, Planning and Control"** by Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo
2. **"Introduction to Autonomous Robots"** by Nikolaus Correll, Bradley Hayes, Christoffer Heckman, and Alessandro Roncone
3. **"Robot Dynamics and Control"** by Mark W. Spong, Seth Hutchinson, and M. Vidyasagar
4. **"Probabilistic Robotics"** by Sebastian Thrun, Wolfram Burgard, and Dieter Fox

- **Online Courses:**
  - "Robotics Specialization" by University of Pennsylvania on Coursera
  - "Control of Mobile Robots" by Georgia Institute of Technology on Coursera

Detailed programs for the sixth semester

Teaching Unit	Subject Title	Code	Semester
UEF1.2.1	**Introduction to Machine Learning**	**IML**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:
1. Understand the fundamental concepts and algorithms of machine learning.
2. Learn how to apply machine learning techniques to control and robotics problems.
3. Gain hands-on experience with ML tools and frameworks.
4. Develop the ability to design and implement ML-based solutions for real-world robotics and control systems.

Course Outline:
Chapter 1: Introduction to Machine Learning (1 week)
- What is machine learning? Overview and applications
- Types of machine learning: Supervised, unsupervised, reinforcement learning
- Machine learning in robotics and Autonomous systems: Case studies

Chapter 2: Supervised Learning – Regression (3 weeks)
- Linear regression: Theory and implementation
- Support Vector Machine (SVM) Classification
- Evaluation metrics: Mean squared error (MSE), R-squared
- Decision trees and random forests
- Evaluation metrics: Accuracy, precision, recall, F1-score, ROC-AUC

Chapter 4: Unsupervised Learning (2 weeks)
- Clustering: K-means, hierarchical clustering
- Dimensionality reduction: Principal Component Analysis (PCA)

Chapter 5: Neural Networks – Basics (2 weeks)
- Biological inspiration and artificial neurons
- Perceptron, single layer and multi-layer perceptron (MLP)
- Activation functions: Sigmoid, ReLU, softmax

Chapter 8: Reinforcement Learning (RL) (2 weeks)
- Basics of reinforcement learning: Agent, environment, reward, policy
- Markov Decision Processes (MDPs)
- Q-learning
- Applications of RL in robotics: Path planning, control, manipulation
- Case studies: RL for autonomous vehicles, robotic arms

Chapter 10: Machine Learning for Control Systems (2 weeks)
- Model predictive control (MPC) with machine learning
- Adaptive control using ML

Chapter 11: Machine Learning for Sensor Data (2 weeks)
- Processing sensor data: IMU, LiDAR, cameras
- Feature extraction and selection

Chapter 12: Ethics and Challenges in Machine Learning (1 weeks)
- Bias and fairness in ML models
- Overfitting, underfitting, and generalization
- Ethical considerations in AI and robotics

Laboratory Work:
1. LW 1: Introduction to Python and ML libraries (NumPy, pandas, scikit-learn).
2. LW 2: Implementing linear regression.
3. LW 3: SVM classification and Building and evaluating classification models.
4. LW 4: Simulating a reinforcement learning task/ Simulating a robotic task using RL (e.g., OpenAI Gym)
5. LW 5: Applying ML to a control system problem.

Assessment Methods:
- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:
1. Textbooks:
- "Pattern Recognition and Machine Learning" by Christopher M. Bishop.
- "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
- "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto.

2. Online Resources:
- Coursera: Machine Learning by Andrew Ng.
- TensorFlow and PyTorch documentation.
- OpenAI Gym for reinforcement learning.

Teaching Unit	Subject Title	Code	Semester
UEF1.2.1	**Optimization Techniques**	**OPT**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h30	**4/4**

Course Objectives:
This course introduces the principles and methods of optimization, focusing on both theoretical foundations and practical applications.
By the end of this course, students will:

- Understand the theoretical foundations of optimization.
- Formulate and solve optimization problems using appropriate techniques.
- Implement and evaluate optimization algorithms for real-world applications.
- Analyze and address challenges in optimization, such as scalability and non-convexity.

Targeted Skills

Upon completing this course, students will have developed the following skills:

- **Technical Skills:** Implementing optimization algorithms (e.g., gradient descent, simplex method). Using optimization libraries (e.g., SciPy, CVXPY).
- **Analytical Skills:** Formulating and solving optimization problems. Evaluating and improving algorithm performance.
- **Practical Skills:** Designing and testing optimization solutions for real-world applications. Debugging and optimizing algorithms.

Learning Outcomes:
By the end of this course, students will:

- Understand the theoretical foundations of optimization.
- Formulate and solve optimization problems using appropriate techniques.
- Implement and evaluate optimization algorithms for real-world applications.
- Analyze and address challenges in optimization, such as scalability and non-convexity.

Recommended Prerequisites
To succeed in this course, students should have:

- Basic knowledge of calculus, multivariable calculus and linear algebra.
- Mastering numerical analysis.
- Familiarity with probability and statistics.
- Programming experience in Python.

Course outline
Chapter 1: Mathematical Foundations & Optimization (1 week-3H)

- Convex Analysis.
- Multivariable calculus, Gradient, Hessian.
- Definition, application and types of optimization: constrained, linear, discrete …
- Optimality conditions: first-order and second-order conditions.

Chapter 2: Linear Programming (3 weeks-9H)

- Linear Program Formulation.
- Fundamental Solution Concepts – Convex Sets.
- Geometry View of Linear Programming.
- Solution Methods for Linear Programming:
  - Examples of modeling practical problem as linear program.
  - Graphical Solution Method.
  - Simplex Method (Phase Method, Big-M Method)
- Duality in Linear Programming:
  - Dual Programs
  - Relationship Between Primal and Dual Problems
  - Dual Simplex Method
- Special Cases in Linear Programming:
  - Transportation Problem
  - Assignment Problem

Chapter 3: Unconstrained Nonlinear Optimization (3 weeks-9H)

- Optimality Conditions:
  - Differential Case
  - Convex Case
- Algorithms for Unconstrained Optimization:
  - Gradient Descent and Steepest Descent Methods.
  - Newton's Method and Quasi-Newton Methods (e.g., BFGS, DFP).
  - Conjugate Direction Methods.
  - Convergence Analysis and Stopping Criteria.

Chapter 4: Nonlinear Constrained Optimization (3 weeks-9H)

- Necessary Optimality Conditions:
  - Equality Constraints and Lagrange Multipliers
  - Inequality Constraints, Karush-Kuhn-Tucker Condition, Second Order Conditions.
  - Convex Case and Saddle Point Theorem
- Algorithms for Constrained Optimization:
  - Feasible Direction Method
  - Projected Gradient Method
  - Augmented Lagrangian Method
  - Arrow-Hurwicz Algorithm

Chapter 5: Nonlinear Programming and Global Optimization (3 weeks-9H)

- Convex Optimization Problems.
- Interior-Point Methods.
- Challenges of Non-Convex Optimization.
- Heuristic Methods: Genetic Algorithms, Greedy Heuristic, Simulated Annealing.
- Applications in Engineering Design and Logistics.

Chapter 6: Project Management and Optimization Tools (2 weeks-6H)

- PERT Method for Project Scheduling.
- Applications of Optimization in Decision-Making.
- Case Studies in Operations Research.

Laboratory work

LW 1: Gradient Descent for Function Minimization.
LW 2: Linear Programming with the Simplex Method.
LW 3: Constrained Optimization Using KKT Conditions.
LW 4: Nonlinear Programming with Interior-Point Methods.
LW 5: Heuristic Optimization: Genetic Algorithms & Simulated Annealing.

Assessment Methods:

- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended textbooks and resources

1. Boyd, L. Vandenberghe, *Convex Optimization, Cambridge University Press,* (2004).
2. D. P. Bertsekas, *Nonlinear Programming,* **Athena Scientific** (3rd Edition, 2016).
3. M. Bierlaire, *Optimization: Principles and Algorithms, EPFL Press,* (2nd Edition 2018).
4. Cambini, L. Martein, *Generalized Convexity and Optimization*, Springer International Publishing, (2009).
5. E. K. P. Chong & S. H. Zak, *An Introduction to Optimization, John Wiley & Sons, (5*th Edition, 2023).
6. R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, Inc., (2nd Edition 2000).
7. H. Tuy, *Convex Analysis and Global Optimization*, Springer International Publishing (2nd Edition 2016).
8. J. Nocedal & S. J. Wright, Springer Series in Operations Research *and Financial Engineering: Numerical Optimization,* Springer Science+Business Media, LLC. (2nd Edition, 2006).
9. R. J. Vanderbei, *Linear Programming*, Springer International Publishing, (4th Edition, 2014).

Teaching Unit	Subject Title	Code	Semester
UEF1.2.2	**Microprocessors and microcontrollers**	**MPMC**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:

1. Understand the architecture and functionality of microprocessors and microcontrollers.
2. Learn the basics of assembly language programming and interfacing techniques.
3. Develop skills to design and implement embedded systems using microcontrollers.
4. Gain hands-on experience with hardware and software tools for microprocessor/microcontroller-based systems.

Course Outline:

Chapter 1: Introduction to Microprocessors and Microcontrollers (1 Week)

- Overview of microprocessors and microcontrollers
- Differences between microprocessors and microcontrollers
- Applications of microprocessors and microcontrollers in real-world systems
- Evolution of microprocessors (from 4-bit to 64-bit architectures)

Chapter 2: Microprocessor Architecture (1 week)

- Basic block diagram of a microprocessor
- Registers, ALU, control unit, and bus systems
- Instruction cycle: Fetch, Decode, Execute
- Overview of Intel 8085/8086 architecture (as a case study)

Chapter 3: Assembly Language Programming (1 week)

- Introduction to assembly language
- Instruction set architecture (ISA)
- Addressing modes: Immediate, Direct, Indirect, Register, etc.
- Writing simple programs (addition, subtraction, data transfer)

Chapter 4: Memory and I/O Interfacing (1 week)

- Memory organization: ROM, RAM, and cache
- Memory mapping and address decoding
- Interfacing I/O devices: Ports, latches, and buffers
- Interrupts and interrupt handling

Chapter 5: Microcontroller Architecture (1 week)

- Introduction to microcontrollers (e.g., 8051, PIC, ARM)
- Block diagram of a microcontroller
- On-chip peripherals: Timers, counters, serial communication, ADC/DAC

Chapter 6: The 8051 Microcontroller (1 week)

- Detailed architecture of the 8051 microcontroller
- Pin diagram and functionality
- Memory organization: Internal RAM, SFRs, and external memory
- Instruction set and addressing modes

Chapter 7: Programming the 8051 Microcontroller (2 weeks)

- Writing assembly programs for the 8051
- Timer/counter programming
- Serial communication programming

- Interrupt programming

Chapter 8: Advanced Microcontrollers (ARM Cortex-M) (1 week)
- Overview of ARM architecture
- Features of ARM Cortex-M series microcontrollers

Chapter 9: Real-Time Operating Systems (RTOS) for Microcontrollers (2 weeks)
- Introduction to RTOS
- Tasks, scheduling, and synchronization
- Implementing RTOS on microcontrollers

Chapter 10: Communication Protocols (2 weeks)
- Serial communication: UART, SPI, I2C
- Wireless communication: Bluetooth, Zigbee, Wi-Fi
- IoT and microcontroller-based communication systems

Chapter 12: Introduction to Design and Development of Embedded Systems (1 week)
- System design methodology
- Case studies: Home automation, robotics, automotive systems
- Debugging and testing embedded systems

Laboratory Work:

1. Lab 1: Writing and executing simple assembly language programs.

2. Lab 2: Interfacing LEDs and switches with a microcontroller/ Timer/counter programming.

3. Lab 3: Serial communication programming.

4. Lab 4: Interfacing LCDs/ sensors (e.g., temperature sensor LM11) with a microcontroller.

Assessment Methods:
- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources:

1. Textbooks:
- "Microprocessors and Interfacing" by Douglas V. Hall.
- "ARM System-on-Chip Architecture" by Steve Furber.

2. Online Resources:
- ARM Cortex-M documentation and tutorials.
- Embedded systems forums and communities (e.g., Stack Overflow, EEVblog).

Teaching Unit	Subject Title	Code	Semester
UEF1.2.2	**Digital Control Systems**	**DCS**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:

1. To understand the fundamental principles of digital control systems.
2. To analyze and design digital controllers using various techniques.
3. To implement digital control systems using software tools and hardware.
4. To apply digital control concepts to real-world applications.

Course Outline:

Chapter 1: Introduction to Digital Control Systems (2 Weeks)
- Overview of control systems: analog vs. digital
- Advantages and challenges of digital control
- Applications of digital control systems
- Review of continuous-time systems and Laplace transforms

Chapter 2: Sampling and Reconstruction (2 Week)
- Sampling process and sampling theorem
- Aliasing and anti-aliasing filters
- Zero-order hold (ZOH) and reconstruction
- Quantization and its effects

Chapter 3: Z-Transform and Discrete-Time Systems (3 Weeks)
- Introduction to Z-transform
- Properties of Z-transform
- Inverse Z-transform
- Transfer functions of discrete-time systems
- Stability analysis in the Z-domain

Chapter 4: Discrete-Time System Analysis (4 Weeks)
- Difference equations and their solutions
- State-space representation of discrete-time systems
- Controllability and observability
- Discrete-time system response: transient and steady-state

Chapter 5: Digital Controller Design (4 Weeks)
- Design specifications for digital control systems
- Digital PID controllers
- Direct design methods: root locus and frequency response
- Pole placement and state feedback control
- Deadbeat control

Laboratory Work:

LW 1: Sampling and Reconstruction
- Sampling of analog signals
- Reconstruction using ZOH

LW 2: Z-Transform and System Analysis
- Z-transform of discrete signals
- Analysis of discrete-time systems

LW 3: Digital Controller Design

- Design and simulation of digital PID controllers
- Pole placement and state feedback control

LW 4: Real-Time Implementation

- Implementation of digital controllers on microcontrollers or DSPs
- Real-time control of a physical system (e.g., of a DC motor)

Assessment Methods:

- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources:

1. "Digital Control Systems", by Benjamin C. Kuo.
2. "Modern Control Engineering", by Katsuhiko Ogata.
3. "Digital Control of Dynamic Systems", by Gene F. Franklin, J. David Powell, and Michael L. Workman.
4. Online resources: MIT OpenCourseWare, Control Systems Lectures on YouTube, and MATLAB tutorials.

Teaching Unit	Subject Title	Code	Semester
UEF1.2.2	**Mechanical systems**	**IMS**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30		**3/3**

Course Objectives:
This course equips students with the knowledge and skills to select, validate, and size an optimal mechanical system solution. It introduces the general theory of mechanisms, provides an in-depth analysis of various types of joints and transmission systems, and presents a structured approach for mechanism analysis and synthesis.

Learning Outcomes
By the end of this course, students will be able to:

- Select, validate, and dimension a mechanical system.
- Understand the fundamental theory of mechanisms.
- Identify and analyze different types of kinematic pairs and transmission systems.
- Apply methodologies for mechanism analysis and synthesis.
- Model and solve mechanical problems related to complex systems.

Recommended Prerequisites
- **Mathematics:** Matrix operations, determinants, linear system resolution.
- **Mechanics:** Vector analysis, rigid body mechanics, kinematics, and dynamics.
- **Engineering Drawing:** Technical sketching, industrial design standards.
- **General Technology:** Mechanical components, materials, and manufacturing processes.

Course outline:

Chapter 1: Introduction to Mechanism Theory (1 week)
- Definitions and key terminology in mechanical systems
- Historical evolution and development of mechanisms
- Applications in robotics, automation, and mechanical engineering
- The role of mechanical systems in industrial processes

Chapter 2: Contact and Kinematic Pairs (2 weeks)
- Structural chains and interaction between rigid bodies
- Types of kinematic pairs: revolute, prismatic, spherical, cylindrical, etc.
- Contact modeling and classification of constraints
- Specialized transmission mechanisms
- Classification of kinematic systems

Chapter 3: Mechanical Actions and Force Transmission (2 weeks)
- Concepts of force and moment transmission
- Models of transmissible mechanical actions
- Friction laws and their effects on mechanical efficiency

- Analysis of internal and external force interactions

Chapter 4: Kinematic Modeling and Parameterization (2 weeks)

- Degrees of freedom and mobility
- Kinematic diagrams and linkage graphs
- Grübler-Kutzbach criterion for mobility analysis
- Input-output equations and functional analysis of mechanisms

Chapter 5: Kinematic Analysis of Mechanisms (2 weeks)

- Velocity and acceleration analysis of mechanical components
- Graphical and analytical methods for kinematic studies
- Numerical simulation of mechanical motion
- Case studies and industrial applications

Chapter 6: Static Analysis of Mechanisms (2 weeks)

- Static force equilibrium and load distribution
- Force diagrams and wrench notation
- Resolution of static force systems
- Application to common mechanical structures

Chapter 7: Dynamic Analysis of Mechanisms (2 weeks)

- Fundamental principles of dynamics applied to mechanical systems
- Newton-Euler equations and inertia forces
- Kinetic energy, work, and power in mechanical systems
- Analysis of mechanical vibrations and resonance phenomena

Chapter 8: Mechanism Synthesis and Optimization (2 weeks)

- Design methodologies for mechanical systems
- Performance criteria and reliability considerations
- Case studies in mechanical system design
- Integration of CAD software and simulation tools

Chapter 9: Power Transmission via Belt and Pulley Systems (1 week)

- Principles of power transmission using belt drives
- Types of belts and their influence on mechanical efficiency
- Calculation of belt forces and tension distribution
- Industrial applications of belt-driven systems

Chapter 10: Power Transmission via Gear Systems (1 week)

- Classification and function of gear mechanisms
- Calculation of transmission ratios and meshing conditions
- Load distribution, wear analysis, and efficiency considerations
- Design and selection of gears for mechanical applications

Assessment Methods:

- Regular in class assessments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:

1. **Michel Aublin, René Boncompain, Michel Boulaton, Daniel Caron, Émile Jeay et al.,** *Mechanical Systems: Theory and Sizing*, Dunod, 2020.
2. **Jean-Dominique Mosser, Jean-Jacques Marchandeau, Jacques Tanoh,** *Industrial Sciences for Engineers, First Year MPSI-PCSI-PTSI*, Dunod, 2015.
3. **Pierre Agati, Gérard Delville, Yves Brémont,** *Solid Mechanics: Industrial Applications (2nd Edition)*, Dunod, 2020.

Teaching Unit	Subject Title	Code	Semester
UEM1.2.1	**Computer Aided Design**	**CAD**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h30	**3/3**

Course Description:
This course provides an in-depth understanding of Computer-Aided Design (CAD) and its applications in modern engineering and product development. Students will explore CAD modeling techniques, geometric modeling principles, numerical simulation, and emerging trends such as additive manufacturing and digital twins.

Course Objectives:
By the end of this module, students will be able to:
- Understand the role of CAD in the product lifecycle.
- Apply different modeling approaches to create 3D designs.
- Utilize geometric modeling techniques for complex structures.
- Implement mathematical models of curves and surfaces in CAD.
- Conduct numerical simulations for design validation.
- Integrate CAD with additive manufacturing technologies.
- Optimize designs using ecodesign principles and topology optimization.
- Explore integrated design methods and digital twin applications.

Recommended Prerequisites:
Engineering Drawing, Basics of Mechanical Design, Introduction to Robotics.

Course Outline:
Chapter 1: Introduction to the Product Lifecycle (1 week)
- Product lifecycle stages: concept, design, manufacturing, usage, and recycling.
- Role of CAD in modern product development.
- Integration of CAD with PLM (Product Lifecycle Management) systems.

Chapter 2: Modeling a Product (Approaches and Methods) (1 week)
- Parametric vs. direct modeling.
- Feature-based and constraint-based modeling.
- Assembly modeling and design intent.

Chapter 3: Geometric Modelers in CAD (2 weeks)
- Volumetric modeling (solid modeling techniques).
- Surface modeling for complex shapes.
- Parametric modeling and constraint-based design.

Chapter 4: Mathematical Models of Curves and Surfaces (2 weeks)
- Bézier, B-Spline, and NURBS curves.
- Surface representation and continuity.
- Practical applications in CAD software.

Chapter 5: Numerical Simulation and Analysis (2 weeks)
- Finite Element Analysis (FEA) basics.
- Stress, strain, and deformation analysis.
- Thermal and fluid flow simulations.

- Motion analysis and dynamic simulations.
- Optimization of designs using simulation results.

Chapter 6: CAD Data Management and Collaboration (1 week)

- File formats and interoperability (e.g., STEP, IGES, STL).
- Version control and data management.
- Collaborative design using cloud-based CAD tools.
- Introduction to Product Lifecycle Management (PLM).

Chapter 7: CAD and Additive Manufacturing (2 weeks)

- CAD for 3D printing: file formats and preparation.
- Digital machining and subtractive manufacturing.
- Design considerations for additive manufacturing.

Chapter 8: Ecodesign and Topological Optimization (1 week)

- Sustainable design principles in CAD.
- Topology optimization for lightweight structures.
- Case studies on sustainable product development.

Chapter 9: Integrated Design, Simultaneous Engineering, and Digital Twins (2 weeks)

- Concurrent engineering and collaborative CAD workflows.
- Digital twin concepts and applications.
- Future trends in CAD and smart manufacturing.

Laboratory Work:

Lab 1: Introduction to CAD software interface and basic 2D drafting.
Lab 2: 3D modeling of 3D objects.
Lab 3: Advanced 3D modeling techniques and parametric design.
Lab 4: Assembly design and creating exploded views.
Lab 5: Kinematic analysis of robotic components.
Lab 6: Stress analysis using FEA tools.
Lab 7: Motion analysis and dynamic simulations.
Lab 8: Topological Optimization
Lab 9: Ecodesign

Assessment Methods:

- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:

1. "Engineering Design with SolidWorks" – David Planchard
2. "Computer-Aided Design and Manufacturing" – Mikell P. Groover
3. **"CAD Principles and Applications"** by P.N. Rao.
4. **"Introduction to Finite Element Analysis Using SOLIDWORKS Simulation"** by Randy H. Shih.
5. Online tutorials and documentation for AutoCAD, SolidWorks, and Fusion 360.

Software & Tools:

- CAD Software: SolidWorks, CATIA, AutoCAD, Fusion 360
- Simulation Tools: ANSYS, Abaqus, Simulink
- Additive Manufacturing: 3D printing software (Cura, PrusaSlicer)

Teaching Unit	Subject Title	Code	Semester
UEM1.2.1	**Engineering Programming**	**ENGP**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h00	**2/2**

Course Description:
This course introduces students to the fundamental concepts of programming with a focus on applications in robotics and autonomous systems. The course covers programming paradigms, data structures, algorithms, databases, networking, and software development practices essential for designing and implementing robotic systems. Students will gain hands-on experience with programming languages such as Python, C++, and MATLAB, and will apply these skills to solve real-world problems in robotics.

Course Objectives:
By the end of this course, students will be able to:

1. Understand and apply basic programming concepts such as variables, control structures, functions, and object-oriented programming.
2. Develop algorithms to solve engineering problems, particularly in robotics and autonomous systems.
3. Implement and manipulate data structures such as arrays, linked lists, stacks, queues, and trees.
4. Write, debug, and optimize code in Python, C++, and MATLAB.
5. Apply programming skills to control robotic systems, process sensor data, and implement autonomous behaviors.
6. Understand the principles of software engineering, including version control, testing, and documentation.
7. Gain experience with robotics-specific libraries and frameworks such as ROS (Robot Operating System).
8. Develop a foundational understanding of real-time systems and embedded programming for robotics.
9. Understand and apply database concepts for data storage and retrieval in robotic systems.
10. Implement networking concepts for communication between robotic systems and other devices.

Recommended Prerequisites:

- Basic understanding of mathematics (linear algebra, calculus).
- Familiarity with basic computer science concepts.

