Bachelor of Engineering

Robotics Engineering

Bachelor of Engineering – Robotics Engineering

MISSION STATEMENT

*The **Bachelor of Engineering in Robotics Engineering** program at SRU is dedicated to cultivating visionary thinkers and designers in the field of robotics and artificial intelligence. Through scientific excellence, interdisciplinary exploration, and creative freedom, our graduates shape the future of automation and innovation.*

At Springfield Research University, our **Bachelor of Engineering in Robotics Engineering** program equips students for dynamic careers in the field of robotics. Our mission is built upon three fundamental pillars:

1. **Academic Excellence:**
    - We uphold rigorous standards, fostering critical thinking and intellectual growth.
    - Through engaging coursework, practical training, and evidence-based practice, we empower students to excel in the complex world of robotics engineering.
    - Students gain a solid foundation in robotics principles, control systems, artificial intelligence, and mechatronics.
2. **Cutting-Edge Research:**
    - Our faculty and students actively contribute to advancing robotics technology.
    - By addressing real-world challenges, exploring innovative design concepts, and shaping industry practices, we drive positive change within the field.
    - Students engage in research projects, simulations, and experiments, enhancing their ability to provide innovative solutions.
3. **Societal Impact:**
    - We recognize our responsibility to society and the environment.
    - Our graduates are not only skilled engineers but also ethical leaders who advocate for safety, sustainability, and global connectivity.
    - We empower them to make meaningful contributions to automation, healthcare, manufacturing, and environmental conservation.

The mission of the **Bachelor of Engineering in Robotics Engineering** is to provide a comprehensive education that prepares graduates for productive careers and responsible citizenship, with special emphasis on the needs of robotics, automation, and related fields. The program's focus is primarily on designing and developing mission-oriented robotic systems for various applications. Here's how we achieve this:

1. **Foundational Sciences:**
    - Students delve into core subjects such as control theory, kinematics, and sensors. These foundational sciences provide the essential groundwork for understanding robotic mechanisms, motion planning, and sensing technologies.
2. **Applied Correlations:**
    - Lectures and practical sessions correlate theoretical knowledge with real-world robotics scenarios. For example, students learn about robot dynamics and immediately apply it during hands-on exercises using robotic arms and simulators.
3. **Case-Based Learning:**
    - Real-world robotic cases serve as powerful teaching tools. Students analyze robotic system requirements, design specifications, and safety considerations. This approach bridges theory and practice, reinforcing robotics concepts.
4. **Simulated Robotic Experiences:**

- Simulation labs replicate actual robotic environments. Students practice programming robot behaviors, troubleshooting, and handling safety protocols, honing their skills before working with physical robots.

5. **Industry Collaborations and Internships:**
    - During industry placements and internships, students work directly with robotics professionals.
    - They apply theoretical knowledge in designing, testing, and maintaining robotic systems for manufacturing, logistics, and automation.
6. **Cutting-Edge Research and Innovation:**
    - Students critically evaluate research articles, explore emerging technologies (such as machine learning for robotics), and contribute to advancements in robotics.
    - This integration of evidence-based practices ensures that knowledge acquisition aligns with current best practices in the robotics industry.

By seamlessly weaving theory, practical experiences, and evidence-based approaches, our program prepares graduates to contribute effectively to the dynamic field of robotics engineering. The mission of the robotics engineering program at SRU is to help students acquire scientific skills, and use them to good effect: while some people merely code, control or monitor robots and drones, the program will turn the student into a visionary thinker, designer and trendsetter in the world of robotics and artificial intelligence, with plenty of freedom and scope for action in their professional life.

RATIONALE FOR THE BACHELOR OF ENGINEERING IN ROBOTICS ENGINEERING

At Springfield Research University, our **Bachelor of Engineering in Robotics Engineering** program is meticulously crafted to prepare students for impactful careers in the field of robotics. Rooted in academic excellence, this program equips students with essential knowledge, practical skills, and hands-on experience. By emphasizing evidence-based practices and innovation, our graduates emerge as competent professionals poised to shape the future of automation, artificial intelligence, and robotics.

National Needs (Eswatini):

1. **Quantitative Expertise:**
    - Eswatini demands skilled robotics professionals who can navigate complex scenarios in robot design, control systems, and safety.
    - The program equips students with mathematical proficiency and critical thinking abilities to assess robotic conditions effectively.
2. **Cutting-Edge Practices:**
    - Graduates advocate for evidence-based decision-making, ensuring safety, efficiency, and equitable treatment in robotics industries.
    - By enhancing their understanding of kinematics, sensors, and machine learning, they contribute to better robotic outcomes.
3. **Policy and Innovation:**
    - The program fosters critical thinking, enabling graduates to engage in research, policy formulation, and informed decision-making at the national level.

Regional Needs (SADC):

1. **Harmonization of Practices:**
    - SADC member states face common robotics challenges.

   - The program aligns with SADC's goal of harmonizing robotics frameworks, promoting cooperation, and advancing automation practices.
2. **Human Capital Development:**
   - Robotics professionals play a pivotal role in regional industries, transportation, and communication systems.
   - The program contributes to building a skilled workforce capable of addressing cross-border robotics complexities.
3. **Technological Advancements:**
   - SADC's prosperity relies on informed robotics practices.
   - Our graduates contribute to maintaining order in robotic systems, resolving challenges, and fostering regional well-being.

Purpose of the Program:

1. **Technical Leadership:**
   - The program educates ethical leaders who champion evidence-based practices, fairness, and safety in robotics engineering.
   - Graduates not only assess technical data but also shape policies, regulations, and protocols.
2. **Innovative Research:**
   - Students engage in specialized robotics research, addressing contemporary challenges and contributing to technological advancements.

Program Educational Objectives (PEOs) for Robotics Engineering Graduates:

1. **Professional Careers:**
   - Graduates will establish successful careers in robotics and related industries, academia, or government agencies.
   - They will contribute to the design, development, and deployment of robotic systems, automation solutions, and intelligent technologies.
2. **Research and Development:**
   - Graduates will actively engage in research, pushing the boundaries of robotics knowledge.
   - They will explore novel algorithms, sensor technologies, and control strategies to advance the field.
3. **Entrepreneurial Endeavors:**
   - Graduates will demonstrate an entrepreneurial mindset, identifying opportunities to create innovative robotic products or services.
   - They may launch startups, collaborate with industry partners, or lead their own ventures.
4. **Lifelong Learning and Advanced Degrees:**
   - Graduates will recognize the importance of continuous learning.
   - They will pursue advanced degrees (such as master's or doctoral programs) to deepen their expertise and contribute to cutting-edge research.

5. **Leadership in Multidisciplinary Projects:**
    - Graduates will take leadership roles in diverse projects involving robotics, collaborating with engineers, designers, and domain experts.
    - They will navigate complex challenges, manage teams, and drive project success.
6. **Ethical and Responsible Professionals:**
    - Graduates will uphold ethical standards in their work.
    - They will actively participate in community service, mentorship, and professional guidance, ensuring the responsible use of robotics technology.
7. **Effective Communication:**
    - Graduates will communicate effectively with colleagues, stakeholders, and the public.
    - They will convey technical concepts clearly, advocate for their ideas, and bridge the gap between technology and society.

Program Learning Outcomes (PLOs) for Robotics Engineering Graduates:

1. **Problem Solving and Analysis:**
    - Graduates will demonstrate the ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
    - They will analyze robotic challenges, design innovative solutions, and evaluate their effectiveness.
2. **Engineering Design and Societal Impact:**
    - Graduates will apply engineering design principles to produce solutions that meet specified needs.
    - They will consider public health, safety, welfare, as well as global, cultural, social, environmental, and economic factors in their designs.
3. **Effective Communication:**
    - Graduates will communicate effectively with diverse audiences.
    - They will convey technical information clearly through reports, presentations, and collaborative discussions.
4. **Ethical and Professional Responsibilities:**
    - Graduates will recognize ethical and professional responsibilities in engineering situations.
    - They will make informed judgments, considering the impact of engineering solutions in global, economic, environmental, and societal contexts.
5. **Collaboration and Leadership:**
    - Graduates will function effectively in multidisciplinary teams.
    - They will provide leadership, create inclusive environments, establish goals, plan tasks, and achieve objectives collaboratively.
6. **Experimental Skills and Data Analysis:**
    - Graduates will develop and conduct appropriate experiments related to robotics.
    - They will analyze and interpret data, using engineering judgment to draw meaningful conclusions.