Course Outline:
Chapter 1: Introduction to Programming Concepts (2 weeks)

- Overview of programming languages (Python, C++, MATLAB)
- Variables, data types, and operators
- Control structures: loops and conditionals
- Functions and modular programming

Chapter 2: Object-Oriented Programming (OOP) (2 weeks)
- Classes and objects
- Inheritance, polymorphism, and encapsulation
- Design patterns in robotics

Chapter 3: Data Structures and Algorithms (2 weeks)
- Arrays, linked lists, stacks, and queues
- Trees and graphs
- Sorting and searching algorithms

Chapter 4: Software Development Practices (2 weeks)
- Version control with Git
- Debugging and testing
- Code documentation and style

Chapter 5: Robotics Programming with Python and C++ (2 weeks)
- Introduction to ROS (Robot Operating System)
- Writing ROS nodes and topics
- Sensor data processing and actuator control

Chapter 6: Databases for Robotics (2 weeks)
- Introduction to databases and SQL
- Data modeling and database design
- Using databases in robotic systems for data storage and retrieval

Chapter 7: Networking for Robotics (2 weeks)
- Basics of computer networks
- Communication protocols (TCP/IP, UDP, MQTT)
- Network programming in Python and C++

Laboratory Work :
- **Lab 1:** Basic Python programming exercises
- **Lab 2:** Implementing OOP concepts in C++
- **Lab 3:** Implementing data structures in Python
- **Lab 4:** Collaborative coding project using GitHub
- **Lab 5:** Simulating a simple robot in ROS
- **Lab 6:** Implementing a simple database for sensor data logging
- **Lab 7:** Implementing a simple client-server communication for robotic systems

Assessment Methods:
- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:
- **Textbooks**

1. "Python Programming for Engineers" by Dr. John Zelle
    - Focuses on Python programming with applications in engineering.
2. "C++ for Engineers and Scientists" by Gary J. Bronson
    - Covers C++ programming with a focus on engineering applications.
3. "Programming Robots with ROS: A Practical Introduction to the Robot Operating System" by Morgan Quigley, Brian Gerkey, and William D. Smart
    - A comprehensive guide to using ROS for robotics programming.

4. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
    - A classic text on algorithms and data structures.
5. "MATLAB for Engineers" by Holly Moore
    - Introduces MATLAB programming for engineering applications.
6. "Database Systems: The Complete Book" by Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom
    - Comprehensive coverage of database concepts and SQL.
7. "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross
    - Detailed introduction to networking concepts and protocols.

- **Online Courses:**
    - "Python for Everybody" by Dr. Charles Severance (Coursera)
    - "C++ Programming" by University of London (Coursera)
    - "ROS for Beginners" by Anis Koubaa (Udemy)
    - "Introduction to Databases" by Stanford University (Coursera)
    - "Computer Networking" by Georgia Institute of Technology (Coursera)
- **Software Tools:**
    - Python IDEs (PyCharm, Jupyter Notebook)
    - C++ IDEs (Visual Studio, Code::Blocks)
    - MATLAB
    - ROS (Robot Operating System)
    - Database Management Systems (MySQL, PostgreSQL)
    - Network simulation tools (Wireshark, GNS3)
- **Online Documentation:**
    - Python Documentation: https://docs.python.org/3/
    - C++ Reference: https://en.cppreference.com/
    - ROS Wiki: http://wiki.ros.org/
    - SQL Documentation: https://dev.mysql.com/doc/
    - Networking Protocols: https://tools.ietf.org/html/

Teaching Unit	Subject Title	Code	Semester
UEM1.2.1	**Electric Actuators for Robotics**	**EAR**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h00	**2/2**

Course Objectives:
1. To understand the principles, types, and characteristics of electric actuators used in robotics.
2. To analyze the design, control, and integration of electric actuators in robotic systems.
3. To develop practical skills in selecting, modeling, and controlling actuators for specific robotic applications.
4. To explore advanced topics such as energy efficiency, precision control, and emerging technologies in electric actuation.

Learning Outcomes:
By the end of the module, students will be able to:
1. Explain the working principles and characteristics of various electric actuators.
2. Select and design appropriate actuators for specific robotic applications.
3. Model and control electric actuators using mathematical and software tools.
4. Integrate actuators into robotic systems while considering mechanical, electrical, and software constraints.
5. Analyze and optimize actuator performance in terms of energy efficiency, precision, and reliability.

Course Outline:
Chapter 1: Introduction to Electric Actuators (2 weeks)
- Overview of actuators in robotics: importance and applications.
- Classification of actuators: electric, hydraulic, pneumatic, and others.
- Advantages and limitations of electric actuators.
- Key performance metrics: torque, speed, efficiency, precision, and power density.

Chapter 2: Fundamentals of Electromechanical Systems (2 weeks)
- Basic principles of electromagnetism and electromechanical energy conversion.
- Review of electrical circuits and power electronics relevant to actuators.
- Introduction to motors and generators as actuators.

Chapter 3: DC Motors (1 week)
- Working principles of brushed and brushless DC motors (BLDC).
- Mathematical modeling of DC motors.
- Speed-torque characteristics and control methods (PWM, H-bridge).
- Applications in robotics: wheeled robots, robotic arms, etc.

Chapter 4: Stepper Motors (1 week)
- Principles of operation: full-step, half-step, and microstepping.
- Open-loop and closed-loop control of stepper motors.
- Applications in precision robotics: 3D printers, CNC machines, etc.

Chapter 5: AC Motors and Induction Motors (2 weeks)
- Overview of AC motors and their use in robotics.
- Induction motors: principles, modeling, and control.

- Applications in industrial robotics and heavy-duty systems.

Chapter 6: Servo Motors (1 week)

- Principles of operation and types of servo motors (AC, DC, brushless).
- Position, speed, and torque control using feedback systems (encoders, resolvers).
- Applications in robotic arms, drones, and autonomous vehicles.

Chapter 7: Linear Actuators (1 week)

- Principles of linear electric actuators: screw-driven, belt-driven, and piezoelectric.
- Applications in robotics: linear motion systems, grippers, and positioning systems.

Chapter 8: Actuator Control Systems (2 weeks)

- Introduction to control theory for actuators: PID control, state-space models.
- Motor drivers and power electronics: H-bridges, inverters, and amplifiers.
- Embedded systems for actuator control: microcontrollers and DSPs.

Chapter 9: Energy Efficiency and Thermal Management (1 week)

- Energy consumption and efficiency in electric actuators.
- Thermal modeling and heat dissipation techniques.
- Design considerations for energy-efficient robotic systems.

Chapter 10: Integration of Actuators in Robotic Systems (1 week)

- Mechanical integration: coupling, gearing, and transmission systems.
- Electrical integration: wiring, power supplies, and signal conditioning.
- Software integration: interfacing actuators with robotic control systems (ROS, MATLAB, etc.).

Chapter 11: Advanced Topics in Electric Actuators (1 week)

- Emerging technologies: smart actuators, shape memory alloys, and piezoelectric actuators.
- Miniaturization and micro-actuators for medical and micro-robotics.
- Trends in actuator design for autonomous systems and AI-driven robotics.

Assessment Methods:

- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:

1. "Electric Motors and Drives: Fundamentals, Types, and Applications" by Austin Hughes and Bill Drury.
2. "Robotics: Modelling, Planning, and Control" by Bruno Siciliano and Lorenzo Sciavicco.
3. "Modern Control Engineering" by Katsuhiko Ogata (for control theory).
4. Online resources: MATLAB tutorials, ROS documentation, and manufacturer datasheets (e.g., Maxon, Faulhaber).

Teaching Unit	Subject Title	Code	Semester
UET1.2.1	**Introduction to UAVs Operations and Regulations**	**UAVOR**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h00			**1/1**

Course Objectives:
At the end of this course, students should be able to:

- Understand the fundamentals of UAV operations, components, and classifications.
- Learn and apply UAV flight regulations at national and international levels.
- Assess risk management and safety procedures in UAV operations.
- Explore the ethical, legal, and societal impact of UAV technologies.

Course Outline:

Chapter 1: Introduction to UAVs (1 week)

- Definition and historical evolution of UAVs
- UAV classification: fixed-wing, rotary-wing, hybrid UAVs
- Applications of UAVs: military, commercial, and industrial uses

Chapter 2: UAV Components and Systems (2 weeks)

- Airframe and propulsion systems
- Avionics, sensors, and communication systems
- Power systems and battery management

Chapter 3: International and National UAV Regulations (2 weeks)

- Overview of regulatory authorities (FAA, EASA, ICAO)
- Overview of the **Algerian** UAV **Regulation**
- UAV classification under regulatory frameworks
- Registration, certification, and operational limitations

Chapter 4: UAV Operational Procedures (2 weeks)

- UAV pilot licensing and qualifications
- Pre-flight checks and mission planning
- Post-flight procedures and data analysis

Chapter 5: Airspace Management and UAV Integration (2 weeks)

- Controlled vs uncontrolled airspace
- UAV Traffic Management (UTM) systems
- Beyond Visual Line of Sight (BVLOS) operations

Chapter 6: Risk Management and Safety in UAV Operations (2 weeks)

- Risk assessment techniques for UAV missions
- Collision avoidance and emergency procedures
- Case studies on UAV incidents and accidents

Chapter 7: UAV Communication and Navigation Systems (1 week)

- GPS-based navigation and alternative positioning systems
- Data links and ground control stations (GCS)
- Autonomous flight and remote piloting

Chapter 8: Commercial and Industrial UAV Applications (1 week)

- UAVs in agriculture, infrastructure inspection, and logistics
- Drone delivery systems and urban air mobility
- Future trends in UAV commercial markets

Chapter 9: Hands-on UAV Mission Planning and Simulation (1 week)

- UAV flight simulation tools (e.g., ArduPilot, PX4, Gazebo)
- Planning and executing a UAV mission in a controlled environment
- Data collection and analysis

Assessment Methods:

- Final exam (100%).

Recommended Textbooks and Resources:

1. *Introduction to Unmanned Aircraft Systems* – Barnhart, Shappee, and Marshall.
2. *Unmanned Aircraft Systems: UAVS Design, Development, and Deployment* – R. Austin.
3. Regulatory documents from **FAA, EASA, ICAO, and local authorities.**

Teaching Unit	Subject Title	Code	Semester
UET1.2.1	**Capstone Project I**	**CAP-I**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH			1h30	**1/1**

Course Objectives:

- Form teams and identify project ideas.
- Conduct feasibility analysis and initial design documentation.
- Establish milestones and present progress to faculty advisors.

Targeted Skills

- Teamwork and collaboration.
- Problem identification and solution structuring.
- Technical writing and presentation skills.
- Project planning and feasibility assessment.

Recommended prerequisites

- Fundamentals of Autonomous systems.
- Basic knowledge of Robotics.
- Basic project management.
- Technical writing and documentation.

Course Content

1. **Team Formation and Brainstorming (3 weeks)**
   - Introduction to project selection.
   - Team roles and responsibilities.
   - Idea generation techniques.
2. **Feasibility Analysis and Initial Design Documentation (5 weeks)**
   - Market and technical feasibility.
   - Risk assessment.
   - Drafting project proposals and documentation.
3. **Regular Milestones and Faculty Presentations (6 weeks)**
   - Setting up project goals and deadlines.
   - First milestone presentation.
   - Refining project scope based on feedback.

Recommended textbooks and Resources

- Project Management for Engineering and Technology by J. Nicholas
- Engineering Design: A Project-Based Introduction by Clive Dym

Teaching Unit	Subject Title	Code	Semester
UED1.2.1	**Linux Operating System**	**LOS**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH			1h30	**1/1**

Course Objectives:
By the end of this module, students will be able to:
1. Understand the fundamental concepts of operating systems, including their role in managing hardware and software resources.
2. Explore the architecture and components of the Linux operating system, including the kernel, shell, and file system.
3. Gain hands-on experience with Linux through basic shell commands, scripting, and system administration tasks.
4. Understand Linux process management, including processes, threads, and scheduling.
5. Apply synchronization techniques in a Linux environment.
6. Manage memory and I/O operations in Linux.
7. Explore embedded Linux and its real-world applications.

Course outline:
Chapter 1: Introduction to Operating Systems (1 week)
- Overview of Operating Systems
  - Definition and purpose of an OS
  - Types of Operating Systems (e.g., Batch, Real-time, Distributed, etc.)
- Key functions of an OS:
  - Process management
  - Memory management
  - File systems
  - I/O management
- Evolution of Operating Systems

Chapter 2: Linux Architecture (3 weeks)
-Kernel, Shell, and User Space
  - Role of the Linux kernel
  - Shell as a command interpreter
  - User-space applications
- Monolithic vs. Microkernel Architecture
  - Comparison with other OS architectures
- Linux Distributions
  - Overview of popular distributions (e.g., Ubuntu, Fedora, Debian)
  - Choosing a distribution for different use cases

Chapter 3: Linux Shell and Command Line (3 weeks)
- Introduction to the SHELL
  - What is a shell? (e.g., Bash, Zsh)
  - Basic shell commands (`ls`, `cd`, `mkdir`, `rm`, `cp`, `mv`, etc.)
- Shell Scripting Basics
  - Writing and executing shell scripts
  - Variables, loops, and conditionals
  - Automating tasks with scripts
- File System Navigation

- Linux file system hierarchy (`/bin`, `/etc`, `/home`, etc.)
- File permissions and ownership (`chmod`, `chown`)

Chapter 4: Linux Process Management (2 weeks)
- Processes in Linux
- Process lifecycle
- Commands to manage processes (`ps`, `top`, `htop`, `kill`)
- Threads and Multithreading
- Creating threads using `pthread` in C
- Process Scheduling
- Overview of Linux scheduling algorithms (CFS - Completely Fair Scheduler)
- Using `nice` and `renice` to adjust process priorities

Chapter 5: Synchronization in Linux (2 weeks)
-Synchronization Mechanisms
- Semaphores, mutexes, and monitors
- Implementing synchronization in C using POSIX threads
- Classic Synchronization Problems
- Producer-Consumer problem
- Dining Philosophers problem
- Deadlock Prevention and Avoidance
- Practical examples in Linux

Chapter 6: Memory and I/O Management in Linux (3 weeks)
- Memory Management
- Virtual memory and paging
- Commands to monitor memory usage (`free`, `vmstat`)
- I/O Management
- Disk scheduling in Linux
- File I/O operations in C (`open`, `read`, `write`, `close`)
- Buffering and caching

Chapter 7: Embedded Linux (1 week)
- Overview of Embedded Systems
- What are embedded systems?
- Role of Linux in embedded systems
- Customizing Linux for Embedded Devices
- Using Buildroot or Yocto to create a minimal Linux system
- Cross-compiling applications for embedded targets
- Real-world Applications
- Case studies of embedded Linux in IoT, robotics, and automotive systems

Laboratory Work

Lab 1: Introduction to Linux
- Install a Linux distribution (e.g., Ubuntu) on a virtual machine.
- Explore the Linux file system hierarchy.
- Practice basic shell commands (`ls`, `cd`, `mkdir`, `rm`, `cp`, `mv`, etc.).
- Write and execute a simple shell script.

Lab 2: Linux Shell Scripting
- Write a shell script to automate a task (e.g., backup files, monitor system resources).
- Use variables, loops, and conditionals in scripts.
- Schedule scripts using `cron`.

Lab 3: Linux Process Management
- Use commands like `ps`, `top`, and `htop` to monitor processes.
- Write a C program to create processes using `fork()`.
- Write a multithreaded program using `pthread`.

Lab 4: Synchronization in Linux
- Implement the Producer-Consumer problem using semaphores.
- Solve the Dining Philosophers problem using mutexes.
- Analyze and prevent deadlock scenarios.

Lab 5: Memory and I/O Management
- Write a C program to simulate memory allocation and paging.
- Perform file I/O operations (`open`, `read`, `write`, `close`).
- Monitor memory usage using `free` and `vmstat`.

Lab 6: Embedded Linux
- Set up a minimal Linux environment using Buildroot.
- Cross-compile a simple application for an embedded target (e.g., Raspberry Pi).
- Deploy and test the application on the embedded device.

Assessment Methods:
- Laboratory reports and practical assignments (100%).

Recommended Textbooks and Resources

Textbooks

1. Linux Kernel Development, by Robert Love
2. Operating System Concepts, by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne
3. The Linux Command Line, by William E. Shotts Jr.
4. Embedded Linux System Design and Development, by P. Raghavan, Amol Lad, and Sriram Neelakandan

Online Resources

1. Linux Documentation Project (https://tldp.org/)
2. Kernel.org (https://www.kernel.org/)
3. GNU/Linux Command Line Tools Summary

Teaching Unit	Subject Title	Code	Semester
UED1.2.1	**Training Internship I**	**TI-I**	**S2**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH			1h30	**1/1**

Course Objectives

- Introduce students to professional work environments.
- Develop basic technical and soft skills in an industry setting.
- Understand company structure, workflow, and project dynamics.

Targeted Skills

- Professional communication.
- Basic technical proficiency.
- Adaptability to work environments.
- Teamwork and collaboration.

Recommended prerequisites

- Fundamentals of Robotics and Autonomous systems.
- Basic project management principles.

Content

1. **Introduction to the Company (3 hours)**
    - Company overview and mission.
    - Workplace safety and regulations.
2. **Observation and Shadowing (4 hours)**
    - Understanding team roles and responsibilities.
    - Exposure to ongoing projects.
3. **Hands-on Experience (6 hours)**
    - Assisting in minor tasks and operations.
    - Learning basic tools and methodologies.
4. **Final Review and Feedback (3 hours)**
    - Supervisor feedback session.
    - Internship report preparation.

Recommended Texbooks and Resources

- The Internship, Practicum, and Field Placement Handbook by Brian N. Baird

Professionalism: Skills for Workplace Success by Lydia E. Anderson

Detailed programs for the seventh semester

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.1.1	**Nonlinear and Optimal Control Design**	**NLOCD**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:
By the end of this course, students will:

- Understand the challenges of nonlinear systems.
- Analyze stability and behavior of nonlinear systems.
- Design and implement nonlinear control strategies.
- Understand the principles of optimal control.
- Design and implement LQR and MPC controllers.
- Apply optimal control techniques to real-world problems.

Recommended prerequisites:

- Good knowledge of Continuous and Discrete-Time Control Systems.
- Strong mathematical background (calculus, linear algebra, differential equations).

Course Outline:

Chapter 1: Nonlinear systems (2 weeks)

- Nonlinear state space model.
- Existence conditions of the solution.
- Tangent linear model and first Lyapunov theorem.
- Lyapunov function.
- Second Lyapunov theorem and equilibrium stability analysis.
- Lassale/Barbalat extensions.

Chapter 2: Nonlinear Control Techniques (2 weeks)

- Feedback linearization.
- Sliding mode control.
- Back stepping and Forwarding control.

Chapter 4: Adaptive Control (2 weeks)

- Indirect Adaptive Control
- Direct Adaptive Control
- Adaptive Model reference control.

Chapter 5: Fuzzy modeling and Control (1 week)

- Fuzzy sets and fuzzy numbers.
- Fuzzy inference systems
- Mamdani Fuzzy controller
- TSK models and control design

Chapter 6: Neural Network modeling and Control (1 week)

- Neural Network approximation and control
- Adaptive Neuro Fuzzy Inference System (ANFIS) control.
- Applications in control and robotics.

Chapter 7: Introduction to Optimal Control (1 week)

- Problem formulation and cost functions.

- Euler-Lagrange equation.
- Hamilton-Jacobi-Bellman equation.
- Applications in engineering and economics.

Chapter 8: Linear Quadratic Regulator (LQR) (1 week)

- Design and implementation of linear quadratic regulator.
- Applications in state feedback control.

Chapter 9: Dynamic Programming (1 week)

- Principle of optimality.
- Applications in control and robotics.

Chapter 10: Model Predictive Control (MPC) and Adaptive Control (3 weeks)

- Principles and implementation.
- Linear and Nonlinear MPC.
- Direct and Indirect Adaptive Control.
- Adaptive Model reference control.
- Applications in process control and robotics.

Laboratory Work

Lab 1: Control of NL Systems using 1st and 2nd Lyapunov method
Lab 2: Sliding Mode Control (SMC) and adaptive control
Lab 3: TSK based Fuzzy control for NL systems
Lab 4: Control of NL Systems using LQ method
Lab 5: Minimum time control/Minimum energy control of Planner UAV
Lab 6: Model Predictive Control for NL systems

Assessment Methods:

- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources:

1. P. Corke, *Robotics, Vision and Control,* Springer International Publishing, (2nd Edition 2017).
2. F. Csáki, *Modern Control Theories: Nonlinear, Optimal and Adaptive Systems*, Akadémiai Kiadó, (Budapest 1972).
3. R. C. Dorf, R. H. Bishop, *Moden Control Systems*, Prentice-Hall, Inc., (11th Edition 2011).
4. M. J. Grimble, P. Majecki, *Nonlinear Industrial Control Systems,* Springer-Verlag London Ltd., (2020).
5. B. C. Kuo, *Automatic Control Systems*, Prentice-Hall, Inc., (3rd Edition 1975).
6. W. S. Levine, *Control system Fundamental*, CRC Press, Taylor & Francis Group, (2011).
7. I. J. Nagrath, M. Gopal, *Control Systems Engineering*, New Age International (P) Ltd., (4th Edition 2006).
8. N. S. Nise, *Control Systems Engineering*, John Wiley & Sons, Inc., (7th Edition 2015).

9. T. L. Vincent, W. J. Grantham, *Nonlinear and Optimal Control Systems*, John Wiley & Sons, Inc., (1997).

10. Q. Wei, R. Song, B. Li, X. Li, *Self-Learning Optimal Control of Nonlinear Systems*, Science Press, (2018).

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.1.1	**Multi-Physics Modeling of Technological Systems**	**MPMTS**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:
This module aims to provide students with the knowledge and tools necessary to:

- Understand the principles of multi-physics modeling applied to technological systems.
- Integrate the interactions between different disciplines (mechanical, thermal, electrical, hydraulic, etc.).
- Build and simulate mathematical models of complex systems.
- Use specialized software for multi-physics modeling and analysis.

Recommended Prerequisites:

- Basic knowledge of mechanics, electronics, and control systems
- Familiarity with programming and numerical methods is recommended

Learning Outcomes:

- Upon successful completion of this course, students will be able to:
- Formulate and implement multiphysics models for various technological systems.
- Utilize numerical and computational methods to solve coupled physical problems.
- Evaluate the impact of interactions between mechanical, electrical, thermal, and fluidic domains in engineering applications.
- Conduct simulations using industry-standard software and interpret the results.
- Optimize robotic and mechatronic systems through validated multiphysics models.
- Apply multiphysics concepts to innovative designs and research in robotics

Course Outline:

Chapter 1: Role of Simulation in the Design Cycle of Complex Technological Systems (1 Week)

Chapter 2: Fundamental Concepts of Lumped Parameter-Based Multi-Physics Modeling (2 Weeks)

2.1. Definition and modeling levels of mechatronic systems
2.2. Modeling of mechatronic systems with lumped parameters
2.3. Multi-physics modeling of a power window system

Chapter 3: Setting Up a Lumped Parameter Model (2 Weeks)

3.1. Introduction to the notion of adapted model
3.2. Identifying the main effects
3.3. Modeling approaches and selection of adapted models

Chapter 4: Numerical Simulation of Multi-Physics Systems (2 Weeks)

4.1. From mathematical model to numerical model
4.2. From numerical model to computer simulated model (causality, consistency, Bond graph modeling)
4.3. Simulation: numerical resolution of ODEs
4.4. The main sources of error in modelling and simulation

Chapter 5: Dynamic Performance Analysis Tools (2 Weeks)

5.1. Dynamic performance indicators
5.2. Laplace transform and transfer functions
5.3. Stability of linear dynamic systems
5.4. Analysis of first- and second-order systems. Model reduction

Chapter 6: Mechanical and Electromechanical Power Transmissions (2 Weeks)

6.1. Variational approaches
6.2. Modeling by direct integration of local laws: bulk and multi-layer ceramics
6.3. Principle of virtual works: amplified actuators
6.4. Energy and co-energy balances
6.5. Lagrange equations: Langevin transducers

Chapter 7: Power Transmission by Low-Compressibility Fluids (2 Weeks)

7.1. Fluid power
7.2. Presentation of a helicopter actuation system
7.3. Minimal fluid modeling according to the phenomena involved
7.4. Modeling of the various physical phenomena
7.5. Modeling of the main hydraulic components

Chapter 8: Heat Power Transmission (1 Weeks)

8.1. Heat exchangers
8.2. Effectiveness-based thermal modeling of heat exchangers. Constant effectiveness
8.3. Estimation of the heat exchanger effectiveness
8.4. Estimation of the global heat transfer coefficient of a heat exchanger
8.5. Estimation of the pressure drops (losses) in the heat exchangers

Chapter 9: Thermal Power Conversion (1 Weeks)

9.1. Several examples of heat engines
9.2. Behavior of compressible fluids
9.3. Thermodynamics review
9.4. Modeling of the components of heat engines
9.5. Simulation of a thermal power plant

Laboratory Work (Using Modelica)

N.B.: It can be also performed using **Matlab/ Simscape**.