7. **Lifelong Learning and Adaptability:**
    - Graduates will acquire and apply new knowledge as needed.
    - They will employ appropriate learning strategies to stay current in the rapidly evolving field of robotics.
8. **Integration of Cyberphysical Systems:**
    - Graduates will evaluate and integrate the mechanical, electrical, and computational components of cyberphysical systems.
    - They will design cohesive systems that seamlessly combine hardware and software for practical applications.
9. **Entrepreneurial Mindset:**
    - Graduates will recognize and take advantage of entrepreneurial opportunities.
    - They may explore startup ventures, innovative projects, or industry collaborations.

CAREER OPPORTUNITIES

1. **Robotics Engineer:**
    - Design, develop, and maintain robotic systems for various applications, including manufacturing, healthcare, and exploration.
    - Specialize in areas such as control algorithms, computer vision, and motion planning.
2. **Electromechanical and Robotics Technician:**
    - Install, troubleshoot, and repair robotic equipment.
    - Collaborate with engineers to ensure smooth operation of robotic systems.
3. **Mechanical Engineer:**
    - Apply mechanical principles to design and analyze robotic components.
    - Work on projects related to robotics, automation, and mechatronics.
4. **Design Engineer:**
    - Create detailed designs for robotic systems, considering functionality, safety, and manufacturability.
    - Use CAD software to model and simulate robot components.
5. **Software Engineer:**
    - Develop software for controlling robots, implementing algorithms, and integrating sensors.
    - Optimize code for real-time performance and reliability.
6. **Hardware Engineer:**
    - Design and test electronic components used in robotic systems.
    - Collaborate with software engineers to ensure seamless integration.
7. **Sales Engineer:**

- Combine technical knowledge with sales skills to promote robotic products and solutions.
- Understand customer needs and provide tailored recommendations.

8. **Aerospace Engineer:**
    - Apply principles of robotics to aerospace systems, such as drones, satellites, and space exploration vehicles.
    - Work on avionics, propulsion, and autonomous navigation.
9. **Computer Scientists:**
    - Research and develop algorithms for robotics, machine learning, and artificial intelligence.
    - Contribute to advancements in perception, localization, and decision-making.
10. **User Interface/User Experience (UI/UX) Designer:**
    - Create intuitive interfaces for robot control panels, mobile apps, and web-based interfaces.
    - Focus on user experience, accessibility, and efficient interaction with robotic systems.

ENTRY REQUIREMENTS

The student must have 6 passes in SGCSE/GCE/IGCSE O' level including a pass with Grade C or better in English Language and at least four other subjects. Special: Mathematics and any other two from chemistry, Combined Science, Physics, Physical Science. Faculty may set mature entry requirements subject to approval by Senate.

The Bachelor's Degree shall:

The Bachelor's degree program in Robotics Engineering at Springfield Research University is designed to equip students with the skills and knowledge necessary for a successful career in this dynamic field. Here are the key features of our program:

1. **Duration**:
    - The program spans **four years** for full-time students or **six years** for part-time students, including an industrial attachment or internship period.
2. **Semester Structure**:
    - Each academic year consists of **two semesters**.
    - **Semester Duration**: Each semester runs for **20 weeks**.
        - **Orientation Week**: One week dedicated to orientation.
        - **Teaching Weeks**: A minimum of **14 weeks** for instruction.

- **Mid-Semester Break**: A one-week break for students.
- **Examination Period**: Two weeks for final exams.
- **Results Processing**: Two weeks allocated for marking and result processing.

Our program ensures a rigorous academic experience while allowing flexibility for part-time students. Students engage in hands-on learning, theoretical coursework, and practical projects, preparing them for the exciting challenges of the robotics industry.

Special Departmental Regulations

1. **Course Completion Requirements**:
   - All **Core**, **Prerequisite**, **Required**, **General**, and **Elective** courses within the degree program are compulsory. Students must pass these courses with a minimum grade of **50%** to graduate.
   - However, during the third and fourth years, all courses must be passed with a minimum grade of **60%** (equivalent to a CGPA of **3.00**) to qualify for graduation.
2. **Optional Courses**:
   - Optional courses do not contribute to the final grade. Their marks are excluded from the computation of the overall grade.
3. **Externalization of Courses**:
   - All courses within the degree programs must be completed internally. Externalization is not permitted.
4. **Quality Control and Evaluation**:
   - Regular academic audits and reviews occur every four years, overseen by external moderators. Internal program evaluation is ongoing.
5. **Competence and Preparation**:
   - The courses offered in the Bachelor of Engineering in Robotics Engineering program provide adequate competences, preparing students for professional practice at the required academic level.
6. **Core and Prerequisite Courses**:
   - Students must pass all Core and Prerequisite courses with a minimum grade of **50%** before progressing to the next level or enrolling in additional courses.

Degree Award and Classification

- Upon successful completion of all **Core**, **Required**, and **Education** courses, as well as meeting the program requirements, a student will be awarded the degree of **Bachelor of Engineering in Robotics Engineering** at the end of the final year.
- The **normal classification** of a Bachelor's Degree is determined based on the academic performance during the third and fourth years of study.

Rationale to Course Numbering

At Springfield Research University, we meticulously design our Robotics Engineering curriculum to empower students with the knowledge and skills needed to thrive in this dynamic field. Our course numbering system serves as a roadmap, guiding students through their academic journey - **100-level courses** introduce foundational concepts. - **200-level courses** build on those foundations. - **300-level courses** explore more specialized topics. - **400-level courses** are advanced and often include research or project components. Let's delve into the reasons behind our thoughtful approach:

1. **Logical Progression**: Our course numbers reflect a logical progression. Foundational concepts begin with the "100" series, followed by deeper explorations in the "200" and "300" levels. Advanced topics and research opportunities reside in the "400" series.
2. **Prerequisites and Coherence**: Clear numbering helps students understand prerequisites and co-requisites. For instance, a 200-level course assumes knowledge from related 100-level courses, ensuring a coherent learning experience.
3. **Specialization and Depth**: As students advance, higher-level courses delve into specialized areas such as control systems, machine learning, and autonomous robotics. The numbering system communicates this depth of study.
4. **Alignment with Program Goals**: Each course number aligns with our program's learning outcomes. Whether it's mastering kinematics or diving into image processing, students can track their progress.
5. **Transferability**: Consistent numbering facilitates credit transfer between institutions, supporting seamless academic mobility.

In summary, our course numbering isn't just a sequence—it's a deliberate framework that enhances learning, fosters curiosity, and prepares our students for impactful careers in robotics engineering. Robotics Engineering courses simplifies the course numbering system.

1. **100-Level Courses**:
   - **ROB 101**: Introduction to Robotics Engineering
   - **ROB 110**: Linear Algebra for Robotics
   - **ROB 120**: Mechanics and Dynamics
2. **200-Level Courses**:
   - **ROB 201**: Robot Kinematics and Dynamics
   - **ROB 210**: Digital Electronics
   - **ROB 220**: Materials Science for Robotics
3. **300-Level Courses**:
   - **ROB 301**: Control Systems for Robots
   - **ROB 310**: Programming Embedded Systems
   - **ROB 320**: Sensors and Perception
4. **400-Level Courses**:

- **ROB 401**: Machine Learning in Robotics
- **ROB 410**: Robotic Vision and Image Processing
- **ROB 420**: Autonomous Robotics

The Bachelor of Engineering is a four (4) program. The student is expected to accumulate 576 credit points to be considered to have met the requirements of the Bachelor of Engineering in Robotics Engineering and must pass each module by at least 50%.

- Level 1 = minimum of credits 144 (1440 notional hours of study)
- Level 2 = minimum of credits 144 (1440 notional hours of study)
- Level 3 = minimum of credits 144 (1440 notional hours of study)
- Level 4 = minimum of credits 144 (1440 notional hours of study)

TOTAL credit points 576 (5760 notional hours of study)

Credit Transfer and Accumulation

1. Credits are derived from engagement of students in learning activities during lectures, seminars, tutorials, micro or macro field trips, directed and self-directed learning and writing examination tests and assignments.
2. Modules from the engineering faculty are worth 12 credit. Lecture attendance is compulsory. Students who attend less than 80% of lessons will not be allowed to sit for their sessional examinations.