Lab 1: Introduction to Modelica and Simulation Tools

Objective: Get familiar with Modelica syntax, modeling environment (OpenModelica, Dymola, or Wolfram SystemModeler), and basic components.

- Overview of Modelica libraries (MultiBondLib, ThermoFluid, Electrical, etc.)
- Creating simple models: RC circuit, mass-spring-damper system
- Running simulations and analyzing results

Lab 2: Lumped Parameter Modeling of Mechatronic Systems

Objective: Develop and simulate mechatronic systems using lumped parameter models.

- Modeling an **electric motor** (DC motor with resistive, inductive, and back-EMF effects)
- Simulating a **power window system** (mechanical + electrical interaction)
- Analyzing system response to different input voltages

Lab 3: Setting Up a Lumped Parameter Model in Modelica

Objective: Apply the concept of adapted modeling by simplifying system dynamics.

- Identifying dominant effects in a **robotic actuator**
- Comparing **detailed vs. reduced models**
- Evaluating computational efficiency and accuracy

Lab 4: Numerical Simulation of Multiphysics Systems

Objective: Convert mathematical models into numerical simulations in Modelica.
- Implementing **ordinary differential equations (ODEs)** in Modelica
- Running simulations with different numerical solvers
- Investigating errors due to numerical approximation

Lab 5: Dynamic Performance Analysis of Robotic Systems
Objective: Use Modelica to analyze system stability and response characteristics.
- Implementing **Laplace transforms and transfer functions**
- Simulating **robotic joint dynamics** (step response, frequency response)
- Performing **first- and second-order system analysis**

Lab 6: Modeling and Simulation of Electromechanical Actuators
Objective: Investigate electromechanical power transmission in robotics.
- Simulating a **DC motor-driven robotic joint**
- Modeling a **piezoelectric actuator** for precision control
- Evaluating **torque-speed characteristics**

Lab 7: Fluid Power Transmission in Hydraulic Systems
Objective: Model hydraulic actuation systems in robotics and UAVs.
- Creating a **basic hydraulic actuator model**
- Simulating a **helicopter flight control actuator**
- Investigating **fluid compressibility effects on system response**

Lab 8: Thermal Management in Robotic Systems
Objective: Simulate heat transfer in robotic components.
- Modeling **heat dissipation in an electric motor**
- Simulating a **cooling system for UAV electronics**
- Investigating **thermal lag and steady-state temperature distribution**

Lab 9: Thermal Power Conversion and Energy Systems
Objective: Model thermodynamic processes and their application in robotics.
- Simulating a **heat engine and thermodynamic cycles**
- Modeling a **battery thermal management system for UAVs**
- Evaluating **efficiency and power losses**

Lab 10: Multiphysics Optimization and Model Validation
Objective: Optimize and validate a multiphysics system model.
- Comparing **simulation results with experimental data**
- Performing **sensitivity analysis on actuator performance**
- Optimizing a **robotic joint for minimal energy consumption**

Assessment Methods:
- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources
1. Marc Budinger, Ion Hazyuk, Clément Coïc, “Multi-Physics Modeling of Technological Systems, » ISTE Ltd, 2019, ISBN 978-1-78630-378-3.
2. Rochdi Merzouki, Arun Kumar Samantaray, Pushparaj Mani Pathak, Belkacem Ould Bouamama, “Intelligent Mechatronic Systems, ».
3. Qun Zhang, Song Cen, “Multiphysics Modeling. Numerical Methods and Engineering Applications,” Elsevier 2016.
4. Patrick Kaltjob, “Mechatronic Systems and Process Automation Model-Driven Approach and Practical Design Guidelines. » CRC Press 2018. ISBN 978-0-8153-7079-6.
5. Klaus Janschek « Mechatronic Systems Design. Methods, Models, Concepts, » springer 2012. ISBN 978-3-642-17530-5.

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.1.2	**Robot Motion Planning**	**RMP**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:

- This course provides students with both **theoretical and practical** knowledge, allowing them to develop efficient motion planning algorithms for real-world robotic applications.
- Understand the fundamental concepts of motion planning for robots.
- Master algorithms and techniques for trajectory planning.
- Apply these concepts to real-world problems in mobile robotics and robotic manipulators.
- Utilize optimization-based and artificial intelligence approaches for motion planning.

Recommended Prerequisites:

- Basic knowledge of robotics (kinematics, dynamics).
- Programming skills (Python, C++).
- Applied mathematics (linear algebra, differential calculus).

Course outline:

Chapter 1: Introduction to Motion Planning (1 Week)

- Definitions and key challenges.
- Difference between trajectory planning and path planning.
- Overview of classical and modern approaches (global/local planners).

Chapter 2: Configuration Space and State Space (2 Weeks)

- Definitions and examples.
- Transformations between different spaces.
- Obstacles and constraints in motion planning.
- Representation of obstacles and environments (Occupancy grids, Distance transform, Graph Representation, etc.).

Chapter 3: Classical Path Planning Algorithms (Search-Based Algorithms) (3 Weeks)

- **A*** Algorithm (Hybrid A*).
- Dijkstra's Algorithm.
- D* Algorithm.
- Cell decomposition methods.
- Visibility graph.
- Dubins path planner.
- Reeds-Shepp path planner.

Chapter 4: Probabilistic Methods (Sampling-Based Planning) (3 Weeks)

- PRM (Probabilistic Roadmaps).
- RRT (Rapidly-exploring Random Trees).
- Introduction to available motion planning libraries.

Chapter 5: Potential Field-Based Planning (2 Weeks)

- Attractive potential fields.
- Repulsive potential fields.
- Planning techniques.
- The local minima problem.

Chapter 6: Optimization-Based Motion Planning (3 Weeks)

- Trajectory optimization using mathematical programming.
- Variational methods in trajectory planning.
- Heuristic optimization-based motion planning.
- Optimization and machine learning-based motion planning.

Laboratory Work

These hands-on lab sessions aim to reinforce theoretical concepts through practical implementation using **Python, C++, and robotics simulation tools (ROS, Gazebo, MATLAB, V-Rep, MoveIt!)**.

Lab 1: Introduction to Motion Planning Tools

Objective: Familiarization with simulation environments and motion planning libraries.

Tasks:

- Setting up **Gazebo**, **V-Rep, MoveIt!** for robot motion planning.
- Using **OMPL (Open Motion Planning Library)** for path planning.
- Simulating a basic **joint-space movement** of a robotic manipulator.

Lab 2: Search-Based Motion Planning Algorithms

Objective: Implement and compare classical search-based algorithms.

Tasks:

- Implement **A***, **Dijkstra**, and **D*** algorithms in Python.
- Visualize paths using a **grid-based environment**.
- Compare **efficiency (computation time, optimality)** of different methods.

Lab 3: Sampling-Based Motion Planning

Objective: Implement **PRM (Probabilistic Roadmap) and RRT (Rapidly-exploring Random Trees)** for robot navigation.

Tasks:

- Implement **PRM and RRT algorithms** in Matlab.
- Test PRM and RRT in **2D obstacle environments**.
- Simulate an RRT path for a **mobile robot in Matlab/Gazebo**.

Lab 4: Potential Field-Based Motion Planning

Objective: Explore attractive and repulsive potential fields for collision avoidance.

Tasks:

- Implement **Artificial Potential Field (APF)** in Matlab.
- Test APF for **a 2D robot navigating through obstacles**.
- Analyze the **local minima problem** and propose solutions.

Lab 5: Optimization-Based Trajectory Planning

Objective: Generate smooth, optimized trajectories for robots.

Tasks:

- Implement **quadratic programming** for trajectory optimization.
- Use **Bezier curves** for smooth motion.
- Apply trajectory planning to a **robotic arm using Matlab/MoveIt!**.

Lab 6: Motion Planning for a Mobile Robot in a Realistic Environment

Objective: Develop a complete motion planning pipeline in a **realistic simulated environment**.

Tasks:

- Define a **2D map with obstacles** in **Matlab**/Gazebo.
- Implement a *hybrid planner (A + RRT)** for autonomous navigation.
- Use **LIDAR or vision sensors** for real-time obstacle detection.

Lab 7: Motion Planning for a Robotic Manipulator

Objective: Plan collision-free trajectories for a robotic arm using **Matlab/MoveIt!**.
Tasks:

- Load a **robotic arm** (**UR5/KUKA)** in **Matlab**/Gazebo.
- Plan end-effector trajectories using **inverse kinematics**.
- Simulate **pick-and-place** tasks with real-time collision avoidance.

Lab 8: AI-Based Motion Planning (Bonus Lab)
Objective: Explore reinforcement learning (RL) and neural networks for motion planning.
Tasks:

- Implement **Deep Reinforcement Learning (DQN/PPO)** for robot motion.
- Train a robot to navigate **dynamic obstacles**.
- Compare AI-based planning with classical methods.

Assessment Methods:
- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources:

- Lynch, Kevin M., and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017
- Howie Choset, Kevin Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia Kavraki,
  and Sebastian Thrun "Principles of Robot Motion: Theory, Algorithms, and Implementations," MIT Press, 2005.
- Steven M. LaValle « Planning Algorithms, » Cambridge University Press, 2006.
- Bruno Siciliano, Oussama Khatib (Eds.) "Springer Handbook of Robotics," Springer 2016. ISBN: 978-3-319-32550-7, DOI 10.1007/978-3-319-32552-1.
- Peter Corke " Robotics, Vision and Control Fundamental Algorithms in MATLAB, » springer 2017.

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.1.2	**Kinematics, Dynamics and Control of Robot Manipulators**	**KDCRM**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30		**4/4**

Course Description:
This module provides students with a comprehensive understanding of the mathematical modeling and simulation techniques used in industrial robotics. It covers the geometric, kinematic, and dynamic modeling of robotic manipulators, along with trajectory planning and optimization. Students will learn to develop and implement models using computational tools such as MATLAB, Python, and robotics simulation environments (Gazebo, V-REP, ROS, etc.). The course also introduces parallel robots and industrial robot programming languages.

Course Objectives:
By the end of this course, students will be able to:
- Understand the Fundamentals of Industrial Robotics
- Develop Geometric and Kinematic Models of Robots
- Perform Calibration and Parameter Identification
- Plan and Simulate Robot Trajectories
- Model and Simulate Robot Dynamics
- Explore Parallel Robot Modeling and Simulation
- Gain Hands-on Experience with Industrial Robot Programming

Recommended Prerequisites
1. Basic understanding of robotic systems and mechanisms.
2. Linear algebra (matrix operations, transformations, eigenvalues).
3. Differential calculus (Jacobians, partial derivatives).
4. Ordinary differential equations (ODEs) for dynamic modeling.
5. Programming skills in Python or MATLAB for implementing algorithms.
6. Newtonian mechanics and Lagrangian formulation.
7. Basics of numerical techniques.

Course outline
Chapter 1: General Introduction to Industrial Robotics (1 Week)
- History of industrial robotics
- Different types of industrial robots (articulated robots, SCARA robots, parallel robots, Cartesian robots)
- Applications in industrial robotics

Chapter 2: Geometric Modeling (1 Week)
- Geometric description of serial robots
- Forward geometric model
- Inverse geometric model

Chapter 3: Kinematic Modeling (2 Weeks)
- Forward kinematic model
- Inverse kinematic model
- Redundancy in robotic systems

- Computational implementation in Matlab/Python

Chapter 4: Identification of Geometric Parameters (1 Week)

- Geometric parameters of a robot
- Generalized differential model of a robot
- Principles of geometric calibration
- Calibration methods for geometric parameters
- Correction of geometric parameters

Chapter 5: Trajectory Planning (2 Weeks)

- Concepts of path and trajectory
- Planning in the joint space
- Planning in the operational space

Chapter 6: Dynamic Modeling (2 Week)

- Lagrange formalism
- Newton-Euler formalism
- Determination of basic inertial parameters

Chapter 7: Identification of Dynamic Parameters (1 Week)

- Principles of dynamic parameter identification
- Identification model using the dynamic model
- Sequential formulation of the dynamic model
- Practical considerations

Chapter 8: Introduction to Parallel Robot Modeling (2 Weeks)

- Classification of parallel robots
- Kinematic modeling of a parallel robot (Delta/Stewart)
- Dynamic modeling of a parallel robot

Chapter 9: Simulation and Programming of Industrial Robots (2 Week)

- Programming and simulation languages for robots (VAL 3, UniVAL drive, UniVAL plc, Stäubli Robotics Suite, Optimize Lab, V-REP, TPE from Fanuc, VAL3 from Staubli, KRL from KUKA, RAPID from ABB, etc.)
- Robot programming levels
- Offline programming systems
- Task description: learning and programming

Laboratory Work:

Each lab session will focus on a hands-on approach using **Matlab/Python** for modeling, simulation, and analysis of industrial robots.

Lab 1: Introduction to Robotics Simulation Tools

Objective: Familiarization with robot simulation environments (**Matlab, Python, V-REP, Staubli Robotics Suite, RoboDK**)

Tasks:

- Setting up a robot simulation environment
- Importing industrial robot models (e.g., UR5, KUKA, ABB)
- Running simple forward kinematics simulations

Lab 2: Geometric Modeling of Serial Robots

Objective: Implementing **forward and inverse geometric models** of a robotic arm

Tasks:

- Defining DH parameters for a 6-DOF manipulator
- Computing the forward geometric model using transformation matrices
- Implementing inverse kinematics using analytical/numerical methods

Lab 3: Kinematic Modeling & Simulation

Objective: Developing and simulating **forward and inverse kinematics** models

Tasks:

- Coding the **forward kinematic model** for a robotic arm
- Implementing the **inverse kinematics** using the Jacobian method

- Visualizing robot movement in a simulation tool

Lab 4: Calibration of Geometric Parameters
Objective: Identifying and correcting robot **geometric parameters**
Tasks:

- Extracting and analyzing robot **geometric parameters**
- Implementing **geometric calibration** using real/simulated data
- Adjusting the parameters to improve robot accuracy

Lab 5: Trajectory Planning in Joint and Operational Spaces
Objective: Implementing **trajectory generation techniques** for robotic motion
Tasks:

- Programming **linear and cubic trajectories** in joint space
- Implementing **polynomial interpolation** for smooth motion
- Simulating planned trajectories in operational space

Lab 6: Dynamic Modeling using Lagrange and Newton-Euler
Objective: Understanding and implementing **robot dynamics models**
Tasks:

- Formulating the **Lagrange equations** for a simple manipulator
- Implementing the **Newton-Euler recursive algorithm**
- Simulating dynamic behavior in Matlab/Python

Lab 7: Identification of Dynamic Parameters
Objective: Estimating **inertial and friction parameters** of a robotic arm
Tasks:

- Implementing a **parameter identification algorithm**
- Running simulations to extract dynamic parameters
- Validating the identified parameters through simulation

Lab 8: Modeling and Simulation of Parallel Robots (Delta/Stewart)
Objective: Developing **kinematic and dynamic models** for parallel robots
Tasks:

- Implementing **forward kinematics** of a Delta/Stewart robot
- Simulating **inverse kinematics** for trajectory tracking
- Modeling **dynamic behavior** using Matlab/Python

Lab 9: Robot Programming and Offline Simulation
Objective: Writing **control scripts for industrial robots**
Tasks:

- Programming robotic tasks using **KRL (KUKA), RAPID (ABB), or VAL3 (Staubli)**
- Simulating motion in **RoboDK or V-REP (CoppeliaSim)**
- Validating the robot's performance with offline programming

Lab 10: Final Project – Full Robot Simulation
Objective: Applying all learned concepts in a **realistic industrial scenario**
Tasks:

- Choosing an industrial task (e.g., pick-and-place, welding, assembly)
- Implementing kinematics, dynamics, and trajectory planning
- Running a full simulation and analyzing results

Assessment Methods:
- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources:

- John J. Craig “Introduction to Robotics. Mechanics and Control.” Fourth Edition, Pearson Education Limited
- Peter Corke “ Robotics, Vision and Control Fundamental Algorithms in MATLAB, » springer
- Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo “ Robotics - Modelling, Planning and Control,” springer
- Etienne Dombre, Wisama Khalil « Modeling, Performance Analysis and Control of Robot Manipulators, » ISTE 2006.
- Taghirad, Hamid - Parallel robots mechanics and control (2013, CRC Press)
- Kevin M. Lynch and Frank C. Park "Modern Robotics: Mechanics, Planning, and Control,", Cambridge University Press, 2017, ISBN 9781107156302.

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.1.2	**Power Electronics**	**POWE**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:
Provide students with an in-depth understanding of power semiconductor devices and power converter topologies (AC/DC, DC/DC, AC/AC, DC/AC). By the end of this module, students will be able to design, analyze, and implement power electronic circuits for embedded systems such as drones, electric vehicles, and industrial drives..

Recommended Prerequisites:
- Basic electronics (diodes, transistors, operational amplifiers)
- Fundamentals of electrical circuits (R, L, C, and simple AC analysis)
- Introductory knowledge of microcontroller-based control (PWM, gating signals)

Course outline:
Chapter 1: Introduction and Power Semiconductors (02 weeks)
- Mechanical and Solid-State Relays: fundamentals, switching characteristics
- Power Semiconductor Devices: diode, thyristor (SCR), power transistor (BJT, MOSFET, IGBT)
- Static and dynamic characteristics, safe operating areas
- Heat dissipation, packaging, and cooling methods.

Chapter 2: AC/DC Converters (Rectifiers) (03 weeks)
- Uncontrolled Rectifiers: single-phase and three-phase with R, RL, RLE load; freewheel diode usage
- Controlled Rectifiers: single-phase and three-phase with R, RL, RLE load; thyristor-based control
- Analysis of average output voltage, ripple, and power factor

Chapter 3: DC/DC Converters (Choppers) (03 weeks)
- Buck, Boost, and Buck-Boost Topologies
- Series vs. parallel choppers with various loads (R, RL, LC, RC, RLC)
- Continuous vs. discontinuous conduction mode
- PWM control strategies and efficiency considerations

Chapter 4: AC/AC Converters (Dimmers) (02 weeks)
- Single-Phase AC Voltage Controllers: R, RL, RLC loads
- Three-Phase AC Voltage Controllers: unidirectional and bidirectional dimmers
- Applications: lighting control, motor speed control (AC induction motors).

Chapter 5: DC/AC Inverters (03 weeks)
- Single-Phase Inverters: half-bridge, full-bridge with R, RL loads
- Three-Phase Inverters: 120°, 180° conduction modes, voltage control (SPWM)
- Harmonic analysis, THD, and filtering techniques.

Chapter 6: Applications and Advanced Topics (02 weeks)
- Power factor correction (PFC) basics
- Battery charging circuits and advanced converter topologies (resonant converters)
- Introduction to ESCs for brushless DC motors (BLDC), multi-level inverters

- Practical design considerations (EMI, snubbers, gate drivers).

Laboratory Work:

- **Lab 1:** Semiconductor Characterization.
- **Lab 2:** Uncontrolled and Controlled Rectifiers.
- **Lab 3:** DC/DC Buck and Boost Converters.
- **Lab 4:** AC Voltage Control (Dimmer).
- **Lab 5:** Single-Phase Inverter.
- **Lab 6:** Advanced Converter or ESC for BLDC.

Assessment Methods:

- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended textbooks and Resources

- M. H. Rashid, *Power Electronics: Circuits, Devices, and Applications*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2013.
- N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd ed. Hoboken, NJ, USA: Wiley, 2002.
- D. W. Hart, *Power Electronics*, 1st ed. New York, NY, USA: McGraw-Hill, 2010.
- B. K. Bose, Modern Power Electronics and AC Drives. Upper Saddle River, NJ, USA: Pearson, 2001.

Teaching Unit (UF)	Subject Title	Code	Semester
UEM2.1.1	**Mobile Robotics**	**MOR**	**S3**

	Lectures	TUTORIAL	LABS	Credit/ Coeff
VHH	1h30		1h00	**3/3**

Course Description:
This course provides a comprehensive introduction to the design, modeling, control, and navigation of mobile robots. It covers both theoretical foundations and practical applications, focusing on autonomous systems that operate in dynamic environments. Students will learn about robot kinematics, sensors, localization, mapping, path planning, and decision-making algorithms. The course includes hands-on lab sessions to reinforce theoretical concepts.

Course Objectives:
By the end of this course, students will be able to:

1. Understand the fundamental principles of mobile robotics, including kinematics, dynamics, and control.
2. Design and implement algorithms for robot localization, mapping, and navigation.
3. Analyze and select appropriate sensors and actuators for mobile robotic systems.
4. Develop path planning and obstacle avoidance strategies for autonomous robots.
5. Apply machine learning and AI techniques for decision-making in mobile robots.
6. Gain hands-on experience with simulation tools and real-world robotic platforms.

Recommended Prerequisites:
Introduction to Robotics (BPR), Linear Algebra, Introduction to System Dynamics and Control (ISDC), Programming in Python/C++.

Course Outline:

Chapter 1: Introduction to Mobile Robotics (1 Week)

- Overview of mobile robotics and applications
- Types of mobile robots: wheeled, legged, aerial, and underwater robots
- Challenges in mobile robotics: uncertainty, dynamic environments, and autonomy

Chapter 2: Robot Kinematics and Dynamics (2 Weeks)

- Kinematic models for wheeled robots (differential drive, Ackermann steering)
- Forward and inverse kinematics
- Dynamics of mobile robots: forces, torques, and motion constraints

Chapter 3: Sensors and Perception (2 Weeks)

- Sensor types: LiDAR, cameras, IMUs, ultrasonic, and infrared sensors
- Sensor fusion techniques
- Feature extraction and environmental perception

Chapter 4: Localization and Mapping (3 Weeks)

- Localization: Kalman filters, particle filters, and Monte Carlo localization
- Mapping: occupancy grid maps, feature-based maps
- Simultaneous Localization and Mapping (SLAM) algorithms

Chapter 5: Path Planning and Navigation (3 Weeks)

- Path planning algorithms: A*, Dijkstra, RRT, and potential fields
- Obstacle avoidance and reactive navigation
- Trajectory generation and optimization

Chapter 6: Control and Decision-Making (3 Weeks)

- Feedback control for mobile robots: PID, state-space control
- Behavior-based robotics and finite state machines
- Introduction to AI and machine learning for decision-making

Chapter 7: Applications and Emerging Trends (1 Week)

- Autonomous vehicles, drones, and service robots
- Multi-robot systems and swarm robotics
- Ethical and societal implications of mobile robotics

Laboratory Work:

1. **Robot Simulation with Gazebo and ROS**
   - Setting up a simulated robot environment
   - Basic navigation and control in simulation
2. **Sensor Integration and Data Processing**
   - Working with LiDAR, cameras, and IMUs
   - Implementing sensor fusion algorithms
3. **Localization and Mapping**
   - Building occupancy grid maps using SLAM
   - Testing localization algorithms in a simulated environment
4. **Path Planning and Navigation**
   - Implementing A* and RRT algorithms
   - Testing obstacle avoidance in dynamic environments

Assessment Methods:

- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:

- **Textbooks :**
  1. **"Introduction to Autonomous Mobile Robots"** by Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza.
  2. **"Probabilistic Robotics"** by Sebastian Thrun, Wolfram Burgard, and Dieter Fox.
  3. **"Robotics, Vision and Control: Fundamental Algorithms in MATLAB"** by Peter Corke.
  4. **"Planning Algorithms"** by Steven M. LaValle.
- **Online Courses and Tutorials:**
  - ROS (Robot Operating System) tutorials: http://wiki.ros.org
  - Coursera: "Robotics Specialization" by University of Pennsylvania
- **Software Tools:**
  - ROS (Robot Operating System)
  - Gazebo for robot simulation
  - MATLAB and Simulink for control system design
- **Open-Source Projects:**
  - TurtleBot3: An open-source mobile robot platform
  - OpenSLAM: Open-source SLAM algorithms

Teaching Unit (UF)	Subject Title	Code	Semester
UEM2.1.1	**Advanced and Current Trends in Artificial Intelligence**	**ACTAI**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h00	**2/2**

Course Objectives:

- Understand key advancements in deep learning, recurrent networks, and transformers.
- Explore large language models (LLMs) and their transition to smaller models.
- Learn practical implementation using TensorFlow, Keras, and PyTorch.
- Investigate AI applications in robotics and smart manufacturing.
- Study advanced topics such as Vision-Language Models (VLM) and federated learning.