Weighting

The degree class shall be based on weighting the results from part 1, 2, 3, and 4, the Degree weighting shall be as follows:

Level 1	20%
Level 2	20%
Level 3	30%
Level 4	30%

Distribution of Notional Hours

Module	Lecture Hrs	Tutorials/ Seminars	Self-Directed Study	Assignment Tests/Exams	Notional Hrs	Credits
ROB000	36	24	30	30	120	12
PROJECT	0	0	60	60	120	12

ASSESSMENT METHODS

1. **Formative Assessment (30%)**:
   - **Class Participation**: Actively engage in discussions, seminars, and practical activities related to robotics engineering.

- **Quizzes and Short Tests**: Regular assessments on specific robotics engineering topics.
- **Draft Assignments**: Receive feedback on early assignment drafts related to robotics engineering principles.
- **Peer Review**: Collaborate with peers to review and improve each other's engineering project work.

2. **Summative Assessment (60%)**:
   - **Final Examinations**: Comprehensive exams covering course content specific to robotics engineering.
   - **End-of-Semester Projects**: Assess students' knowledge and problem-solving skills related to robotics engineering challenges.
   - **Oral Presentations**: Evaluate communication abilities within the context of robotics engineering solutions.
   - **Engineering Design Competitions**: Simulate real-world robotics engineering scenarios.
3. **Continuous Assessment (10%)**:
   - **Internships or Work Placements**: Engage in supervised robotics engineering placements, applying theoretical knowledge to practical projects.
   - **Assignments and Projects**: Regular tasks contribute to the overall grade, emphasizing practical skills in robotics engineering design and analysis.
   - **Research Papers**: Demonstrate research abilities related to robotics engineering innovations and advancements.
   - **Attendance and Active Participation**: Engage in lectures, workshops, and industry events specific to robotics engineering practices.

These adapted assessment methods align with Springfield Research University's commitment to academic excellence and the development of competent engineers.

Teaching Methods

At Springfield Research University (SRU), we are committed to employing a diverse array of teaching methods to ensure a comprehensive and engaging learning experience for our students. Our teaching methods are carefully selected to align with the programme's objectives and to meet the needs of our diverse student body. The following are the key teaching methods utilized across all SRU programmes:

1. Lectures:

- Lectures are used to introduce and explain key concepts, theories, and principles. They provide a structured and systematic approach to delivering content, allowing students to gain a solid foundation in their respective fields. Lectures are often supplemented with visual aids, multimedia presentations, and interactive elements to enhance understanding and engagement.

2. Seminars:

- Seminars are interactive sessions that promote critical thinking and in-depth discussion on specific topics. Students are encouraged to actively participate, share their perspectives, and engage in debates. Seminars provide an opportunity for students to develop their analytical and communication skills.

3. Workshops:

- Workshops are hands-on sessions that focus on practical skills and applications. These sessions allow students to engage in experiential learning, apply theoretical knowledge to real-world scenarios, and collaborate with peers on projects and activities. Workshops are designed to foster creativity, problem-solving, and teamwork.

4. Problem-Based Learning (PBL):

- Problem-Based Learning is a student-centered approach that involves presenting students with complex, real-world problems to solve. Students work in small groups to research, discuss, and propose solutions, developing critical thinking and collaborative skills in the process. PBL encourages independent learning and active engagement.

5. Case Studies:

- Case studies are used to analyze real-life situations and decision-making processes. Students examine and discuss case studies to understand the context, identify key issues, and evaluate possible solutions. This method helps students develop their analytical and problem-solving abilities while relating theoretical concepts to practical situations.

6. Clinical Practice:

- For programmes that include a clinical component, such as Health and Medical Sciences, clinical practice is an integral part of the curriculum. Students gain hands-on experience in clinical settings, working under the supervision of qualified professionals. This method provides valuable opportunities for students to apply their knowledge, develop clinical skills, and gain insights into professional practice.

7. Research Projects:

- Research projects are designed to cultivate a culture of inquiry and innovation. Students engage in independent or group research projects, exploring topics of interest and contributing to the body of knowledge in their field. Research projects develop students' research skills, critical thinking, and ability to communicate findings effectively.

8. Online Learning:

- Online learning is incorporated to provide flexible and accessible education. SRU utilizes online platforms to deliver lectures, conduct discussions, and facilitate collaborative projects. Online learning allows students to access course materials, participate in virtual classrooms, and engage with peers and instructors remotely.

9. Continuous Assessment:

- Continuous assessment methods, such as quizzes, assignments, and presentations, are used to monitor students' progress and provide ongoing feedback. These assessments help identify areas for improvement and ensure that students are meeting learning objectives throughout the course.

10. Peer Learning:

- Peer learning encourages students to collaborate and learn from each other. Group projects, study groups, and peer review sessions provide opportunities for students to share knowledge, offer feedback, and support each other's learning journey.

At SRU, our commitment to employing diverse and effective teaching methods ensures that our students receive a well-rounded education that prepares them for success in their chosen fields. We continuously review and enhance our teaching practices to provide the best possible learning experience for our students.

Delivery Methods

At Springfield Research University (SRU), we utilize a variety of delivery methods to ensure that our educational programmes are accessible, engaging, and effective. Our delivery methods are designed to cater to the diverse needs of our students and to provide flexible learning opportunities. The following are the key delivery methods employed across all SRU programmes:

1. In-Person Delivery:

- **Classroom Lectures:** Traditional classroom lectures provide a structured and interactive environment where students can engage with instructors and peers. These sessions often include discussions, presentations, and multimedia resources to enhance learning.

- **Laboratory Sessions:** For programmes that require practical and experimental learning, laboratory sessions are conducted in specialized labs equipped with the necessary tools and equipment. These hands-on sessions allow students to apply theoretical knowledge in a controlled environment.

- **Clinical Placements:** Health and Medical Sciences programmes include clinical placements in hospitals, clinics, and healthcare facilities. These placements provide students with real-world experience under the supervision of qualified professionals.

2. Online Delivery:

- **Virtual Classrooms:** Online platforms are used to deliver lectures, conduct discussions, and facilitate collaborative projects. Virtual classrooms enable students to access course materials, participate in live sessions, and engage with peers and instructors from remote locations.

- **Recorded Lectures:** Recorded lectures are made available for students to access at their convenience. This flexible approach allows students to review and revisit course content as needed.

- **Online Assessments:** Online assessments, including quizzes, assignments, and exams, are conducted through secure online platforms. These assessments provide timely feedback and help monitor students' progress.

3. Blended Learning:

- **Hybrid Courses:** Blended learning combines in-person and online delivery methods to provide a flexible and comprehensive learning experience. Hybrid courses may involve alternating between classroom sessions and online activities.

- **Flipped Classroom:** In the flipped classroom model, students access instructional content online before class and use in-person sessions for interactive, application-based activities. This approach encourages active learning and deeper engagement with the material.

4. Independent Study:

- **Self-Paced Learning:** Self-paced learning allows students to progress through course materials at their own speed. This method is ideal for students who prefer to learn independently and manage their own schedules.
- **Research Projects:** Independent research projects provide students with the opportunity to explore topics of interest, develop research skills, and contribute to the body of knowledge in their field. Faculty advisors provide guidance and support throughout the research process.

5. Collaborative Learning:

- **Group Projects:** Group projects foster teamwork and collaboration among students. These projects often involve problem-solving, research, and presentations, allowing students to learn from each other and develop interpersonal skills.
- **Peer Review:** Peer review sessions encourage students to provide and receive constructive feedback on each other's work. This method promotes critical thinking, reflection, and improvement.

6. Experiential Learning:

- **Internships and Work Placements:** Internships and work placements provide students with practical experience in their chosen field. These opportunities allow students to apply their knowledge in real-world settings, develop professional skills, and build industry connections.
- **Field Trips and Excursions:** Field trips and excursions offer experiential learning opportunities outside the classroom. These activities provide students with firsthand exposure to relevant sites, industries, and practices.

7. Continuous Assessment:

- **Formative Assessments:** Formative assessments, such as quizzes, assignments, and in-class activities, provide ongoing feedback to students and help track their progress. These assessments are designed to support learning and identify areas for improvement.
- **Summative Assessments:** Summative assessments, including final exams, projects, and presentations, evaluate students' overall performance and mastery of course content.