Course Outline

Chapter 1: Introduction to Deep Learning (4h)

- Fundamentals of deep learning and neural networks.
- Backpropagation, activation functions, and optimization techniques.
- Introduction to TensorFlow, Keras, and PyTorch for deep learning.

Chapter 2: Recurrent Neural Networks (LSTM & GRU) (4h)

- Introduction to sequence modeling and time-series data.
- Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures.
- Applications in speech recognition, robotics, and predictive maintenance.

Chapter 3: Transformer Architecture and Large Language Models (5h)

- Introduction to the Transformer model: self-attention mechanism.
- Comparison with RNN-based models (advantages and limitations).
- Introduction to Large Language Models (LLMs) (GPT, BERT, T5, etc.).

Chapter 4: From Large to Small Language Models (5h)

- Model distillation and parameter-efficient tuning (LoRA, adapters).
- Techniques for optimizing LLMs for embedded and edge AI.
- Applications in robotics and industrial automation.

Chapter 5: RAG Techniques vs Fine-Tuning (4h)

- Retrieval-Augmented Generation (RAG) vs traditional fine-tuning.
- Trade-offs in efficiency, accuracy, and adaptability.
- Case studies in robotics and manufacturing AI applications.

Chapter 6: AI Development Ecosystem and Frameworks (4h)

- Overview of AI frameworks: TensorFlow, PyTorch, Hugging Face, LangChain.
- MLOps: Deployment and scaling of AI models.
- Cloud-based vs on-device AI solutions.

Chapter 7: Advanced Topics in AI (4h)

- Vision-Language Models (VLM) and multimodal AI.
- Federated learning for privacy-preserving AI.
- Trends in robotics AI optimization (autoML, meta-learning).
- Future perspectives in AI for robotics and smart manufacturing.

Laboratory Work

Lab 1: Implementing a basic neural network using TensorFlow/Keras.

Lab 2: Implementing LSTM and GRU networks in Python.
Lab 3: Implementing a basic Transformer model using PyTorch.
Lab 4: Fine-tuning a small language model with Hugging Face Transformers.
Lab 5: Implementing a simple RAG-based chatbot.
Lab 6: Deploying an AI model using TensorFlow Serving.

Assessment Methods:
- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:
1. "Artificial Intelligence: A Modern Approach", Stuart Russell, Peter Norvig, Pearson, 4th Edition, 2020.
2. "Deep Learning", Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press, 1st Edition, 2016.
3. "Reinforcement Learning: An Introduction", Richard S. Sutton, Andrew G. Barto, MIT Press, 2nd Edition, 2018.
4. "Pattern Recognition and Machine Learning", Christopher M. Bishop, Springer, 1st Edition, 2006.
5. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", Aurélien Géron, O'Reilly Media, 2nd Edition, 2019.
6. "Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play", David Foster, O'Reilly Media, 1st Edition, 2019.

Teaching Unit (UF)	Subject Title	Code	Semester
UEM2.1.1	**Robot Operating System & Computer Tools for Robotics**	**ROSCTR**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h30	**2/2**

Course Description:
This course provides a comprehensive introduction to the **Robot Operating System (ROS)**, a widely used framework for developing robotic applications. Students will learn the core concepts of ROS, including nodes, topics, messages, services, and actions, and gain hands-on experience through practical lab exercises. The course also covers simulation tools like Gazebo and visualization tools like RViz, enabling students to build and test robotic systems in a simulated environment.

Course Objectives:
- Understand the architecture and core concepts of ROS.
- Learn to create and manage ROS packages, nodes, and messages.
- Gain proficiency in using ROS tools for simulation, visualization, and debugging.
- Develop the ability to build and test robotic applications using ROS.
- Apply ROS to real-world robotics problems.

Recommended Prerequisites
- Basic knowledge of programming (Python and/or C++).
- Familiarity with Linux operating systems.
- Understanding of robotics fundamentals (kinematics, sensors, and control).

Course Outline:
Chapter 1: Introduction to ROS (1 week)
- Overview of ROS and its ecosystem.
- Key concepts: nodes, topics, messages, services, and actions.
- Installing and setting up ROS on Ubuntu.

Chapter 2: ROS Basics (1 week)
- Creating and managing ROS packages.
- Writing and running simple ROS nodes.
- Understanding ROS topics and messages.

Chapter 3: ROS Communication (1 week)
- Publisher-subscriber model.
- Custom message types.
- Hands-on: Implementing a simple publisher-subscriber system.

Chapter 4: ROS Services and Actions (1 week)
- Service-client model.
- Actionlib for complex tasks.
- Hands-on: Creating and using ROS services and actions.

Chapter 5: ROS Tools (1 week)
- Introduction to ROS tools: roscore, rosrun, roslaunch.
- Using RViz for visualization.
- Hands-on: Visualizing sensor data in RViz.

Chapter 6: ROS Simulation with Gazebo (1 week)
- Introduction to Gazebo and its integration with ROS.

- Creating and simulating a robot model in Gazebo.
- Hands-on: Simulating a mobile robot in Gazebo.

Chapter 7: ROS Navigation Stack (1 week)

- Overview of the ROS navigation stack.
- Mapping, localization, and path planning.
- Hands-on: Implementing autonomous navigation for a simulated robot.

Chapter 8: ROS Control (2 weeks)

- Introduction to ROS control packages.
- Controlling robotic arms and mobile robots.
- Hands-on: Controlling a robotic arm using ROS.

Chapter 9: ROS and Sensors (2 weeks)

- Integrating sensors (e.g., cameras, LiDAR) with ROS.
- Processing sensor data using ROS.
- Hands-on: Implementing object detection using a camera in ROS.

Chapter 10: ROS and Machine Learning (1 week)

- Integrating machine learning models with ROS.
- Using ROS for real-time inference.
- Hands-on: Implementing a simple ML-based object recognition system.

Chapter 11: Advanced ROS Topics (1 week)

- ROS 2: Differences and migration from ROS 1.
- Multi-robot systems and ROS.
- Hands-on: Simulating a multi-robot system.

Laboratory Work

Lab 1: Setting Up ROS

- **Objective:** Install and configure ROS on Ubuntu.
- **Tasks:**
  - Install ROS and necessary packages.
  - Set up the ROS workspace.
  - Verify installation by running a simple ROS node.

Lab 2: Creating ROS Packages and Nodes

- **Objective:** Learn to create and manage ROS packages and nodes.
- **Tasks:**
  - Create a ROS package.
  - Write and run a simple ROS node.
  - Understand the structure of a ROS package.

Lab 3: Publisher-Subscriber Model

- **Objective:** Implement a basic publisher-subscriber system.
- **Tasks:**
  - Create a publisher node to send messages.
  - Create a subscriber node to receive messages.
  - Visualize the communication using rostopic.

Lab 4: Custom Messages and Services

- **Objective:** Create and use custom ROS messages and services.
- **Tasks:**
  - Define a custom message type.
  - Implement a ROS service and client.
  - Test the service-client interaction.

Lab 5: Using RViz for Visualization

- **Objective:** Learn to visualize data in RViz.
- **Tasks:**
  - Publish sensor data (e.g., laser scan, camera image) to ROS topics.
  - Visualize the data in RViz.

- Configure RViz displays for different data types.

Lab 6: Simulating a Robot in Gazebo

- **Objective:** Simulate a robot model in Gazebo.
- **Tasks:**
  - Create a URDF (Unified Robot Description Format) model.
  - Load the model in Gazebo.
  - Control the robot using ROS topics.

Lab 7: Autonomous Navigation with ROS

- **Objective:** Implement autonomous navigation for a simulated robot.
- **Tasks:**
  - Set up the ROS navigation stack.
  - Create a map using SLAM.
  - Perform path planning and obstacle avoidance.

Lab 8: Controlling a Robotic Arm

- **Objective:** Control a robotic arm using ROS.
- **Tasks:**
  - Simulate a robotic arm in Gazebo.
  - Implement joint control using ROS control packages.
  - Perform simple pick-and-place tasks.

Lab 9: Integrating Sensors with ROS

- **Objective:** Integrate and process sensor data in ROS.
- **Tasks:**
  - Simulate a camera or LiDAR sensor in Gazebo.
  - Process sensor data using ROS nodes.
  - Visualize the processed data in RViz.

Lab 10: Machine Learning with ROS

- **Objective:** Integrate a machine learning model with ROS.
- **Tasks:**
  - Train a simple object detection model.
  - Deploy the model in ROS for real-time inference.
  - Visualize the results in RViz.

Lab 11: Multi-Robot Simulation

- **Objective:** Simulate a multi-robot system in ROS.
- **Tasks:**
  - Set up multiple robots in Gazebo.
  - Implement coordination and communication between robots.
  - Visualize the multi-robot system in RViz.

Assessment Methods:

- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:

1. Morgan Quigley, Brian Gerkey, and William D. Smart."Programming Robots with ROS: A Practical Introduction to the Robot Operating System "
2. Gang Peng, Tin Lun LAM, Chunxu Hu, Yu Yao, Jintao Liu, Fan Yang , “Introduction to Intelligent Robot System Design. Application Development with ROS, » Springer (2023)
3. Lentin Joseph, Jonathan Cacace « Mastering ROS for Robotics Programming, Best practices and troubleshooting solutions when working with ROS, » (2021, Packt Publishing)
4. Carol Fairchild and Thomas L. Harman "ROS Robotics by Example".
5. Official ROS documentation: http://wiki.ros.org
6. ROS tutorials and courses on platforms like Udemy and Coursera.

Teaching Unit (UF)	Subject Title	Code	Semester
UET2.1.1	**Marketing and Economy**	**MARE**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30			**2/2**

Course Objectives:

1. To introduce students to fundamental economic principles and their application in engineering decision-making.
2. To develop an understanding of marketing strategies and their role in product development and commercialization.
3. To equip students with the skills to analyze market trends, consumer behavior, and economic environments.
4. To enable students to integrate marketing and economic insights into engineering design and innovation processes.

Recommended Prerequisites:

Basic understanding of mathematics, statistics, and introductory business concepts.

Learning Outcomes:

By the end of the course, students will be able to:

1. Analyze economic environments and their impact on engineering projects.
2. Develop marketing strategies for engineering products and services.
3. Conduct market research and interpret data to inform engineering decisions.
4. Integrate economic and marketing insights into the design and commercialization of engineering solutions.

Course Outline:

Chapter 1: Introduction to Economics for Engineers (3 Weeks)

- **1.1 Basic Economic Concepts**
  - Scarcity, choice, and opportunity cost
  - Supply and demand
  - Market equilibrium
- **1.2 Microeconomics**
  - Consumer behavior and utility theory
  - Production and cost analysis
  - Market structures (perfect competition, monopoly, oligopoly)
- **1.3 Macroeconomics**
  - GDP, inflation, and unemployment
  - Fiscal and monetary policies
  - Economic growth and development

Chapter 2: Marketing Fundamentals (3 Weeks)

- **2.1 Introduction to Marketing**
  - Definition and scope of marketing
  - Marketing mix (4Ps: Product, Price, Place, Promotion)
- **2.2 Consumer Behavior**

- Factors influencing consumer decisions
- Buyer decision-making process

- **2.3 Market Research**
  - Qualitative and quantitative research methods
  - Data collection and analysis techniques

Chapter 3: Strategic Marketing in Engineering (3 Weeks)

- **3.1 Product Development and Lifecycle**
  - Stages of product development
  - Managing product lifecycle in engineering industries
- **3.2 Pricing Strategies**
  - Cost-based, value-based, and competition-based pricing
  - Pricing for innovative engineering products
- **3.3 Distribution and Supply Chain Management**
  - Channels of distribution
  - Logistics and supply chain optimization

Chapter 4: Economic Analysis for Engineering Projects (3 Weeks)

- **4.1 Cost-Benefit Analysis**
  - Evaluating engineering projects from an economic perspective
  - Discounted cash flow and net present value (NPV)
- **4.2 Risk and Uncertainty in Economic Decision-Making**
  - Sensitivity analysis
  - Scenario planning
- **4.3 Global Economic Environment**
  - International trade and globalization
  - Impact of economic policies on engineering industries

Chapter 5: Integrating Marketing and Economics in Engineering (3 Weeks)

- **5.1 Innovation and Commercialization**
  - Role of marketing in engineering innovation
  - Intellectual property and patents
- **5.2 Case Studies**
  - Real-world examples of successful integration of marketing and economics in engineering projects
- **5.3 Future Trends**
  - Digital marketing and e-commerce
  - Sustainable engineering and green marketing

Assessment Methods:
- Final exam (100%).

Recommended Textbooks and Resources:

1. **Principles of Economics** by N. Gregory Mankiw
2. **Marketing Management** by Philip Kotler and Kevin Lane Keller
3. **Engineering Economics** by William G. Sullivan, Elin M. Wicks, and C. Patrick Koelling
4. **Harvard Business Review** case studies on marketing and innovation
5. Online resources: Coursera, Khan Academy, and industry reports.

Teaching Unit (UF)	Subject Title	Code	Semester
UED2.1.1	**Sensors for Autonomous Systems**	**SNAS**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h00			**1/1**

Course Description:
This course provides a comprehensive understanding of the sensors used in autonomous systems, including their principles, design, integration, and applications. Students will learn how sensors enable perception, localization, navigation, and decision-making in autonomous systems such as self-driving cars, drones, and robotic platforms. The course combines theoretical knowledge with hands-on laboratory exercises and projects to prepare students for real-world challenges in robotics and autonomous systems design.

Course Objectives:
By the end of this course, students will be able to:

1. Understand the fundamental principles of sensors and their role in autonomous systems.
2. Analyze and compare different types of sensors (e.g., LiDAR, radar, cameras, IMUs, ultrasonic, etc.) and their applications.
3. Design and integrate sensor systems for autonomous robots and vehicles.
4. Process and interpret sensor data for perception, localization, and mapping.
5. Evaluate the performance, limitations, and trade-offs of sensors in real-world scenarios.
6. Develop algorithms for sensor fusion and multi-sensor data integration.
7. Apply sensor knowledge to solve practical problems in autonomous systems design.

Recommended Prerequisites:
Introduction to Robotics, Electronics and Signal Processing, Control Systems**.**

Course Outline:
Chapter 1: Introduction to Sensors for Autonomous Systems (2 weeks)

- Overview of autonomous systems and their sensor requirements
- Sensor classification: active vs. passive, exteroceptive vs. proprioceptive
- Key performance metrics: accuracy, resolution, range, latency, and reliability

Chapter 2: Sensor Technologies (4 weeks)

- **Vision Sensors:** Cameras (monocular, stereo, RGB-D), principles of computer vision
- **LiDAR:** Principles, 2D vs. 3D LiDAR, applications in mapping and obstacle detection
- **Radar:** Principles, Doppler effect, applications in automotive systems
- **Inertial Sensors:** IMUs (accelerometers, gyroscopes), dead reckoning
- **Ultrasonic Sensors:** Principles, applications in proximity sensing
- **GNSS (Global Navigation Satellite Systems):** GPS, RTK-GPS, and their role in localization

Chapter 3: Sensor Data Processing (4 weeks)

- Signal processing basics: noise filtering, signal conditioning
- Feature extraction and object detection using sensor data
- Introduction to sensor calibration and synchronization

Chapter 4: Sensor Fusion and Localization (2 weeks)

- Introduction to sensor fusion techniques: Kalman filters, particle filters
- Multi-sensor data integration for robust perception
- Localization techniques: SLAM (Simultaneous Localization and Mapping), odometry

Chapter 5: Applications of Sensors in Autonomous Systems (2 weeks)

- Autonomous vehicles: perception, navigation, and control
- Drones: obstacle avoidance, terrain mapping
- Industrial robots: object detection, precision manipulation
- Emerging trends: bio-inspired sensors, quantum sensors

Chapter 6: Challenges and Future Directions (1 week)

- Limitations of current sensor technologies
- Ethical and safety considerations in sensor design
- Future trends: AI-driven sensor systems, miniaturization, and energy efficiency

Assessment Methods:

- Final exam (100%).

Recommended Textbooks and Resources:

1. "Sensors for Mobile Robots" by H.R. Everett
2. "Introduction to Autonomous Mobile Robots" by Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza
3. "Probabilistic Robotics" by Sebastian Thrun, Wolfram Burgard, and Dieter Fox
4. "Computer Vision: Algorithms and Applications" by Richard Szeliski

Teaching Unit (UF)	Subject Title	Code	Semester
UED2.1.1	**Capstone Project II**	**CAP-II**	**S3**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH			1h30	**1/1**

Course Objectives:

- Develop an initial prototype.
- Conduct iterative testing and debugging.
- Perform a mid-project review.

Targeted Skills

- Hands-on prototype development.
- Software/hardware debugging.
- Performance evaluation and improvement.

Recommended prerequisites

- Embedded systems and control strategies.
- Robotics principles
- Prototyping tools and software.
- Testing methodologies.

Course Outline

1. **Prototype Development (5 weeks)**
    - Hardware and software integration.
    - Initial assembly and testing.
2. **Iterative Testing and Debugging (5 weeks)**
    - Identifying and resolving system issues.
    - Performance tuning and optimization.
3. **Mid-Project Review and Demonstration (5 weeks)**
    - Presenting progress to faculty.
    - Gathering feedback and making improvements.

Recommended textbooks and Resources

1. Rapid Prototyping: Principles and Applications by Chua C.K.
2. Debugging: The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems by David Agans

Detailed programs for the eighth semester

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.2.1	**Industrial Automation**	**IAUT**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:
This 15-week program emphasizes a logical progression, starting with the basics of PLCs, moving through HMIs and field networks, and concluding with SCADA supervision. It is designed to provide students with comprehensive and practical training in industrial automation.

- Deepen the programming of complex functions and the management of input/output (I/O).
- Master programming tools and development of automated projects.
- Understand and implement information exchange between PLCs, HMIs, and smart devices via field networks.
- Acquire skills in industrial supervision.

Recommended Prerequisites
• Basic knowledge of PLCs (Programmable Logic Controllers).
• Combinatorial and sequential logic.
• Fundamentals of sensors and actuators.

Targeted Skills
Upon completion of this course, students will be able to:

1. Program industrial PLCs for complex applications.
2. Configure and implement field networks for data exchange between PLCs and smart devices.
3. Design and integrate HMIs with PLCs.
4. Understand and use SCADA systems for managing automated systems.

Course outline
Chapter 1: Overview of Automated Systems (1 week)

- Concepts of automated systems.
- Hardware and software architecture of an automated system.
- Practical examples of automated systems.
- Transition from wired logic to programmed logic.

Chapter 2: Programmable Logic Controllers (PLCs) (2 weeks)

- Definition and role of a PLC.
- Different types of PLCs.
- Components of a PLC.
- Criteria for selecting a PLC.
- Data types in PLCs.
- I/O modules:
  - Digital I/O modules.
  - Analog I/O modules.
- Communication modules.

- Function modules:
    - PID control modules.
    - Motion control modules.
    - High-speed counter modules.

Chapter 3: PLC Programming (6 weeks)

- Introduction to combinatorial logic.
- Logical equations and logic gates.
- Introduction to Grafcet (Sequential Function Chart).
- Translation of Grafcet to Ladder logic.
- Translating specifications into Grafcet.
- Overview of PLC's programming languages (Ladder, Function Block Diagram, STL, and SCL).

Chapter 4: Field Networks for PLCs (3 weeks)

- Role and importance of industrial communication networks.
- Characteristics of networks:
    - Standardisations.
    - Transmission media: twisted pair, coaxial cable, fiber optics.
    - Transmissions standards: 4-20 mA, RS232, RS422/485, etc.
- Network principles: topologies, access methods, protocols.
- Field-level networks:
    - TELWAY7, FIPWAY/FIPIO.
    - MODBUS PLUS.
    - PROFIBUS DP.
- Field-level networks (continued):
    - ASI (Actuator Sensor Interface).
    - DEVICE NET, ETHERNET.
- Selection and implementation of networks:
    - Decomposition of an automation system into subsystems.
    - Synchronization of subsystems.
    - Heterogeneous networks and communication modules.
    - Gateways between different types of networks.

Chapter 5: Human-Machine Interfaces (HMIs) (2 weeks)

- Role and importance of HMIs in automated systems.
- Designing user interfaces for PLCs.
- Integration of HMIs with PLCs.
- Practical examples of HMI-PLC communication.

Chapter 6: SCADA Systems (1 week)

- Purpose and importance of industrial supervision.
- Overview of SCADA software.
- Integration of HMIs and field networks into a SCADA system.
- Case study: Application to a sample project.

Laboratory Work:

- **Lab 1:** PLC Configuration
- **Lab 2:** Implementation of PLC Programs
- **Lab 3:** Implementation of Communication Between PLCs
- **Lab 4:** HMI Configuration and Integration with PLCs
- **Lab 5:** Development of Navigation Hierarchy and Object Dynamization in Views

Assessment Methods:

- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources:

Textbooks

1. **“**Automation, Production Systems, and Computer-Integrated Manufacturing”, Mikell P. Groover, Pearson, 2020 (5th Edition).
2. **“**Programmable Logic Controllers”, Frank D. Petruzella, McGraw-Hill Education, 2020 (6th Edition).
3. “Industrial Automation: Hands-On”, Frank Lamb, McGraw-Hill Education, 2013.
4. “Control Systems Engineering”, Norman S. Nise, Wiley, 2020 (8th Edition).
5. **“**SCADA: Supervisory Control and Data Acquisition”, Stuart A. Boyer, ISA (International Society of Automation), 2009 (4th Edition).
6. “Industrial Robotics: Theory, Modelling and Control”, Sam Cubero**,** Pro Literature Verlag, 2006.

Websites

1. **PLC Academy**
   - Website: plcacademy.com
2. **RealPars**
   - Website: realpars.com
3. **ISA (International Society of Automation)**
   - Website: isa.org
4. **Coursera**
   - Website: coursera.org
   - Focus: Online courses on industrial automation, PLCs, and control systems.
5. **Festo Didactic**
   - Website: festo-didactic.com

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.2.1	**Industrial Robots Control**	**IRC**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h00	**2/2**

Course Description:
This course provides an in-depth understanding of the control systems used in industrial robots. It covers the theoretical foundations, practical implementations, and advanced techniques for controlling robotic manipulators. The course emphasizes the integration of control theory with real-world applications in industrial automation.

Course Objectives:
By the end of this course, students will be able to:

1. Understand the fundamental principles of industrial robot control systems.
2. Design and implement control algorithms for robotic manipulators.
3. Analyze and optimize the performance of robotic control systems.
4. Apply advanced control techniques such as adaptive control, robust control, and optimal control in robotics.
5. Develop and simulate control systems using industry-standard software tools.
6. Understand the integration of sensors and actuators in robotic control systems.
7. Evaluate the safety and reliability of industrial robot control systems.

Course Outcomes:
Upon successful completion of this course, students will have a solid foundation in the control of industrial robots, enabling them to design and implement control systems for a wide range of robotic applications. They will also gain practical experience with industry-standard tools and techniques, preparing them for careers in robotics and autonomous systems design.

Course Outline:

Chapter 1: Introduction to Industrial Robots and Control Systems (2 weeks)

- Overview of industrial robots and their applications
- Basic components of robotic systems: actuators, sensors, controllers
- Introduction to control systems: open-loop vs. closed-loop control
- Mathematical modeling of robotic systems

Chapter 2: Classical Control Techniques (3 weeks)

- PID control: theory and tuning methods
- State-space representation and analysis
- Frequency response and Bode plots

Chapter 3: Advanced Control Techniques (3 weeks)

- Adaptive control: model reference adaptive control (MRAC)
- Robust control: H-infinity control
- Optimal control: Linear Quadratic Regulator (LQR)

Chapter 4: Sensor Integration and Feedback Control (2 weeks)

- Types of sensors used in robotics: encoders, force/torque sensors, vision systems
- Sensor fusion techniques
- Feedback control: state feedback, output feedback

Chapter 5: Introduction to Collaborative Robotics (1 week)

- Definition and evolution of collaborative robotics (cobots).
- Differences between traditional industrial robots and cobots.

Chapter 6: Introduction to Collaborative Robotics (2 weeks)

- Task sharing: Cobots for repetitive, dangerous, or precision tasks.
- Applications: Assembly, material handling, machine tending, welding, and quality inspection.
- Integration with existing industrial robot control systems.

Chapter 7: Safety and Reliability in Industrial Robots (1 week)

- Safety standards and regulations (ISO 10218, ISO/TS 15066)
- Reliability analysis: fault detection and diagnosis
- Redundancy and fail-safe mechanisms

Laboratory Work :

- **Lab 1:** Introduction to simulation tools (MATLAB/Simulink, ROS)
- **Lab 2:** Implementation of PID control on a robotic arm
- **Lab 3:** Simulation of advanced control techniques using MATLAB/Simulink
- **Lab 4:** Integration of sensors and implementation of feedback control
- **Lab 5:** Collaborative robot programming and task execution

Assessment Methods:

- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:

- **Textbooks**

1. "Robot Modeling and Control" by Mark W. Spong, Seth Hutchinson, and M. Vidyasagar.
2. "Introduction to Robotics: Mechanics and Control" by John J. Craig.
3. "Modern Control Engineering" by Katsuhiko Ogata.
4. "Robotics, Vision and Control: Fundamental Algorithms in MATLAB" by Peter Corke.

- **Online Courses:**
  - **Coursera:** "Robotics Specialization" by University of Pennsylvania
  - **edX:** "Robotics MicroMasters" by University of Pennsylvania
- **Software Tools:**
  - **MATLAB/Simulink**
  - **Robot Operating System (ROS)**
  - **Gazebo Simulator**

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.2.1	**Internet Of Things (IOT),**	**IOT**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h00	**4/4**

Course Objectives:
This Lecturese explores the foundational concepts of IoT with a focus on their application in autonomous embedded systems. Students will learn IoT architectures, protocols, data storage techniques, and deployment strategies for low-power, efficient, and reliable systems in real-world applications like drones, robots, and smart devices.

Recommended Prerequisites:
- Basic understanding of computer systems and programming.

Course outline:
Chapter 1: Introduction to IoT
- Definitions and key concepts of IoT.
- Components of an IoT system: sensors, actuators, gateways, servers.
- Challenges: energy consumption, latency, reliability.
- Applications in autonomous systems (e.g., drones, Industry 4.0).

Chapter 2: IoT Network Protocols
- MQTT (Message Queuing Telemetry Transport): Publish/Subscribe model, QoS.
- CoAP (Constrained Application Protocol): Lightweight alternative to HTTP.
- HTLABS& WebSockets: Use cases in IoT.
- LPWAN protocols: LoRaWAN and NB-IoT for long-range, low-power networks..

Chapter 3: IoT Architecture & Edge Processing
- Client-Server vs Publish/Subscribe models.
- IoT gateways as Edge nodes: preprocessing, data filtering, and aggregation.
- LPWAN networks: LoRaWAN, Sigfox, NB-IoT.
- Basic Edge Computing: energy-efficient data processing before Cloud transmission.

Chapter 4: IoT Data Storage & Visualization
- Data formats: JSON, CSV, lightweight alternatives.
- Preprocessing for storage: filtering, aggregation, and lightweight computing.
- Local storage: SQLite and InfluxDB for embedded devices.
- Introduction to Node-RED for IoT data orchestration and visualization.

Chapter 5: IoT Deployment & Power Optimization
- Interfacing ESP32/Raspberry Pi with sensors: DHT22, gas sensors, IMUs...
- Data transmission via MQTT or HTTP.
- Power-saving techniques: Deep Sleep, wake-up on interrupt.

Laboratory Work:
- **Lab 1**: MQTT communication between ESP32 and a server
- **Lab 2**: Edge processing for IoT data aggregation.
- **Lab 3**: SQLite and Node-RED for data storage and visualization.

- **Lab 4**: Sensor interfacing and power-saving implementation on ESP32.
- **Lab 5**: Designing and deploying a functional IoT system.

Assessment Methods:
- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:
1. J. Holler, V. Tsiatsis, C. Mulligan, S. Avesand, S. Karnouskos, and D. Boyle, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Amsterdam, Netherlands: Elsevier, 2014.
2. R. Minerva, A. Biru, and D. Rotondi, *Towards a Definition of the Internet of Things (IoT)*. IEEE Internet Initiative, 2015.
3. D. Uckelmann, M. Harrison, and F. Michahelles, *Architecting the Internet of Things*. Berlin, Germany: Springer, 2011.
4. O. Hersent, D. Boswarthick, and O. Elloumi, *The Internet of Things: Key Applications and Protocols*. Hoboken, NJ, USA: Wiley, 2012.

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.2.2	**UAV Flight Testing & Operations**	**UAVFTO**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h30	**4/4**

Course Overview:
This course provides a comprehensive introduction to Unmanned Aerial Vehicles (UAVs), covering their design, operation, and applications. Students will learn about the history, aerodynamics, performance, stability, propulsion, and mission planning of UAVs, as well as their payloads and control systems. The course combines theoretical knowledge with practical insights, preparing students for careers in UAV design, operation, and research.

Course Objectives:
- Understand the history, classification, and missions of UAVs.
- Learn the principles of aerodynamics and performance for UAVs.
- Gain knowledge of flight stability, control, and propulsion systems.
- Explore UAV structures, mission planning, and control stations.
- Understand the integration and control of reconnaissance/surveillance payloads.
- Apply theoretical knowledge to real-world UAV applications.

Recommended Prerequisites
1. Mathematics: Linear algebra, calculus, and probability/statistics.
2. Physics: Classical mechanics and fluid mechanics.
3. Programming: Python or MATLAB (essential), C++ (optional).
4. Engineering Fundamentals: Basic electronics and control systems.
5. Aerospace Basics (Optional): Aerodynamics and structural mechanics.
6. Soft Skills: Problem-solving and teamwork.

Course Outline:
Chapter 1: Introduction to UAVs: History and Overview (1 week)
- Evolution of UAVs: From early developments to modern applications.
- Key milestones in UAV technology.
- Overview of UAV applications: military, commercial, and civilian.

Chapter 2: Classes and Missions of UAVs (1 week)
- Classification of UAVs: Fixed-wing, rotary-wing, and hybrid UAVs….
- Mission types: reconnaissance, surveillance, delivery, and mapping…
- Case studies: Examples of UAV missions in different industries.

Chapter 3: Aerodynamics (2 weeks)
- Fundamental principles of aerodynamics: Lift, drag, thrust, and weight.
- Aerodynamic forces and moments on UAVs.
- Wing design and performance: Airfoils, aspect ratio, and wing loading.

Chapter 4: Performance (1 week)
- UAV performance metrics: Range, endurance, speed, and altitude.
- Cruising Flight, Climbing Flight, Gliding Flight, Launch, Recovery
- Factors affecting performance: Weather, payload, and propulsion.
- Performance calculations and trade-offs.

Chapter 5: Flight Stability and Control (3 weeks)

- Stability principles: Static and dynamic stability.
- Control surfaces: Ailerons, elevators, and rudders.
- Trim, Stability, Control
- Autopilot systems and flight control algorithms.

Chapter 6: Propulsion (1 week)

- Types of propulsion systems: Electric, internal combustion, and hybrid.
- Propeller and engine performance.
- Thrust Generation, Powered Lift, Sources of Power, Sources of Electric Energy, Power and Thrust

Chapter 7: Air Vehicle Structures (1 week)

- Materials used in UAV construction: Composites, metals, and polymers.
- Structural design considerations: Weight, strength, and durability.
- Stress analysis and failure modes.

Chapter 8: Mission Planning and Control Station MPCS (2 weeks)

- Mission planning process: Pre-flight checks, route planning, and contingency planning.
- MPCS: Hardware, software components, Configuration, Interfaces, Architecture .
- Communication systems: Line-of-sight and beyond-line-of-sight.

Chapter 9: Control of Air Vehicle and Payload (1 week)

- Flight control systems: Manual, semi-autonomous, and autonomous modes.
- Payload integration: Sensors, cameras, and communication devices.
- Real-time monitoring and control of payloads.

Chapter 10: Reconnaissance/Surveillance Payloads (1 week)

- Types of payloads: Electro-optical, infrared, and LiDAR.
- Data acquisition and processing.
- Applications of reconnaissance/surveillance payloads: Agriculture, disaster management, and security.

Chapter 11: Advanced Topics in UAVs (1 week)

- Swarm robotics: Coordination and communication in multi-UAV systems.
- AI and machine learning for UAVs.
- Emerging trends: Urban air mobility and autonomous delivery.

Laboratory Work

Lab 1: Introduction to UAV Simulation

- Objective: Familiarize students with UAV simulation tools.
- Tasks:
  - Set up a UAV simulation environment (e.g., Gazebo, X-Plane).
  - Simulate basic UAV maneuvers.

Lab 2: Aerodynamic Analysis

- Objective: Analyze aerodynamic forces on a UAV.
- Tasks:
  - Use simulation tools to study lift and drag forces.
  - Perform wing design optimization.

Lab 3: Flight Stability and Control

- Objective: Implement and test flight control algorithms.
- Tasks:
  - Simulate UAV stability and control using an autopilot system.
  - Test control surfaces and their impact on flight dynamics.

Lab 4: Mission Planning

- Objective: Plan a UAV mission.
- Tasks:
  - Use mission planning software to create a flight plan.

- Simulate the mission in a controlled environment.

Lab 5: Payload Integration

- Objective: Integrate and control a payload.
- Tasks:
  - Attach a camera or sensor to a UAV.
  - Test data acquisition and transmission.

Lab 6: Reconnaissance/Surveillance Payload Testing

- Objective: Test reconnaissance/surveillance payloads.
- Tasks:
  - Simulate data collection using electro-optical or infrared sensors.
  - Process and analyze the collected data.

Assessment Methods:

- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources:

- Textbook: "Introduction to UAV Systems" by Paul G. Fahlstrom and Thomas J. Gleason.
- "Small Unmanned Aircraft: Theory and Practice" by Randal W. Beard and Timothy W. McLain.
- "Unmanned Aircraft Systems: UAVS Design, Development, and Deployment" by Reg Austin.

Teaching Unit (UF)	Subject Title	Code	Semester
UEF2.2.2	**Introduction to Materials & Mechanical Manufacturing Processes**	**IMMMP**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h30	**4/4**

Course Objectives:

- Understand the properties of the four categories of materials (metallic, ceramic, polymeric, and composites).
- Master the techniques of shaping and processing these materials.
- Learn characterization methods and degradation mechanisms.
- Apply this knowledge to practical cases in the construction of electronic-mechanical devices.

Course outline

Chapter 1: Introduction to Materials (1 week)

- Definition of the four categories of materials: metallic, ceramic, polymeric, and composites.
- Role of materials in electronic-mechanical devices (sensors, actuators, microsystems, etc.).
- Examples of applications for each category of materials.

Chapter 2: Properties of Materials (2 weeks)

- **Mechanical properties**: elasticity, plasticity, strength, hardness, toughness.
- **Electrical properties**: conductivity, resistivity, piezoelectricity, semiconductivity.
- **Thermal properties**: thermal conductivity, thermal expansion, heat resistance.
- **Magnetic properties**: ferromagnetism, magnetostriction.
- Comparison of properties between the four categories of materials.

Chapter 3: Shaping and Processing of Materials (2 weeks)

- **Metallic materials**: casting, machining, heat treatment, thin-film deposition.
- **Ceramic materials**: sintering, molding, deposition techniques (CVD, PVD).
- **Polymeric materials**: injection molding, extrusion, 3D printing.
- **Composite materials**: layering, molding, reinforcement techniques (carbon fibers, glass).
- **Practical examples:** fabrication of silicon wafers, thin-film deposition for sensors.

Chapter 4: Characterization of Materials (3 weeks)

- **Microscopic methods**: electron microscopy (SEM, TEM), atomic force microscopy (AFM).
- **Macroscopic methods**: mechanical testing (tensile, compression), electrical testing (resistivity measurement), thermal testing (DSC, TGA).
- **Analysis of results**: data interpretation, defect identification.
- Comparison of characterization techniques for each category of materials.

Chapter 5: Degradation and Durability of Materials (3 weeks)

- **Degradation mechanisms**: corrosion, fatigue, wear, ion migration, aging.
- **Impact of environmental conditions**: temperature, humidity, radiation, mechanical stress.

- **Protection methods**: coatings, surface treatments, selection of resistant materials.
- Case studies: material degradation in extreme environments.

Chapter 6: Industrial Applications (3 weeks)

- **Metallic materials**: shape memory alloys, electrical interconnects.
- **Ceramic materials**: piezoelectric sensors, electronic substrates.
- **Polymeric materials**: encapsulants, conductive membranes.
- **Composite materials**: lightweight structures for drones, reinforcement of printed circuits.
- **Recent innovations:** 2D materials (graphene), nanocomposites, self-healing materials.

Chapter 7: Review (1 week)

- Review of key concepts.
- Written exam or synthesis project.

Assessment Methods

- Regular in class assessments (20%).
- Final exam (100%).

Recommended Textbooks and Resources:

1. Callister, W. D., & Rethwisch, D. G. - *Materials Science and Engineering: An Introduction* (10th Edition).
2. Ashby, M. F., & Jones, D. R. H. - *Engineering Materials 1: An Introduction to Properties, Applications, and Design*.
3. Kasap, S. O. - *Principles of Electronic Materials and Devices*.
4. Moulson, A. J., & Herbert, J. M. - *Electroceramics: Materials, Properties, Applications*.
5. Chawla, K. K. - *Composite Materials: Science and Engineering*.
6. Madou, M. J. - *Fundamentals of Microfabrication and Nanotechnology* (for MEMS).

Teaching Unit (UF)	Subject Title	Code	Semester
UEM2.2.1	**Advanced Embeded Systems**	**AES**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h00	**2/2**

Course Description:
This course provides a comprehensive understanding of embedded systems design where students will explore the fundamental principles of embedded systems architecture, hardware/software co-design, firmware development techniques, IoT technologies and safety-critical system design methodologies and regulations.

Course Objectives:
- Understanding of embedded system design principles and architectures.
- Develop proficiency in firmware coding.
- Explore IoT technologies and protocols.
- Understand safety-critical system design methodologies and regulations.

Recommended Prerequisites:
Familiarity with microcontroller programming and debugging, knowledge of real-time operating systems.

Course outline:

Chapter 1: Introduction to Embedded Systems
- Definition and characteristics of embedded systems.
- Embedded system applications (consumer electronics, industrial automation, home appliances, automotive systems, IoT …etc.).
- Embedded system architecture.
- Embedded system design challenges and constraints.

Chapter 2: Microcontrollers Types and Peripherals
- Recap of microcontroller architectures and peripherals.
- Microcontroller classes: low-power, high performance, mixed-signal, wireless …etc.
- Advanced microcontroller architectures and SoCs (e.g., ARM Cortex-M series, TI TDA4x, Zynq 7000).

Chapter 3: Hardware/Software Co-Design
- Co-design methodologies and strategies.
- Partitioning of hardware and software tasks.
- Design optimization.

Chapter 4: Firmware Development
- Firmware development process and lifecycle.
- Firmware structure (initialization and startup, interrupt service routines, peripheral drivers and abstraction layers, RTOS kernel, main application logic …etc.).
- Modular design principles.
- Consideration of API usage in optimizing code efficiency and performance.
- Implementing sleep modes and wake-up sources to conserve energy.

Chapter 5: Internet of Things (IoT)
- Definition of IoT and its significance in modern technology.
- Overview of IoT architecture and components.

- Evolution of IoT and its applications in various industries.

Chapter 6: IoT Ecosystem, Security and Standards

- IoT ecosystem: devices, connectivity, cloud platforms, and applications.
- IoT security challenges: device authentication, data encryption, access control, and secure firmware updates.
- IoT standards and protocols (e.g., MQTT, CoAP, HTTP, Zigbee).

Chapter 7: IoT Use Cases

- Exploration of real-world IoT use cases across industries: smart cities, healthcare, agriculture, industrial automation ...etc.
- Emerging trends in IoT technology: edge AI, digital twins, blockchain integration, and 5G connectivity.
- Predictions for the future of IoT and its potential impact on society and the economy.

Chapter 8: Safety-Critical Systems

- Definition and characteristics of safety-critical systems.
- Importance of safety in automotive, aerospace, medical, and industrial domains.
- Historical case studies of safety-critical system failures and their impact.

Chapter 9: Safety Standards and Regulations

- Safety standards such as ISO 26262 (automotive), DO-178C (aerospace), IEC 62304 (medical devices), and IEC 61508 (industrial).
- Regulatory frameworks and compliance requirements in different industries.

Chapter 10: Safety-Critical System Design Principles

- Functional safety concepts (SIL, ASIL).
- Safety lifecycle models (V-model, waterfall model).
- Hazard analysis and risk assessment techniques.

Chapter 11: Integration of Safety-Critical Considerations

- Incorporating safety requirements into system architecture and design.
- Safety validation and verification methodologies.

Laboratory work:

Lab 1: Sensor Interfacing: Interface a sensor (e.g., temperature sensor, light sensor) with a microcontroller and display/communicate the values read. Use interruptions.

Lab 2: Communication Interface Driver: Develop a basic driver for a communication interface (e.g., UART, I2C, SPI) using APIs. Use the developed driver in an application.

Lab 3: Sensor Driver: Develop a basic driver for a specific sensor using APIs. Use the developed driver in an application.

Lab 4: RTOS Implementation: Integrate an RTOS (e.g., FreeRTOS or µC/OS-II) into a project. Create multiple tasks to handle different functionalities concurrently, such as sensor reading, data processing, and communication.

Lab 5: IoT Communication Protocols: Implement a simple IoT project using MQTT protocol for device-to-cloud communication.

Lab 6: IoT application: Developing IoT applications using popular frameworks (e.g., Node-RED, AWS IoT Core, Azure IoT Hub).

Lab 7: Safety Architecture Design: Design a safety architecture for the system incorporating redundancy, diversity, and fault tolerance mechanisms. Implement safety-critical functions using hardware and software techniques to achieve the required safety integrity level.

Assessment Methods:

- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:

1. "Embedded Systems: Architecture, Programming and Design", Raj Kamal.
2. "Programming Embedded Systems in C and C++", Michael Barr.
3. "Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications", Robert Oshana and Mark Kraeling.
4. "Embedded Systems: Introduction to ARM Cortex-M Microcontrollers", Jonathan Valvano
5. "Safety Critical Systems Handbook: A Straightforward Guide to Functional Safety, IEC 61508 (2010 Edition) and Related Standards", David J. Smith.

Teaching Unit (UF)	Subject Title	Code	Semester
UEM2.2.1	**Machine Dynamics**	**MADY**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30		**3/3**

Course Objectives:

- Understand the fundamental principles of machine dynamics and their role in engineering design.
- Analyze vibrations, forces, and motion in various mechanical systems.
- Develop skills in diagnosing and mitigating dynamic issues in robotic and autonomous systems.
- Apply theoretical concepts to real-world engineering applications.

Recommended Prerequisites:

Basic Vibrations, Dynamics, Machine Design.

Course Outline

Chapter 1: Introduction to Machine Dynamics (2 weeks)

- Importance of dynamics in mechanical systems
- Types of motion and force transmission in machines
- Dynamic modeling of mechanical systems
- Case studies of machine failures due to dynamic issues

Chapter 2: Vibrations in Machines (3 weeks)

- Free and forced vibrations in mechanical systems
- Damping effects and resonance phenomena
- Vibration isolation and mitigation strategies
- Case studies on vibration-related failures in machinery

Chapter 3: Balancing of Rotating and Reciprocating Machines (2 weeks)

- Unbalance in rigid and flexible rotors
- Single-plane and multi-plane balancing techniques
- Balancing of reciprocating engines and compressors
- Experimental methods for vibration and balancing analysis

Chapter 4: Gyroscopic Effects and Stability in Machines (2 weeks)

- Gyroscopic forces in rotating systems
- Stability considerations in high-speed machinery
- Dynamic response of gyroscopic systems
- Applications in robotic systems and aerospace engineering

Chapter 5: Critical Speeds, Whirl, and System Stability (3 weeks)

- Critical speed analysis in rotating and reciprocating systems
- Campbell diagrams and stability considerations
- Whirl and self-excited vibrations in machines
- Case studies on system failures due to instability

Assessment Methods:

- Regular in class assessments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:

1. Vibration Problems in Machines. Diagnosis and Resolution, ARTHUR W. LEES, CRC Press Taylor & Francis Group, 2016.
2. Dynamics of rotating machines, Michael I. Friswell, et al., Cambridge University Press, 2010.
3. Machinery vibration and rotor dynamics, John Vance et al., 2010.
4. Vibrations of Rotating Machinery: Volume 1. Basic Rotor dynamics: Introduction to Practical Vibration Analysis., Osami Matsushita et al., Mathematics for Industry, Springer, Japan, 2017.

Teaching Unit (UF)	Subject Title	Code	Semester
UEM2.2.2	**Computational intelligence for Optimization**	**CIO**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h30	**3/3**

Course Objectives:

- Understand the fundamentals of computational intelligence and optimization techniques.
- Explore heuristic, metaheuristic, and machine learning-based optimization methods.
- Apply optimization techniques to real-world engineering, robotics, and smart manufacturing problems.
- Gain hands-on experience with computational tools and frameworks for optimization.

Course Outline

Chapter 1: Introduction to Computational Intelligence and Optimization (2 weeks)

- Definition of optimization problems and classification.
- Exact vs. heuristic vs. metaheuristic optimization.
- Single-objective vs. multi-objective optimization.
- Case studies: Industrial applications in robotics and automation.

Chapter 2: Heuristic and Metaheuristic Optimization (3 weeks)

- Introduction to heuristic and metaheuristic approaches.
- Evolutionary Algorithms: Genetic Algorithms (GA), Differential Evolution (DE).
- Swarm Intelligence: Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO).
- Applications in robotics path planning, scheduling, and industrial automation.

Chapter 3: Bio-Inspired and Nature-Inspired Algorithms (3 weeks)

- Artificial Bee Colony (ABC), Firefly Algorithm, Cuckoo Search.
- Hybrid optimization approaches combining multiple algorithms.
- Industrial applications in energy-efficient systems, predictive maintenance.

Chapter 4: Machine Learning and Optimization (2 weeks)

- Reinforcement Learning (RL) for optimization (Q-Learning, Deep Q-Networks).
- Neural Networks for optimization (ANN, Deep Learning).
- Neuro-Fuzzy systems and their applications in control and decision-making.

Chapter 5: Multi-Objective Optimization and Decision-Making (3 weeks)

- Pareto Optimization: NSGA-II, SPEA2, trade-off analysis.
- Handling conflicting objectives in industrial applications.
- Smart manufacturing and robotics applications.

Chapter 6: Advanced Implementation and Case Studies (2 weeks)

- Hybrid approaches: Combining optimization with AI and IoT.
- Real-world industrial optimization problems.

Laboratory work:

Lab 1: Introduction to Python optimization libraries (Scipy, DEAP).
Lab 2: Implementing GA and PSO for robotic trajectory optimization.
Lab 3: Comparing the performance of different bio-inspired optimization algorithms.

Lab 4: Using RL for optimizing robotic control strategies.
Lab 5: Multi-objective optimization in mechatronics (energy vs. performance).
Lab 6: Final Project: Students implementation of a solution for a real-world problem in robotics, manufacturing, or logistics.

Assessment Methods:
- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources:

1. "Computational Intelligence: Concepts to Implementations", Russell C. Eberhart, Yuhui Shi, Morgan Kaufmann, 2007.
2. "Swarm Intelligence: From Natural to Artificial Systems", Eric Bonabeau, Marco Dorigo, Guy Theraulaz, Oxford University Press, 1999.
3. "Evolutionary Algorithms for Solving Multi-Objective Problems", Kalyanmoy Deb, Springer, 2nd Edition, 2014.
4. "Neural Networks for Optimization and Signal Processing", Andreas Antoniou, Wu-Sheng Lu, Wiley, 1st Edition, 1993.
5. "Fuzzy Logic for Planning and Decision Making", Freerk A. Lootsma, Springer, 1st Edition, 1997.
6. "Introduction to Evolutionary Computing", A.E. Eiben, J.E. Smith, Springer, 2nd Edition, 2015.

Teaching Unit (UF)	Subject Title	Code	Semester
UET2.2.1	**Project management**	**PROM**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30			**1/1**

Course Objectives:
Students will be trained in:

- Identifying, assembling, and/or evaluating business creation or takeover projects across various dimensions (strategic, financial, organizational, human, commercial, etc.).
- Managing a company with a focus on economic development and financial profitability, emphasizing sustainable production practices, i.e., centered around work and productive investment.
- Advising and supporting business leaders in developing new projects and/or strategies, securing funding, choosing the appropriate legal structure for the company, and handling administrative procedures.
- Presenting and defending a business project to potential partners (financial or institutional).

Skills and Competencies:
Upon completion of the program, students will be able to:

- Analyze the company's environment and resources to develop its growth strategy.
- Implement the company's strategy across various functions (sources and methods of financing, business expansion, production management, innovation management, human resources management, etc.).
- Understand how to integrate the company into its local network in line with current economic concerns (sharing economy, CSR and sustainable development, the role of information technologies, business networks, etc.).
- Design and evaluate project financing conditions.
- Create a business model or "BMC" (Business Model Canvas).
- Develop an operational business plan.
- Design the commercial organization and create a marketing plan.
- Understand the challenges related to the digitalization of the economy and businesses.
- Develop a legal status for the company tailored to the project.

Oversee the management of the company's human resources (recruitment, payroll, training, etc.).

Teaching Unit (UF)	Subject Title	Code	Semester
UED2.2.1	**Reverse Engineering and Rapid Prototyping**	**RERP**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30			**1/1**

Course Overview:
This course introduces students to the principles, techniques, and practices of reverse engineering and rapid prototyping. It covers methods for digitizing physical objects, reconstructing 3D models, and fabricating prototypes using additive and subtractive manufacturing techniques. Students will learn how to analyze existing products, extract design information, and use this data to create new prototypes quickly and efficiently. The course emphasizes the integration of CAD, scanning technologies, and advanced manufacturing for industrial applications.

Course Objectives:
By the end of this course, students will be able to:

1. Understand the fundamentals of reverse engineering and rapid prototyping.
2. Use 3D scanning and data acquisition techniques.
3. Process and reconstruct 3D models from scanned data.
4. Apply additive and subtractive manufacturing technologies for prototyping.
5. Assess the advantages and limitations of different RP technologies.
6. Implement reverse engineering and RP techniques in real-world applications.

Prerequisites:

- Basics of Mechanical Design and Manufacturing
- CAD (Computer-Aided Design) Fundamentals
- Knowledge of Materials and Manufacturing Processes
- Basic Programming (Python, MATLAB, or similar)

Course outline:

Chapter 1: Introduction to Reverse Engineering & Rapid Prototyping

- Definition, scope, and importance of Reverse Engineering (RE) & Rapid Prototyping
- Historical context and modern applications
- Industrial applications
- Relationship with CAD , digital manufacturing, Product Development and Innovation

Chapter 2: Reverse Engineering Process

- Steps in Reverse Engineering
- Object digitization techniques
- 3D scanning technologies (Laser scanning, Structured light, CT scanning)
- Point cloud processing and surface reconstruction

Chapter 3: CAD Modeling for Reverse Engineering

- Introduction to CAD software (e.g., SolidWorks, Fusion 360).
- Converting Scan Data into CAD Models
- Surface Reconstruction and Mesh Processing
- Parametric and Non-Parametric Modeling
- File Formats and Data Conversion (STL, IGES, STEP, ...)
- Error correction and model optimization

Chapter 4: Additive Manufacturing (AM) Technologies

- Overview of 3D printing technologies:
  - Fused Deposition Modeling (FDM)
  - Stereolithography (SLA)
  - Selective Laser Sintering (SLS)
  - Digital Light Processing (DLP)
  - Direct Metal Laser Sintering (DMLS)
- Material considerations: plastics, metals, ceramics, and composites

Chapter 5: Subtractive Manufacturing and Hybrid Techniques

- CNC machining for prototyping
- Laser cutting and engraving
- Hybrid manufacturing: combining additive and subtractive techniques

Chapter 6: Design for Additive Manufacturing (DfAM)

- From 3D Scanning to 3D Printing
- Design constraints and optimization
- Lattice structures and topology optimization
- Post-processing and finishing techniques

Chapter 7: Applications and Case Studies

- Aerospace, automotive, biomedical, and consumer products
- Customization and personalization of products.
- Rapid tooling and manufacturing.
- Legal and ethical considerations in reverse engineering
- Future trends in rapid prototyping and reverse engineering

Assessment Methods:

- Final exam (100%).

Recommended Textbooks and Resources:

1. **Gibson, I., Rosen, D.W., Stucker, B.** (2015). *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*. Springer.
2. **Varady, T., Martin, R.R., Cox, J.** (1997). *Reverse engineering of geometric models – An introduction*. Computer-Aided Design.
3. **Rafiq Noorani (2005)** "Rapid Prototyping: Principles and Applications" ISBN: 978-0-471-73001-9
4. **software manuals**: Geomagic, SolidWorks, Fusion 360, Cura, etc.

Teaching Unit (UF)	Subject Title	Code	Semester
UED2.2.1	**Introduction to Industry 4.0 and Smart Manufacturing**	**IISM**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30			**1/1**

Course Objectives:

- Understand the fundamentals of Industry 4.0 and Smart Manufacturing
- Explore key technologies: MES, Digital Twin, OT/IT, Robotics 4.0, etc.
- Understand the transition to Industry 5.0 and emerging trends
- Analyze real-world applications and industrial use cases

Course Outline:

Chapter 1: Introduction to Industry 4.0 and Smart Manufacturing (2 weeks)

- Industrial evolution: From Industry 1.0 to 4.0
- Definition and key technologies of Industry 4.0
- Impact on production systems and required skillsets
- Towards Industry 5.0: Personalization, human-machine collaboration, and sustainability

Chapter 2: OT/IT Integration in Industry 4.0 (2 weeks)

- Differences and convergence of OT (Operational Technology) and IT (Information Technology)
- Industrial system architecture and interoperability
- Communication standards (OPC UA, MQTT, TSN)
- Cybersecurity and protection of industrial systems

Chapter 3: Robotics 4.0 and Intelligent Automation (2 weeks)

- Evolution of robotics: From programmable robots to collaborative robots (cobots)
- Integration of robotics with IoT and AI for adaptive production
- Application examples: Predictive maintenance, machine vision, intelligent handling
- Towards Robotics 5.0: Human-machine interaction and cognitive augmentation

Chapter 4: Digital Twin and Production System Simulation (2 weeks)

- Definition and types of Digital Twins
- Integration of Digital Twin with robotics and predictive maintenance
- Digital twin for virtual commissioning of robotics and automated manufacturing systems
- Use cases: Design, validation, and real-time supervision

Chapter 5: Operational Management and Production Control (2 weeks)

- Lean Manufacturing and digitalization of processes
- AI and IoT in operational management
- Key performance indicators (OEE, TRS) and industrial process optimization
- Case studies of smart factories

Chapter 6: MES Systems and Product-Controlled Manufacturing (2 weeks)

- Role of Manufacturing Execution Systems (MES) in Industry 4.0
- Production order management, real-time traceability, and quality control
- Flexible manufacturing and mass customization
- MES-ERP integration and its impact on industrial management

Chapter 7: Industry 5.0 and Future Perspectives (2 weeks)

- From Industry 4.0 to 5.0: Complementarity and differences
- Human-centered manufacturing: Personalization, ergonomics, and augmented work
- AI ethics in industry
- Sustainability and environmental impact of industrial technologies

Assessment Methods:
- Final exam (100%).

Recommended Textbooks and Resources:
1. Industry 4.0: The Industrial Internet of Things, Alasdair Gilchrist, 2016.
2. Smart Manufacturing: Concepts and Applications, Masoud Soroush, William L. Luyben, Michael Baldea, 2020.
3. The Fourth Industrial Revolution, Klaus Schwab, 2017.
4. Digital Twin Technology and Smart Manufacturing, Fei Tao, A.Y.C. Nee, 2020.
5. Manufacturing Execution Systems (MES): Optimal Design, Planning, and Deployment, Heinz-Hermann Erbe, 2009.
6. Industrial Internet of Things: Cybermanufacturing Systems, Sabina Jeschke, Christian Brecher, Houbing Song, Danda B. Rawat, 2017.

Teaching Unit (UF)	Subject Title	Code	Semester
UED2.2.1	**Training Internship II**	**TI-II**	**S4**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH				**1/1**

Course Objectives:

- Strengthen practical skills through deeper industry engagement.
- Apply theoretical knowledge to real-world problems.
- Develop problem-solving and critical thinking abilities.

Targeted Skills

- Advanced technical proficiency.
- Independent problem-solving.
- Effective collaboration in teams.
- Adaptation to professional expectations.

Recommended prerequisites

- Intermediate knowledge of Autonomous systems and industrial robotics.
- Hands-on experience with basic tools and technologies.

Content

1. **Company Integration (3 hours)**
    - Reviewing company expectations and past experience.
    - Workplace protocols and project assignments.
2. **Active Participation in Projects (9 hours)**
    - Engaging in specific technical tasks.
    - Working alongside professionals in real-world scenarios.
3. **Reporting and Documentation (3 hours)**
    - Documenting progress and key learnings.
    - Mid-internship feedback session.
4. **Final Assessment and Report Submission (3 hours)**
    - Presenting internship experience to faculty.
    - Final evaluation by company mentor.

Recommended Texbooks and Resources

1. The Successful Internship by H. Frederick Sweitzer
2. Problem Solving 101: A Simple Book for Smart People by Ken Watanabe

Detailed programs for the ninth semester

Teaching Unit (UF)	Subject Title	Code	Semester
UEF3.1.1	**Advanced Machine Learning for Robotics**	**AMLR**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Description:
This course will prepare students for careers in robotics research, autonomous systems development, and AI-driven automation, as well as provide a strong foundation for further academic pursuits in the field.

Course Objectives:

- fundamentals of RL, including Markov Decision Processes (MDPs), policy optimization, and value functions.
- Implement advanced RL algorithms (PPO, DDPG, SAC, TD3) for robotic control tasks.
- Analyze the strengths and limitations of different RL algorithms in real-world robotics applications.

2. Design and Optimize Trajectories Using Model Predictive Control (MPC):
   - Formulate trajectory planning problems with constraints and objectives.
   - Implement MPC for real-time trajectory optimization in robotic systems.
   - Integrate MPC with reinforcement learning for adaptive control.
3. Utilize Deep Learning for Robotic Control:
   - Understand the architecture and applications of RNNs and LSTMs in robotics.
   - Train deep learning models for state prediction and real-time control.
   - Combine deep learning with reinforcement learning for enhanced robotic decision-making.
4. Develop Practical Skills Through Hands-On Projects:
   - Gain experience in simulating and controlling robotic systems using Python, PyTorch, TensorFlow, and ROS.
   - Work on real-world robotics problems through a capstone project.
   - Present and communicate technical solutions effectively.
5. Prepare for Advanced Research and Industry Applications:
   - Explore cutting-edge research in machine learning and robotics.
   - Develop the ability to critically evaluate and apply advanced techniques to complex robotic systems.

Recommended Prerequisites:

- Introduction to Machine Learning
- Linear Algebra and Probability
- Control Systems and Robotics
- Python Programming
- Signal Processing

Chapter 1: Reinforcement Learning for Robotic Control (4 weeks)
Topics Covered:

1. **Introduction to Reinforcement Learning (RL)**
    - Markov Decision Processes (MDPs)
    - Policy, Value Functions, and Q-Learning
    - Exploration vs. Exploitation
2. **Deep Reinforcement Learning (DRL)**
    - Neural Networks in RL
    - Policy Gradient Methods
3. **Advanced RL Algorithms**
    - PPO (Proximal Policy Optimization)
    - DDPG (Deep Deterministic Policy Gradient)
    - SAC (Soft Actor-Critic)
    - TD3 (Twin Delayed DDPG)
4. **Applications in Robotics**
    - Robotic arm control
    - Autonomous navigation

Chapter 2: Trajectory Planning and Optimization (3 weeks)
Topics Covered:

1. **Introduction to Trajectory Planning**
    - Kinematics and dynamics in robotics
    - Constraints and objectives in trajectory planning
2. **Model Predictive Control (MPC)**
    - Basics of MPC
    - Formulating cost functions and constraints
    - Real-time optimization
3. **Applications in Robotics**
    - Path planning for mobile robots
    - Trajectory optimization for robotic manipulators
4. **Integration with Reinforcement Learning**
    - Combining MPC with RL for adaptive control

Chapter 3: Deep Learning for Robotic Control (3 weeks)
Topics Covered:

1. **Recurrent Neural Networks (RNNs)**
    - Basics of RNNs
    - Applications in robotics
2. **Long Short-Term Memory (LSTM) Networks**
    - Architecture and advantages
    - Handling sequential data
3. **State Prediction and Real-Time Control**
    - Predicting robot states using LSTM
    - Real-time control with RNN/LSTM
4. **Integration with Reinforcement Learning**
    - Using RNN/LSTM for policy networks in RL

Chapter 4: Capstone Project (4 weeks)
Objective:
Apply the knowledge gained in the course to a real-world robotics problem.
Project Outline:

1. Develop a reinforcement learning-based controller for a robotic arm using PPO or SAC.
2. Implement MPC for trajectory planning in an autonomous drone.

3. Combine RNN/LSTM with RL for real-time control of a mobile robot.

Deliverables:

- Project proposal and design document
- Implementation and simulation results
- Final presentation and report

Laboratory Work

Lab 1: Reinforcement Learning for Robotic Control

1. **Implement PPO for Robotic Arm Control**
    - Use OpenAI Gym and PyBullet to simulate a robotic arm.
    - Train a PPO agent to control the arm to reach a target position.
    - Analyze the performance and stability of the algorithm.
2. **Compare SAC and TD3 for Autonomous Navigation**
    - Set up a simulated environment for a mobile robot.
    - Train SAC and TD3 agents for navigation tasks.
    - Compare their performance in terms of convergence speed and robustness.

Lab 2: Trajectory Planning and Optimization

1. **Implement MPC for Mobile Robot Navigation**
    - Use Python and CasADi to implement MPC.
    - Simulate a mobile robot navigating through obstacles.
    - Optimize the trajectory to minimize travel time and avoid collisions.
2. **Trajectory Optimization for Robotic Manipulators**
    - Apply MPC to a robotic arm for picking and placing objects.
    - Experiment with different cost functions and constraints.

Lab 3: Deep Learning for Robotic Control

1. **Train an LSTM Network for State Prediction**
    - Collect data from a simulated robotic system.
    - Train an LSTM network to predict future states (e.g., position, velocity).
    - Evaluate the accuracy of the predictions.
2. **Real-Time Control Using RNNs**
    - Implement an RNN-based controller for a robotic system.
    - Test the controller in real-time simulations and analyze its performance.

Assessment Methods:

- Regular in class assessments (20%).
- Laboratory reports and practical assignments + Capstone project (20%).
- Final exam (60%).

Recommended Textbooks and Resources:

- **Textbooks:**
    1. *Reinforcement Learning: An Introduction* by Richard S. Sutton and Andrew G. Barto (2nd Edition) – Chapters 3, 6, 9, 13.
    2. *Deep Reinforcement Learning Hands-On* by Maxim Lapan – Chapters 7, 9, 10, 12.
    3. Model Predictive Control: Theory, Computation, and Design by James B. Rawlings and David Q. Mayne – Chapters 1, 2, 3.
    4. Robot Modeling and Control by Mark W. Spong, Seth Hutchinson, and M. Vidyasagar – Chapters 6, 7.
    5. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville – Chapters 10, 11.

6. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron – Chapters 14, 15.
7. Reinforcement Learning: An Introduction by Richard S. Sutton and Andrew G. Barto (for RL projects).
8. Model Predictive Control: Theory, Computation, and Design by James B. Rawlings and David Q. Mayne (for MPC projects).

- **Online Resources:**
  1. OpenAI Spinning Up Documentation: https://spinningup.openai.com/
  2. Stable-Baselines3 Documentation: https://stable-baselines3.readthedocs.io/
  3. MPC Tutorial by Stanford: https://stanford.edu/class/ee364b/
  4. CasADi (Optimization Tool for MPC): https://web.casadi.org/
  5. TensorFlow RNN Tutorial: https://www.tensorflow.org/guide/keras/rnn
  6. PyTorch LSTM Tutorial: https://pytorch.org/tutorials/beginner/nlp/sequence_models_tutorial.html
  7. OpenAI Gym: https://www.gymlibrary.dev/
  8. PyBullet Robotics Simulator: https://pybullet.org/

- **Tools:**
  - MATLAB, Python (CVXPY, CasADi), ROS (Robot Operating System), OpenAI Gym, PyBullet, PyTorch, TensorFlow, Keras.

Teaching Unit (UF)	Subject Title	Code	Semester
UEF3.1.1	**Distributed Systems and Agent-Based Technologies**	**DSABT**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Objectives:

- Understand the fundamental principles of distributed systems and their applications
- Explore control architectures for robotics and smart manufacturing
- Analyze the evolution of control approaches, from behavior-based systems to swarm intelligence
- Study state-of-the-art techniques in multi-agent systems
- Learn about agent-based technologies for distributed control
- Examine implementation concepts and real-world use cases
- Gain hands-on experience with JADE and other agent-based frameworks

Course Outline:

Chapter 1: Introduction to Distributed Systems (2 weeks)

- Definition and key characteristics of distributed systems
- Advantages and challenges in industrial and robotic applications
- Communication and coordination in distributed environments
- Case studies: Cloud-based robotics, edge computing in automation

Chapter 2: Control Architectures for Robotics and Smart Manufacturing (2 weeks)

- Centralized vs. decentralized vs. distributed control
- Hierarchical and heterarchical architectures
- Industrial applications: Cyber-physical systems, IoT-based control, and smart factories
- Examples from real-world robotic and manufacturing systems

Chapter 3: Evolution of Control: From Behavior-Based Systems to Swarm Intelligence (2 weeks)

- Behavior-based robotics and its principles (Brooks' subsumption architecture)
- Bio-inspired approaches: Ant colony optimization, particle swarm optimization
- Distributed decision-making and self-organizing systems
- Applications in autonomous robotics and industrial automation

Chapter 4: State-of-the-Art Techniques in Distributed Systems (2 weeks)

- Edge AI and real-time decision-making in distributed environments
- Blockchain for industrial automation and robotic coordination
- Digital twins and their integration with multi-agent systems
- Emerging trends in distributed robotics and smart manufacturing

Chapter 5: Multi-Agent Technologies for Distributed Systems (2 weeks)

- Definition and principles of multi-agent systems (MAS)
- Agent communication languages and protocols (FIPA-ACL, KQML)
- Coordination and negotiation mechanisms in MAS
- Applications in robotics, logistics, and manufacturing

Chapter 6: Implementation Concepts and Use Cases (2 weeks)

- Distributed robotics: Swarm robotics and multi-robot collaboration
- Smart manufacturing: Self-adaptive production lines and predictive maintenance
- Supply chain optimization with multi-agent coordination
- Case studies from industry and research

Chapter 7: Practical Implementation with JADE and Other Frameworks (2 weeks)

- Introduction to JADE (Java Agent Development Framework)
- Setting up and developing agent-based applications in JADE
- Overview of other frameworks (SPADE, Jason, AnyLogic)
- Hands-on session: Implementing a simple multi-agent system

Teaching Methods:

- Theoretical lectures with interactive discussions
- Case studies and real-world applications
- Practical demonstrations and coding sessions
- Mini-projects on multi-agent systems and distributed control

Assessment Methods:

- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources:

1. "Distributed Systems: Concepts and Design", George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, Pearson, 5th Edition, 2012.
2. "Distributed Systems: An Algorithmic Approach", Sukumar Ghosh, Chapman and Hall/CRC, 2nd Edition, 2014.
3. "Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence", Jacques Ferber, Addison-Wesley, 1st Edition, 1999.
4. "Reinforcement Learning: An Introduction", Richard S. Sutton, Andrew G. Barto, MIT Press, 2nd Edition, 2018.
5. "Distributed Algorithms", Nancy A. Lynch, Morgan Kaufmann, 1st Edition, 1996.
6. "Agent-Based Modeling and Simulation", Stefano Balietti, Springer, 1st Edition, 2016.

Teaching Unit (UF)	Subject Title	Code	Semester
UEF3.1.2	**Human-Machine, Machine-Machine Interaction/Collaboration**	**MMI**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h00	**3/3**

Course Objectives:
This course is designed to equip students with the knowledge and skills necessary to excel in the rapidly evolving field of robotics and autonomous systems, with a focus on enabling seamless interaction and collaboration between humans and machines.
By the end of this course, students will:

1. Understand the theoretical foundations of human-machine interaction (HMI) and machine-machine collaboration (MMC).
2. Gain practical skills in designing and implementing interactive and collaborative systems.
3. Analyze and evaluate the ethical, social, and technical challenges in HMI and MMC.
4. Develop systems that enable seamless communication and cooperation between humans and robots, as well as between autonomous systems.
5. Apply knowledge to real-world scenarios in robotics and autonomous systems design.

Recommended Prerequisites:

- Introduction to Robotics
- Programming for Robotics (Python/C++)
- Control Systems
- Artificial Intelligence Basics

Course Outline

Chapter 1: Introduction to Human-Machine Interaction (HMI) and Machine-Machine Collaboration (MMC) (1 week)

- Overview of HMI and MMC in robotics and autonomous systems.
- Historical evolution and key milestones.
- Applications in industry, healthcare, transportation, and smart cities.
- Case studies: Collaborative robots (cobots), autonomous vehicles, and swarm robotics.

Chapter 2: Fundamentals of Interaction Design (1 week)

- Principles of user-centered design.
- Cognitive ergonomics and human factors in HMI.
- Interface design for robots and autonomous systems.
- Tools for prototyping HMI (e.g., ROS, Unity, MATLAB).

Chapter 3: Communication Protocols and Standards (2 weeks)

- Overview of communication protocols for MMC (e.g., MQTT, ROS, DDS).
- Wireless communication technologies (Wi-Fi, Bluetooth, Zigbee).
- Standardization in robotics (ISO, IEEE).
- Interoperability challenges and solutions.

Chapter 4: Human-Robot Interaction (HRI) (2 weeks)

- Types of HRI: physical, social, and cognitive.
- Sensors and actuators for HRI (e.g., cameras, microphones, force sensors).
- Natural language processing (NLP) for voice-based interaction.
- Gesture recognition and haptic feedback.

Chapter 5: Machine-Machine Collaboration (MMC) (2 weeks)

- Multi-agent systems and swarm robotics.
- Task allocation and coordination in MMC.
- Communication and negotiation protocols.
- Distributed control and decision-making.

Chapter 6: Artificial Intelligence in HMI and MMC (2 weeks)

- Role of AI in enabling intelligent interaction and collaboration.
- Machine learning for adaptive HMI.
- Reinforcement learning for MMC.
- Explainable AI (XAI) for transparent decision-making.

Chapter 7: Ethical and Social Implications (1 week)

- Ethical considerations in HMI and MMC.
- Privacy and security concerns.
- Impact on employment and society.
- Regulatory frameworks and guidelines.

Chapter 8: Human-Centered Robotics (1 week)

- Designing robots for human environments.
- Safety standards and risk assessment.
- Collaborative robots (cobots) in industrial settings.
- Assistive robotics in healthcare and rehabilitation.

Chapter 9: Autonomous Systems Collaboration (1 week)

- Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication.
- Drone swarms and their applications.
- Industrial IoT (IIoT) and smart factories.
- Challenges in real-time collaboration.

Chapter 10: Advanced Topics in HMI and MMC (2 weeks)

- Brain-computer interfaces (BCI) for direct human-machine communication.
- Augmented reality (AR) and virtual reality (VR) in HMI.
- Edge computing and fog computing for MMC.
- Quantum communication for secure MMC.

Laboratory work:

Lab 1: Human-Machine Interaction (HMI) Design and Evaluation

Objective:

Design and evaluate a user interface for controlling a robotic system.

Tasks:

1. Develop a graphical user interface (GUI) using tools like MATLAB App Designer, Python Tkinter, or ROS-based interfaces.
2. Integrate the GUI with a robotic arm or mobile robot for tasks like pick-and-place or navigation.
3. Conduct usability testing with human operators to evaluate the interface's intuitiveness and efficiency.

4. Analyze feedback and propose improvements to the HMI design.

Tools/Equipment:

- Robotic arm or mobile robot
- Laptop with GUI development software
- Sensors (e.g., cameras, joysticks)

Lab 2: Collaborative Robotics (Cobots) in Shared Workspaces

Objective:

Implement and test a collaborative robot (cobot) working alongside a human operator.

Tasks:

1. Set up a cobot (e.g., Universal Robots UR5 or similar) in a shared workspace.
2. Program the cobot to perform a task (e.g., assembly or sorting) while ensuring safety protocols (e.g., force-limiting, collision detection).
3. Introduce a human operator to interact with the cobot in real-time.
4. Evaluate the efficiency and safety of the collaboration.

Tools/Equipment:

- Collaborative robot (cobot)
- Safety sensors (e.g., LiDAR, force-torque sensors)
- Workpiece and assembly tools

Lab 3: Gesture and Voice-Based Human-Robot Interaction

Objective:

Develop and test a human-robot interaction system using gestures and voice commands.

Tasks:

1. Use a gesture recognition system (e.g., Leap Motion or Kinect) and a voice recognition system (e.g., Google Speech-to-Text or Amazon Alexa) to control a robot.
2. Program the robot to respond to specific gestures and voice commands (e.g., "move forward," "stop").
3. Evaluate the accuracy and responsiveness of the system.
4. Discuss potential applications in industrial or service robotics.

Tools/Equipment:

- Gesture recognition device (e.g., Leap Motion)
- Microphone and voice recognition software
- Robot platform (e.g., mobile robot or robotic arm)

Lab 4: Human-Robot Collaboration in an Assembly Task

Objective:

Simulate and evaluate a human-robot collaborative assembly task.

Tasks:

1. Design an assembly task (e.g., building a small structure or assembling parts) where a human and robot work together.
2. Use a robot (e.g., cobot or industrial robot) to handle repetitive or precision tasks, while the human handles complex decision-making.
3. Implement safety mechanisms (e.g., emergency stop, speed reduction) to ensure safe interaction.
4. Measure the task completion time, error rate, and human workload.

Tools/Equipment:

- Collaborative or industrial robot
- Assembly components (e.g., screws, blocks)
- Safety equipment (e.g., emergency stop button, protective barriers)

Assessment Methods:

- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources

1. **Textbooks**:
    - "Human-Robot Interaction: An Introduction" by Christoph Bartneck et al.
    - "Multi-Robot Systems: From Swarms to Intelligent Automata" by Alan Schultz and Lynne Parker.
    - "Designing Interactive Systems" by David Benyon.
2. **Online Resources**:
    - ROS (Robot Operating System) documentation.
    - Coursera and edX courses on HMI and MMC.

Teaching Unit (UF)	Subject Title	Code	Semester
UEF3.1.2	**Computer Vision for Robotics**	**CVR**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Overview:
This course provides an in-depth understanding of computer vision techniques and their applications in robotics. It covers fundamental concepts, algorithms, and tools used in computer vision, with a focus on how these techniques are applied to enable robots to perceive, interpret, and interact with their environment. The course combines theoretical knowledge with practical implementation, using programming languages like Python and frameworks such as OpenCV and ROS (Robot Operating System).

Course Objectives:
- Understand the principles of computer vision and its role in robotics.
- Learn to implement computer vision algorithms for image processing, object detection, and tracking.
- Gain hands-on experience with vision-based robotic systems.
- Develop the ability to integrate computer vision into robotic applications such as navigation, manipulation, and human-robot interaction.
- Explore advanced topics like deep learning for computer vision and real-time vision systems.

Course Outline:
Chapter 1: Introduction to Computer Vision for Robotics (1 Week)
- Overview of computer vision and its applications in robotics.
- Key challenges in robotic perception.
- Introduction to tools: Python, OpenCV, and ROS.

Chapter 2 : Image Processing Fundamentals (1 Week)
- Image representation: pixels, color spaces (RGB, grayscale, HSV).
- Basic image operations: filtering, thresholding, and edge detection.
- Hands-on: Image processing with OpenCV.

Chapter 3: Feature Detection and Extraction (1 Week)
- Keypoint detection: Harris corner, SIFT, SURF, ORB.
- Feature descriptors and matching.
- Applications in robotics: localization and mapping.

Chapter 4: Camera Models and Calibration (1Week)
- Pinhole camera model and intrinsic/extrinsic parameters.
- Camera calibration techniques.
- Hands-on: Calibrating a camera using OpenCV.

Chapter 5: Stereo Vision and Depth Estimation (1Week)
- Epipolar geometry and stereo correspondence.
- Depth maps and 3D reconstruction.
- Applications in robotics: obstacle avoidance and navigation.

Chapter 6: Object Detection and Recognition (1 Week)
- Traditional methods: Haar cascades, HOG (Histogram of Oriented Gradients).
- Introduction to machine learning for object detection.
- Hands-on: Implementing object detection with OpenCV.

Chapter 7: Deep Learning for Computer Vision (1 Week)
- Introduction to convolutional neural networks (CNNs).
- Transfer learning and pre-trained models (e.g., YOLO, SSD, Faster R-CNN).
- Applications in robotics: real-time object detection and classification.

Chapter 8 : Visual Tracking (1 Week)
- Tracking algorithms: Kalman filter, optical flow, and mean-shift.
- Challenges in dynamic environments.
- Hands-on: Implementing visual tracking for robotic systems.

Chapter 9: Vision-Based Navigation (1 Week)
- Visual SLAM (Simultaneous Localization and Mapping).
- Feature-based and direct methods for SLAM.
- Case studies: Autonomous drones and mobile robots.

Chapter 10: Human-Robot Interaction with Computer Vision (1 Week)
- Gesture recognition and pose estimation.
- Facial recognition and emotion detection.
- Applications in collaborative robotics.

Chapter 11: Real-Time Vision Systems (1 Week)
- Optimizing computer vision algorithms for real-time performance.
- Hardware acceleration: GPUs and embedded systems.
- Hands-on: Building a real-time vision system for a robotic application.

Chapter 12: Advanced Topics and Emerging Trends (1 Week)
- Semantic segmentation and scene understanding.
- Multi-sensor fusion: Combining vision with LiDAR, IMU, etc.
- Ethical considerations in computer vision for robotics.

Laboratory Work

Lab 1: Introduction to OpenCV and Image Processing
- **Objective:** Familiarize students with OpenCV and basic image processing techniques.
- **Tasks:**
  - Load and display images using OpenCV.
  - Convert images between color spaces (RGB, grayscale, HSV).
  - Apply filters (Gaussian blur, median blur) and edge detection (Canny, Sobel).

Lab 2: Feature Detection and Matching
- **Objective:** Implement feature detection and matching algorithms.
- **Tasks:**
  - Detect keypoints using SIFT, SURF, or ORB.
  - Compute feature descriptors and match them between two images.
  - Visualize matched features.

Lab 3: Camera Calibration
- **Objective:** Learn camera calibration techniques to correct lens distortion.
- **Tasks:**
  - Capture images of a calibration pattern (e.g., chessboard).
  - Compute intrinsic and extrinsic camera parameters.
  - Undistort images using the calibration results.

Lab 4: Stereo Vision and Depth Estimation
- **Objective:** Implement stereo vision to estimate depth.
- **Tasks:**
  - Calibrate a stereo camera setup.
  - Compute disparity maps and depth maps.
  - Visualize 3D point clouds.

Lab 5: Object Detection with Traditional Methods
- **Objective:** Detect objects using traditional computer vision methods.

- **Tasks:**
  - Implement Haar cascades for face detection.
  - Use HOG (Histogram of Oriented Gradients) for pedestrian detection.
  - Evaluate detection performance on sample datasets.

Lab 6: Deep Learning-Based Object Detection

- **Objective:** Implement deep learning models for object detection.
- **Tasks:**
  - Fine-tune a pre-trained model (e.g., YOLO, SSD, or Faster R-CNN).
  - Detect objects in real-time using a webcam or video stream.
  - Evaluate model performance using metrics like mAP (mean Average Precision).

Lab 7: Visual Tracking

- **Objective:** Implement visual tracking algorithms.
- **Tasks:**
  - Track objects using optical flow or the Kalman filter.
  - Compare the performance of different tracking algorithms.
  - Apply tracking to a robotic system (e.g., tracking a moving object with a robot arm).

Lab 8: Visual SLAM

- **Objective:** Implement a basic visual SLAM system.
- **Tasks:**
  - Extract features from a video stream and estimate camera motion.
  - Build a map of the environment using feature-based SLAM.
  - Visualize the robot's trajectory and the map.

Lab 9: Human-Robot Interaction with Computer Vision

- **Objective:** Enable robots to interact with humans using vision.
- **Tasks:**
  - Implement gesture recognition using pose estimation (e.g., MediaPipe or OpenPose).
  - Detect and recognize faces using OpenCV or deep learning models.
  - Simulate a human-robot interaction scenario (e.g., robot responding to gestures).

Lab 10: Real-Time Vision System

- **Objective:** Build a real-time vision system for robotics.
- **Tasks:**
  - Optimize a computer vision pipeline for real-time performance.
  - Integrate the vision system with a robotic platform (e.g., TurtleBot or drone).
  - Perform real-time object detection, tracking, or navigation.

Lab 11: Multi-Sensor Fusion

- **Objective:** Combine vision with other sensors for enhanced perception.
- **Tasks:**
  - Fuse camera data with LiDAR or IMU data.
  - Implement a simple obstacle avoidance system using sensor fusion.
  - Visualize the combined data in a simulation environment (e.g., Gazebo).

Assessment Methods:

- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).
- Final exam (60%).

Recommended Textbooks and Resources:

- Jan Erik Solem "Programming Computer Vision with Python" . ISBN **978-1449316549**

- Joseph Howse and Joe Minichino. "Learning OpenCV 4 Computer Vision with Python" by "Deep Learning for Computer Vision" ISBN **978-1789531619.**
- Richard Szeliski "Computer Vision: Algorithms and Applications" (2e édition, 2022)
- Richard Hartley, Andrew Zisserman "Multiple View Geometry in Computer Vision" (2e édition, 2004)
- Rafael C. Gonzalez, Richard E. Woods "Digital Image Processing" (4e édition, 2017)
- Danica Kragic, Ville Kyrki "Vision for Robotics" *(2019)*
- Kristen Grauman "Visual Perception for Manipulation and Imitation in Robotics" *(2020)*
- Peter Corke Robotics, Vision and Control Fundamental Algorithms in MATLAB © Springer International Publishing AG 2017
- Research papers on visual SLAM, object detection, and deep learning for robotics.

Additional Resources:

- Online tutorials and documentation for OpenCV and ROS.
- Datasets for computer vision (e.g., COCO, KITTI, ImageNet).
- Access to simulation tools like Gazebo for testing robotic applications.

Teaching Unit (UF)	Subject Title	Code	Semester
UEF3.1.2	**Real-Time Operating Systems**	**RTOS**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h00	**3/3**

Course Description:
This course provides knowledge about real-time operating systems (RTOS) and their applications in embedded systems. Students will learn about the principles, design considerations, scheduling algorithms, and synchronization mechanisms involved in building real-time systems.

Course Objectives:
- Understand the fundamentals of real-time systems and their requirements.
- Learn about different types of real-time operating systems and their characteristics.
- Gain proficiency in designing, implementing, and analysing real-time scheduling algorithms.
- Explore synchronization techniques for real-time systems.
- Develop skills in programming, debugging and testing real-time applications.

Recommended Prerequisites:
Understanding of computer architecture concepts, familiarity with programming and debugging.

Course outline:
Chapter 1: Introduction to Operating Systems (1 week)
- Definition and functions of an operating system.
- Types of operating systems: batch processing, multiprogramming, time-sharing, distributed, and real-time.
- Processes and threads: definition, attributes, and states.
- Scheduling algorithms: FCFS, SJF, Round Robin, Priority Scheduling.

Chapter 2: Introduction to Real-Time Systems (1 week)
- Definition and characteristics of real-time systems.
- Types of real-time systems: hard, firm, and soft real-time.
- Challenges and requirements in real-time systems design.

Chapter 3: Real-Time Operating System Basics (2 weeks)
- Real-time operating systems concepts.
- Real-time operating system architecture.
- Tasks and processes in real-time systems.

Chapter 4: Real-Time Scheduling (2 weeks)
- Basics of scheduling in real-time systems.
- Rate-monotonic scheduling (RMS).
- Earliest deadline first (EDF) scheduling.
- Comparison of different scheduling algorithms.

Chapter 5: Synchronization and Communication (3 weeks)
- Mutual exclusion and its importance in real-time systems.
- Synchronization primitives: semaphores, mutexes, and condition variables.

- Inter-process communication mechanisms.
- Deadlock and priority inversion in real-time systems.

Chapter 6: Memory Management (2 weeks)

- Memory management requirements in real-time systems.
- Techniques for memory allocation and deallocation.
- Memory protection and access control.

Chapter 7: Real-Time System Design Considerations (2 weeks)

- Design patterns for real-time systems.
- Techniques for reducing system latency.
- Power management in real-time systems.

Chapter 8: Case Studies and Applications (2 weeks)

- Real-world examples of real-time systems.
- Case studies in automotive, aerospace, medical, and industrial domains.
- Challenges and solutions in designing real-time systems for specific applications.

Laboratory Work:

Lab 1: Task Scheduling Simulation

Simulate task-scheduling algorithms, such as Rate-Monotonic Scheduling (RMS) and Earliest Deadline First (EDF), and observe the behaviour of tasks over time, including missed deadlines and task preemptions.

Lab 2: RTOS Porting

Port a real-time operating system (e.g., FreeRTOS) to a specific hardware platform (e.g., STM32 Nucleo Boards). Build and run simple applications (e.g., led blink).

Lab 3: Interrupt Handling and Synchronization

Create a lab scenario where tasks are interrupted by hardware interrupts (e.g., timer interrupts). Use shared resources, semaphores, mutexes.

Lab 4: Implement a simple communication protocol for real-time systems

Define and implement a basic communication protocol (e.g., message passing or shared memory) for inter-task communication. Test the protocol's performance under different loads.

Lab 5: Develop a real-world application using an RTOS

Assign a project where students must design and implement a real-time application (e.g., a data acquisition system, a robotic controller, or a sensor network) using an RTOS platform. Students should define system requirements, partition tasks, implement task scheduling, and handle external events.

Assessment Methods:

- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended Textbooks and Resources

"Real-Time Systems", Jane W. S. Liu, Prentice Hall,2000.
"Real-Time Operating Systems Book 1: The Theory", Jim Cooling, 2019.
"Embedded Systems: Real-Time Operating Systems for Arm Cortex-M Microcontrollers", Jonathan Valvano, CreateSpace Independent Publishing Platform, 2nd ed. Edition, 2012.
"Embedded Systems Building Blocks: Complete and Ready-to-Use Modules in C", Jean J. Labrosse, CRC Press, 1995.

Teaching Unit (UF)	Subject Title	Code	Semester
UEM3.1.1	**Marine Robotics**	**MROB**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30	1h30	1h00	**4/4**

Course Overview:
This course provides a comprehensive introduction to the design and development of marine robotic systems. It covers the fundamental principles of marine robotics, including underwater vehicle design, propulsion systems, navigation, sensing, and control. Students will gain hands-on experience in designing and simulating marine robots, with a focus on applications such as underwater exploration, environmental monitoring, and offshore operations. The course integrates theoretical knowledge with practical projects, using tools like CAD software, simulation environments, and prototyping platforms.

Course Objectives
By the end of this course, students will be able to:
1. Understand the principles of marine robotics and their applications.
2. Model and simulate marine robotic systems.
3. Learn to design and model marine robotic systems.
4. Gain proficiency in propulsion, navigation, and control systems for underwater vehicles.
5. Develop skills in integrating sensors and communication systems for marine environments.
6. Design control strategies for underwater robots.
7. Analyze the challenges of underwater environments (e.g., hydrodynamics, communication, ...).
8. Develop and test algorithms for autonomy, navigation, and sensing in marine robots.

Course Outline:
Chapter 1: Introduction to Marine Robotics
- Overview of marine robots: **AUVs**, **ROVs**, **USVs** and hybrid systems.
- Applications: ocean exploration, environmental monitoring, offshore industries, and defense.
- Challenges in underwater environments: pressure, communication, and navigation.

Chapter 2: Background : Hydrodynamics and Marine Vehicle Dynamics
- Basics of fluid mechanics and hydrodynamics.
- Equations of motion for underwater vehicles (6-DOF model).
- Effects of buoyancy, drag, and added mass.

Chapter 3: Architecture and components of Marine Robots
- Mechanical Systems
- Sensors: sonar, IMUs, depth sensors, cameras, and acoustic positioning.
- Actuators: thrusters, propellers, and control surfaces.
- Power systems and energy management.
- Computers, Communications, and Architecture

Chapter 4: Propulsion Systems
- Types of propulsion systems: propellers, thrusters, and bio-inspired mechanisms.
- Power requirements and energy efficiency.

- Hands-on: Designing a propulsion system using CAD software.

Chapter 5: Modeling and Simulation

- Mathematical modeling of marine robots.
- Case studies: modeling AUVs and ROVs.

Chapter 6: Control Systems for Marine Robots

- PID control for underwater vehicles.
- Advanced control strategies: adaptive control, sliding mode control, and model predictive control.
- Waypoint tracking and trajectory planning.

Chapter 7: Autonomy and Navigation

- Localization and mapping (SLAM) in underwater environments.
- Path planning and obstacle avoidance.
- Communication systems: acoustic modems and challenges.

Chapter 8: Underwater Manipulators

- Modeling
- Control
- Trajectory planning

Chapter 9: USVs (Unmanned Surface Vehicles) or **ASVs** (Autonomous Surface Vehicles),

- Modeling
- Control
- Trajectory planning

Chapter 10: Applications of Marine Robotics

- Environmental monitoring: pollution detection, coral reef mapping.
- Offshore industries: pipeline inspection, underwater construction.
- Search and rescue operations.

Chapter 11: Advanced Topics in Marine Robotics

- Swarm robotics: coordination and communication in multi-robot systems.
- Autonomous decision-making and AI for marine robots.
- Emerging trends: soft robotics and bio-inspired designs.

Laboratory Work

Lab 1: Introduction to MATLAB for Marine Robotics

- **Objective:** Familiarize students with MATLAB and its toolboxes for marine robotics.
- **Tasks:**
    - Basics of MATLAB programming and scripting.
    - Introduction to Simulink for system modeling.
    - Visualization of marine data (e.g., plotting oceanographic data).

Lab 2: Hydrodynamic Modeling of Marine Vehicles

- **Objective:** Model the hydrodynamic behavior of marine vehicles.
- **Tasks:**
    - Simulate drag and lift forces using MATLAB.
    - Analyze the effect of hull shape on hydrodynamic performance.
    - Visualize fluid flow around a vehicle using MATLAB's fluid dynamics toolbox.

Lab 3: Propulsion System Design

- **Objective:** Design and simulate propulsion systems for marine robots.
- **Tasks:**
    - Model propeller thrust and torque using MATLAB.
    - Simulate the effect of different propeller designs on vehicle performance.
    - Analyze energy consumption and efficiency.

Lab 4: Underwater Vehicle Dynamics
- **Objective:** Simulate the dynamics of underwater vehicles.
- **Tasks:**
  - Implement the 6-DOF (Degrees of Freedom) equations of motion in MATLAB.
  - Simulate vehicle motion under different conditions (e.g., currents, waves).
  - Analyze stability and maneuverability.

Lab 5: Navigation and Localization
- **Objective:** Implement navigation algorithms for marine robots.
- **Tasks:**
  - Simulate dead reckoning using inertial measurement unit (IMU) data.
  - Implement acoustic positioning algorithms in MATLAB.
  - Visualize the vehicle's trajectory and position errors.

Lab 6: Sensor Integration and Data Fusion
- **Objective:** Integrate and process sensor data for marine robotics.
- **Tasks:**
  - Simulate sonar and depth sensor data in MATLAB.
  - Implement data fusion algorithms (e.g., Kalman filter) for improved localization.
  - Visualize sensor data and fused outputs.

Lab 7: Control System Design
- **Objective:** Design and simulate control systems for marine robots.
- **Tasks:**
  - Implement PID control for depth and heading control.
  - Simulate adaptive control strategies for dynamic environments.
  - Analyze control system performance under disturbances (e.g., waves, currents).

Lab 8: Underwater Communication Simulation
- **Objective:** Simulate underwater communication systems.
- **Tasks:**
  - Model acoustic communication channels in MATLAB.
  - Simulate data transmission and analyze latency and packet loss.
  - Implement error correction algorithms for reliable communication.

Lab 9: Path Planning and Obstacle Avoidance
- **Objective:** Implement path planning algorithms for marine robots.
- **Tasks:**
  - Use MATLAB's Robotics System Toolbox to implement A* or Dijkstra's algorithm.
  - Simulate obstacle avoidance using sensor data.
  - Visualize planned paths and vehicle motion.

Lab 10: Energy Management Simulation
- **Objective:** Simulate power and energy management for marine robots.
- **Tasks:**
  - Model battery discharge and charging cycles.
  - Simulate energy consumption for different mission profiles.
  - Optimize energy usage for long-duration missions.

Lab 11: Multi-Robot Coordination
- **Objective:** Simulate swarm robotics and multi-robot coordination.
- **Tasks:**
  - Implement coordination algorithms (e.g., leader-follower, consensus).
  - Simulate communication and task allocation in a multi-robot system.
  - Visualize swarm behavior in a marine environment.

Assessment Methods:
- Regular in class assessments (20%).
- Laboratory reports and practical assignments (20%).

- Final exam (60%).

Recommended Textbooks and Resources:

- Gianluca Antonelli, "Underwater Robots", springer 2nd edition, 2006
- Steven W. Moore, Harry Bohm, Vickie Jensen"Underwater Robotics: Science, Design & Fabrication" MATE center 2010, ISBN 978-0-9841737-0-9
- Thor I. Fossen "Handbook of Marine Craft Hydrodynamics and Motion Control" 2011 John Wiley & Sons, Ltd. ISBN:9781119991496 .DOI:10.1002/9781119994138
- Springer Handbook of robotics

Teaching Unit (UF)	Subject Title	Code	Semester
UEM3.1.1	**Data fusion**	**DFU**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30		1h00	**3/3**

Course Objectives:
This module introduces the principles and applications of multi-sensor data fusion. By the end of the course, students will:

- Understand the fundamentals of data fusion, its advantages, and typical use cases in embedded and robotic systems.
- Explore sensor models, data acquisition techniques, and preprocessing methods to handle noisy or incomplete data.
- Apply various data fusion techniques (e.g., Bayesian estimation, Kalman filtering) and assess different fusion architectures (centralized vs. distributed).
- Evaluate fusion performance using metrics and real-world case studies, identifying errors and limits in sensor fusion systems...

Recommended Prerequisites:
- Mathematics and Probability: Familiarity with basic statistics, probability theory, and linear algebra.
- Signal Processing: Understanding of filtering, noise models, and basic digital signal processing (DSP) concepts.
- Embedded Systems/Programming: Experience in reading sensor data (e.g., from microcontrollers) is helpful but not mandatory.

Course outline:
CHAPTER 1: Introduction to Multi-Sensor Data Fusion (02 weeks)
- Concepts and Definitions: Data fusion levels, static vs. dynamic fusion.
- Advantages and Applications: Redundancy, improved accuracy, robotics, autonomous vehicles, IoT.
- Examples of multi-sensor systems: Drone navigation, industrial monitoring..

CHAPTER 2: Sensor Models and Data Acquisition (02 weeks)
- Types of Sensors: Mechanical, thermal, optical, inertial, etc.
- Sensor Characteristics: Accuracy, resolution, sampling rate, noise profiles.
- Data Acquisition Techniques: Polling, interrupts, DMA; multi-sensor synchronization.

CHAPTER3: Data Preprocessing (02 weeks)
- Filtering Noisy Data: Low-pass, high-pass, band-pass, or adaptive filters.
- Sensor Calibration: Offset, scaling, linear/non-linear correction.
- Systematic Error Correction: Temperature drift, bias removal.

CHAPTER4: Data Fusion Techniques (04 weeks)
- Fusion Methods: Averaging, weighting, Bayesian estimation, Kalman filter, complementary filter.
- Mathematical Models: AR/MA processes, probability distributions, error modeling.
- Static vs. Dynamic Fusion Approaches: Batch vs. real-time updates.

CHAPTER5: Fusion Architectures (02 weeks)

- entralized vs. Distributed Fusion: Pros/cons, communication overhead.
- Hierarchical and Cascaded Architectures: Layered design for large systems.
- Synchronization: Timestamps, sensor clocks, dealing with out-of-sync data..

CHAPTER 6: Fusion Evaluation and Performance (02 weeks)

- Evaluation Metrics: RMSE, variance reduction, correlation, confusion matrix (for classification tasks).
- Case Studies: Real-world examples of sensor fusion (e.g., autonomous driving, wearable devices).
- Error Analysis and Limitations: Sensitivity to noise, outliers, missing data.

Laboratory Work:

- **Lab 1:** Sensor Data Acquisition and Noise Profiling.
- **Lab 2:** Preprocessing and Calibration.
- **Lab 3:** Basic Fusion Methods.
- **Lab 4:** Bayesian Estimation or Complementary Filter
- **Lab 5:** Kalman Filter Implementation.
- **Lab 6:** Fusion Architecture Evaluation.

Assessment Method:

- Laboratory reports and practical assignments (40%).
- Final exam (60%).

Recommended textbooks and Resources

- Mitchell, H.B. Data Fusion: Concepts and Ideas. Springer, 2012.
- Mitchell, H.B. Multi-Sensor Data Fusion: An Introduction. Springer, 2007.
- Klein, L.A. Sensor and Data Fusion: A Tool for Information Assessment and Decision Making. SPIE Press, 2004.
- Liggins, M., Hall, D., & Llinas, J. Handbook of Multisensor Data Fusion: Theory and Practice. CRC Press, 2009.
- Blasch, D. M., Blasch, E., & Cervone, D. Introduction to Data Fusion. Artech House, 2007.
- Celeux, C., & Govaert, G. Data Fusion Mathematics: Theory and Practice. Hermes Science Publications, 2010.

Teaching Unit (UF)	Subject Title	Code	Semester
UET3.1.1	**Entrepreneurship & Startup Development**	**ESTART**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30			**1/1**

Course Objectives:

By the end of this module, students will be able to:

- Understand the fundamentals of entrepreneurship and innovation in robotics and autonomous systems.
- Develop a viable business model for a robotics or autonomous systems startup.
- Create a comprehensive business plan and pitch it to potential investors.
- Navigate the legal, financial, and operational challenges of starting and running a tech-based company.
- Apply entrepreneurial thinking to solve real-world problems in robotics and autonomous systems.

Course Structure:

The module is divided into **4 main units**, each combining theoretical knowledge with practical application.

Unit 1: Introduction to Entrepreneurship in Robotics and Autonomous Systems (3 weeks)

Topics Covered:

1. **Entrepreneurship Fundamentals:**
   - What is entrepreneurship?
   - Characteristics of successful entrepreneurs.
   - The role of innovation in robotics and autonomous systems.
2. **Market Opportunities in Robotics and Autonomous Systems:**
   - Emerging trends and technologies.
   - Identifying market needs and gaps.
   - Case studies of successful robotics startups (e.g., Boston Dynamics, DJI, etc.).
3. **Idea Generation and Validation:**
   - Techniques for brainstorming and refining ideas.
   - Market research and customer validation.
   - Feasibility analysis for robotics projects.

Practical Work:

- Group brainstorming sessions to generate startup ideas.
- Conduct a mini-market research project to validate an idea.

Unit 2: Business Modeling and Planning (4 weeks)

Topics Covered:

1. **Business Model Canvas (BMC):**
   - Understanding the 9 building blocks of BMC.
   - Applying BMC to robotics and autonomous systems startups.
2. **Value Proposition Design:**
   - Defining the unique value of your product or service.
   - Aligning value propositions with customer needs.

3. **Business Plan Development:**
    - Key components of a business plan (executive summary, market analysis, financial projections, etc.).
    - Writing a compelling business plan for investors.
4. **Financial Planning and Funding:**
    - Basics of financial management for startups.
    - Funding options: bootstrapping, venture capital, angel investors, crowdfunding.
    - Creating financial projections and break-even analysis.

Practical Work:

- Develop a Business Model Canvas for a robotics startup idea.
- Create a draft business plan and present it to peers for feedback.

Unit 3: Legal, Operational, and Strategic Challenges (3 weeks)

Topics Covered:

1. **Legal and Regulatory Considerations:**
    - Intellectual property (IP) protection: patents, trademarks, and copyrights.
    - Choosing the right legal structure for your startup (LLC, corporation, etc.).
    - Compliance with industry regulations (e.g., safety standards for robotics).
2. **Operational Planning:**
    - Building a team: recruitment, roles, and responsibilities.
    - Supply chain and production planning for hardware/software products.
    - Risk management and contingency planning.
3. **Scaling and Growth Strategies:**
    - Strategies for scaling a robotics startup.
    - Partnerships, collaborations, and networking.
    - International expansion and global markets.

Practical Work:

- Research and present a case study on a robotics startup's legal or operational challenges.
- Develop a risk management plan for a hypothetical startup.

Unit 4: Pitching and Launching Your Startup (4 weeks)

Topics Covered:

1. **Pitching to Investors:**
    - Crafting a compelling pitch deck.
    - Storytelling techniques for technical products.
    - Handling investor questions and objections.
2. **Marketing and Sales Strategies:**
    - Building a brand for a robotics startup.
    - Digital marketing strategies for tech products.
    - Sales strategies for B2B and B2C markets.
3. **Launching Your Startup:**
    - Minimum Viable Product (MVP) development.
    - Beta testing and customer feedback.
    - Planning a successful product launch.

Practical Work:

- Create and present a pitch deck for a robotics startup idea.
- Develop a marketing plan for a robotics product.
- Simulate a product launch event.

Assessment Methods:

- Final exam (100%).

Teaching Unit (UF)	Subject Title	Code	Semester
UET3.1.1	**Quality Control**	**QCN**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30			**1/1**

Course Objectives:

1. To understand the fundamental principles and importance of quality control in engineering and manufacturing.
2. To explore the tools and techniques used for quality assurance and improvement.
3. To analyze the role of Total Quality Management (TQM) and Statistical Process Control (SPC) in achieving quality.
4. To examine quality systems and standards (e.g., ISO 9001, Six Sigma).
5. To study the historical evolution and current trends in quality control.

Course Outline:

Chapter 1: Introduction to Quality Control (1 week)

- Definition and importance of quality control.
- Historical development of quality control: From craftsmanship to modern quality systems.
- Key concepts: Quality assurance, quality control, and quality management.

Chapter 2: Fundamentals of Quality Management (1 week)

- The cost of quality: Prevention, appraisal, and failure costs.
- Quality dimensions: Performance, reliability, durability, and customer satisfaction.
- Introduction to Total Quality Management (TQM): Principles and benefits.

Chapter 3: Quality Tools and Techniques (1 week)

- Seven basic quality tools:
    - Check sheets, Pareto charts, cause-and-effect diagrams, histograms, scatter diagrams, control charts, and flowcharts.

Chapter 4: Statistical Process Control (SPC) (2 weeks)

- Introduction to SPC: Objectives and benefits.
- Control charts: X-bar and R charts, p-charts, and c-charts.
- Process capability analysis: Cp, Cpk, and process performance indices.

Chapter 5: Total Quality Management (TQM) (1 week)

- Key principles of TQM: Customer focus, continuous improvement, and employee involvement.
- Implementation of TQM: Steps and challenges.

Chapter 6: Quality Systems and Standards (1 week)

- Overview of quality systems: ISO 9001, Six Sigma, and Lean Manufacturing.
- ISO 9001: Requirements and certification process.
- Six Sigma: DMAIC methodology and tools.

Chapter 7: Quality in Design and Development (1 week)

- Design for Quality (DFQ): Principles and practices.
- Quality Function Deployment (QFD): Translating customer needs into design requirements.
- Failure Mode and Effects Analysis (FMEA): Identifying and mitigating risks.

Chapter 8: Quality Control in Manufacturing (1 week)

- Inspection and testing: Methods and techniques.
- Sampling plans: Acceptance sampling and AQL (Acceptable Quality Level).
- Non-destructive testing (NDT): Techniques and applications.

Chapter 9: Quality in Supply Chain Management (1 week)

- Supplier quality management: Evaluation and certification.
- Quality audits: Types and processes.

Chapter 10: Continuous Improvement and Lean Principles (1 week)

- Kaizen: Continuous improvement philosophy.
- Lean manufacturing: Principles and tools (e.g., 5S, value stream mapping).

Chapter 11: Current Trends in Quality Control (2 weeks)

- Digital quality control: Use of IoT, AI, and machine learning.
- Industry 4.0 and smart manufacturing: Impact on quality control.
- Sustainability and quality: Green manufacturing and circular economy.

Assessment Methods:

- Final exam (100%).

Recommended Textbooks and Resources:

1. **"Quality Control" by Dale H. Besterfield** – A comprehensive guide to quality control principles and practices.
2. **"Total Quality Management" by John S. Oakland** – In-depth coverage of TQM concepts and implementation.
3. **"Statistical Quality Control" by Douglas C. Montgomery** – Detailed exploration of SPC and process improvement.
4. **ISO 9001 Standards** – Official documentation for quality management systems.

Teaching Unit (UF)	Subject Title	Code	Semester
UED3.1.1	**Service robotics**	**SROB**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30			**1/1**

Course Overview:
This course introduces students to the principles, technologies, and applications of service robotics. Service robots are designed to assist humans in various tasks, ranging from domestic chores to healthcare and public services. The course covers the design, development, and deployment of service robots, with a focus on human-robot interaction, perception, navigation, and task planning. Students will gain hands-on experience through projects and case studies, using tools like ROS (Robot Operating System) and simulation environments.

Course Objectives:
At the end of this course, students should be able to:
- Understand the fundamentals of service robotics and their applications.
- Classify service robots based on their functions and domains.
- Analyze the design, control, and operation of service robots in various industries.
- Address challenges related to safety, ethics, and human-robot interaction.

Recommended Prerequisites:
1. Programming Skills: (in Python /C++).
2. Basic understanding of robotics concepts (kinematics, sensors, actuators).
3. Familiarity with control systems (e.g., PID controllers).
4. Basics of image processing and object recognition (e.g., OpenCV).
5. Experience with ROS (Robot Operating System) and simulation tools like Gazebo.

Course Outline:
Chapter 1: Introduction to Service Robotics (2 weeks)
- Definition and characteristics of service robots
- Difference between industrial and service robots
- Overview of standards and regulations (ISO 13482, safety guidelines)
- Ethical considerations in service robotics

Chapter 2: Classification of Service Robots (1 week)
- Classification of service robots by type of movement
- Classification of service robots by application
- Personal robots (assistive robots, household robots)
- Professional service robots (medical, agricultural, logistics, etc.)
- Public service robots (security, emergency response, customer interaction)
- Autonomous vs semi-autonomous robots

Chapter 3: Robots for Agricultural and Farming Applications (1 weeks)
- Autonomous tractors and robotic harvesters
- Drones for precision agriculture and crop monitoring
- Automated irrigation and soil analysis systems
- Case studies: John Deere, Naïo Technologies, Agrobot

Chapter 4: Robots for Professional Cleaning Applications (1 week)

- Automated floor cleaning robots (e.g., Roomba, Scrubbers)
- Industrial cleaning robots for hazardous environments
- Disinfection and sanitation robots (UV disinfection)
- Hands-on lab: Simulation of a cleaning robot using ROS

Chapter 5: Robots for Inspection and Maintenance (2 weeks)
- Infrastructure inspection (bridges, pipelines, power lines)
- Robots for hazardous environment monitoring (nuclear plants, deep-sea)
- AI-based fault detection and predictive maintenance
- Case studies: Boston Dynamics Spot, ANYmal robot

Chapter 6: Mobile Robots for Transportation & Logistics (2 weeks)
- AGVs (Automated Guided Vehicles) and AMRs (Autonomous Mobile Robots)
- Warehouse robots (Amazon Robotics, Kiva Systems)
- Delivery drones and last-mile delivery robots
- Hands-on lab: Path planning for a mobile robot in a logistics warehouse

Chapter 7: Robots in Medical Applications (2 weeks)
- Surgical robots (e.g., Da Vinci Robot)
- Rehabilitation robots (exoskeletons, prosthetics)
- Hospital service robots (medication delivery, patient care assistance)
- Case studies: TUG robots, ReWalk, Medtronic's Mazor X

Chapter 8: Robots for Emergency Situations (1 week)
- Search and rescue robots (drones, quadrupeds)
- Firefighting and disaster response robots
- Robots for hazardous material handling
- Case studies: PackBot, Colossus, RoboBees

Chapter 9: Robots for Interaction with Guests or Visitors (1 week)
- Humanoid robots in customer service (e.g., Pepper, Nao)
- AI-powered chatbots and interactive kiosks
- Museum and hotel assistant robots (Connie, Temi, LG CLOi)
- Hands-on lab: Programming a chatbot for a robotic receptionist

Chapter 10: AI & Autonomy in Service Robotics (1 week)
- Computer vision for object recognition and tracking
- Natural Language Processing (NLP) for human-robot interaction
- Reinforcement learning for adaptive behavior
- Case studies of AI-powered service robots

Assessment Methods:
- Final exam (100%).

Recommended Textbooks and Resources:
- Textbook: "Robotics, Vision and Control: Fundamental Algorithms in MATLAB" by Peter Corke.
- Textbook: "Programming Robots with ROS: A Practical Introduction to the Robot Operating System" by Morgan Quigley, Brian Gerkey, and William D. Smart.
- Access to ROS and simulation tools (e.g., Gazebo, RViz).
- "Human-Robot Interaction: An Introduction" by Christoph Bartneck, Tony Belpaeme, and Friederike Eyssel.
- "Service Robots and Robotics: Design and Application" by Marco Ceccarelli.
- Research papers on service robotics applications and technologies.

Teaching Unit (UF)	Subject Title	Code	Semester
UED3.1.1	**Cybersecurity**	**CYBER**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH	1h30			**1/1**

Course Objectives:
By the end of this module, students will be able to:
1. Understand Cybersecurity Fundamentals:
2. Apply Cryptographic Techniques:
3. Secure Networked Systems:
4. Protect Embedded and Robotic Systems:
5. Address Autonomous Vehicle Security:
6. Secure IoT and Edge Computing Systems:
7. Develop Practical Cybersecurity Skills:

Course outline:
Chapter 1: Introduction to Cybersecurity (1 week)
- **Objectives**:
  - Understand the importance of cybersecurity in robotics and autonomous systems.
  - Learn the basic concepts of cybersecurity: confidentiality, integrity, availability.

Chapter 2: Fundamentals of Cryptography (2 weeks)
- **Objectives**:
  - Understand the role of cryptography in securing data and communications.
  - Learn about symmetric and asymmetric encryption algorithms.
  - Apply cryptographic techniques to protect data in robotic systems.

Chapter 3: Network Security for Autonomous Systems (1 week)
- **Objectives**:
  - Understand the vulnerabilities in networked robotic systems.
  - Learn how to secure communication channels and protocols.

Chapter 4: Embedded Systems Security (2 weeks)
- **Objectives**:
  - Identify security risks in embedded systems used in robotics.
  - Learn techniques to secure firmware and hardware.
  - Understand secure boot and trusted execution environments.

Chapter 5: Cybersecurity in Robotics (2 weeks)
- **Objectives**:
  - Analyze the unique cybersecurity challenges in robotics.
  - Learn how to secure robotic operating systems (ROS).

Chapter 6: Autonomous Vehicles Security (2 weeks)
- **Objectives**:
  - Understand the cybersecurity risks in autonomous vehicles.
  - Learn how to secure vehicle-to-everything (V2X) communication.
  - Implement security measures for autonomous driving systems.

Chapter 7: IoT and Edge Computing Security (1 week)

- **Objectives**:
  - Understand the security risks in IoT and edge computing systems.
  - Learn how to secure IoT devices and edge nodes.

Chapter 8: Cybersecurity Risk Management (1 week)

- **Objectives**:
  - Learn how to assess and manage cybersecurity risks in robotic systems.
  - Understand the principles of cybersecurity risk management frameworks.
  - Develop a risk mitigation plan for autonomous systems.

Chapter 9: Ethical Hacking and Penetration Testing (1 week)

- **Objectives**:
  - Learn the basics of ethical hacking and penetration testing.
  - Understand how to identify and exploit vulnerabilities in robotic systems.
  - Develop skills to defend against cyberattacks.

Chapter 10: Legal and Ethical Aspects of Cybersecurity (1 week)

- **Objectives**:
  - Understand the legal and ethical implications of cybersecurity.
  - Learn about cybersecurity laws and regulations.
  - Develop an ethical framework for securing autonomous systems.

Assessment Methods:
- Final exam (100%).

Recommended Textbooks and Resources
Textbooks

1. **"Cybersecurity for Robotics and Autonomous Systems"** by Dietmar P.F. Möller and Roland E. Haas.
2. **"Applied Cryptography: Protocols, Algorithms, and Source Code in C"** by Bruce Schneier.
3. **"Network Security Essentials: Applications and Standards"** by William Stallings.
4. **"Embedded Systems Security: Practical Methods for Safe and Secure Software and Systems Development"** by David Kleidermacher and Mike Kleidermacher.
5. **"The IoT Hacker's Handbook"** by Aditya Gupta.
6. **"Cybersecurity and Cyberwar: What Everyone Needs to Know"** by P.W. Singer and Allan Friedman.

Online Resources

1. **OWASP (Open Web Application Security Project)**:
   - Website: https://owasp.org
2. **NIST Cybersecurity Framework**:
   - Website: https://www.nist.gov/cyberframework
3. **Kali Linux Tools**:
   - Website: https://www.kali.org
4. **Coursera and edX Cybersecurity Courses**:
   - Examples: "Cybersecurity Fundamentals" (edX), "Introduction to Cyber Security" (Coursera).
5. **ROS Security Documentation**:
   - Website: https://wiki.ros.org/Security

Teaching Unit (UF)	Subject Title	Code	Semester
UED3.1.1	**Training Internship III**	**TI-III**	**S5**

	Lectures	TUTORIAL	LABS	Crédit/ Coeff
VHH				**1/1**

Course Objectives:

- Develop expertise in an area of specialization.
- Take ownership of a project or task within the company.
- Prepare students for career readiness and industry expectations.

Targeted Skills

- Leadership and initiative.
- Complex problem-solving.
- Professional networking.
- Industry-standard technical competencies.

Recommended prerequisites

- Advanced understanding of Autonomous systems, robotics and control strategies.
- Experience with project execution and troubleshooting.

Content

1. **Advanced Project Engagement (6 hours)**
   - Working on a specific problem-solving project.
   - Coordinating with team members on deliverables.
2. **Independent Task Management (6 hours)**
   - Taking responsibility for assigned tasks.
   - Implementing improvements and optimizations.
3. **Industry Networking and Career Preparation (3 hours)**
   - Engaging with professionals and industry leaders.
   - Learning about career pathways and opportunities.
4. **Final Presentation and Evaluation (3 hours)**
   - Presenting project outcomes to company and faculty.
   - Self-assessment and feedback integration.

Recommended Textbooks and Resources

1. Internship & Career Handbook by Lauren Berger
2. The Start-Up of You by Reid Hoffman

Detailed programs for the tenth semester

Teaching Unit (UE)	Subject Title	Code	Semester
	Internship and Final Year Project	**Final**	**S10**

			Crédit/ Coeff
VHH	14 Weeks	1h30	**30/30**

Objectives

The Internship and Final Year Project serve as critical educational components designed to consolidate academic knowledge while developing professional competencies. These experiences provide students with structured opportunities to apply theoretical concepts in practical settings, develop technical problem-solving abilities, and demonstrate their capacity for independent work. Through these professional experiences, students refine their analytical skills, enhance their project management capabilities, and establish meaningful industry connections that facilitate their transition from academia to professional practice.

The Final Year Project focuses on guiding engineering students toward addressing a specific engineering and/or research challenge. This culminating project is partially conducted within a company or laboratory setting in the form of an internship, creating a seamless integration between academic requirements and real-world professional experience. This approach ensures graduates possess the comprehensive skill set required by employers in the autonomous systems sector while providing them with authentic exposure to industry practices and methodologies.

Evaluation Method

- **Overall Module Evaluation Formula**

Module Average = 20% Internship Assessment + 80% Final Year Project

- **Final Year Project Grading Criteria**

1. Manuscript Quality (Document Formatting, Writing Quality)
2. Oral Presentation (Speech Quality and Delivery, Slide Quality and Visual Presentation, Understanding and Response to Questions)
3. Supervisor Assessment (Student Engagement and Commitment)
4. Innovation (Novelty and Originality of Solutions Provided)

Note: The detailed evaluation criteria for the Final Year Project will be defined by the school's scientific committees.

Agreements and conventions

Decisions and Approvals of Administrative and Advisory Bodies

Department

Scientific Board

School Direction