At SRU, our diverse delivery methods ensure that students receive a well-rounded and flexible education that caters to their individual learning preferences. We are committed to continuously enhancing our delivery methods to provide the best possible learning experience for our students.

Curriculum

Robotics Engineering, ESQF Level 8 BE-ROB degree, typical course sequence

YEAR 1: SEMESTER 1

Code	Course	Lectures	Practicals	Credits
ROB100	Calculus I	120	0	12
ROB101	Computer Science I	100	20	12
ROB102	Physics (Mechanics/Newtonian) and Lab	100	20	12
ROB103	General Chemistry	100	20	12
ROB104	Design I	120	0	12
ROB105	Introduction to Electronics	100	20	12
TOTAL				72

YEAR 1: SEMESTER 2

Code	Course	Lectures	Practicals	Credits
ROB106	Physics I and Laboratory	100	20	12
ROB107	Foundations of Robotics Engineering	100	20	12
ROB108	Calculus II	100	20	12
ROB109	Computer Science II	100	20	12
ROB110	Mechanical Engineering Graphics	100	20	12
TOTAL				60

YEAR 2: SEMESTER 3

Code	Course	Lectures	Practicals	Credits
ROB211	Calculus III	120	0	12
ROB212	Communication for Professional Purposes	120	0	12
ROB213	Physics II and Laboratory	100	20	12
ROB214	Statics	100	20	12
ROB215	Object Oriented Design Concepts	100	20	12
TOTAL				60

YEAR 2: SEMESTER 4

Code	Course	Lectures	Practicals	Credits
ROB216	Mechanics of Materials	100	20	12
ROB217	Unified Robotics I	100	20	12
ROB218	Dynamics	120	0	12
ROB219	Circuits and Electronics	100	20	12
ROB220	Differential Equations	120	0	12
TOTAL				60

YEAR 3: SEMESTER 5

Code	Course	Lectures	Practicals	Credits
ROB321	Probability and Statistics	120	0	12
ROB322	Digital Electronics and Lab	60	60	12
ROB323	Unified Robotics II	60	60	12
ROB324	Measurement Systems	100	20	12
ROB325	Linear Algebra	120	0	12
TOTAL				60

YEAR 3: SEMESTER 6

Code	Course	Lectures	Practicals	Credits

ROB326	Introduction to Projects	**100**	**20**	**12**
ROB327	Microprocessors	**60**	**60**	**12**
ROB328	Unified Robotics III	**60**	**60**	**12**
ROB329	Discrete Mathematics	**120**	**0**	**12**
ROB330	System Modeling and Control	**120**	**0**	**12**
TOTAL				**60**

YEAR 4: SEMESTER 7

Code	Course	Lectures	Practicals	Credits
ROB431	Engineering Cost Analysis	**120**	**00**	**12**
ROB432	Capstone Project I	**0**	**120**	**12**
ROB433	Discrete Control	**120**	**0**	**12**
ROB434	Unified Robotics IV	**60**	**60**	**12**
ROB435	Data Structures	**120**	**0**	**12**
TOTAL				**60**

YEAR 4: SEMESTER 8

Code	Course	Lectures	Practicals	Credits
ROB436	Capstone Project II	**0**	**120**	**12**
ROB437	Embedded Systems	**120**	**0**	**12**
ROB438	Algorithms	**120**	**0**	**12**
ROB439	Accelerated Object-Oriented Design Concepts	**60**	**60**	**12**
ROB440	Leadership and Professional Development for Engineers	**120**	**0**	**12**
TOTAL				**60**

COURSE DESCRIPTIONS

COMPUTER SCIENCE I

Introduction to programming with C++. Binary, two's complement, decimal, hex, and octal representations. Variable types. Simple, iterative, and conditional statements. Procedure and functions with parameters by value and reference with or without a returning value. Arrays and vectors, multidimensional arrays, bubble and selection sorts, linear and binary search. Pointer and dynamic memory allocation, character and C-strings, file input/output (sequential). Classes, friends, array of objects, and operators' overloading. Inheritance, polymorphism, virtual function, and recursion.

FUNDAMENTALS OF ENGINEERING DESIGN PROJECTS

This course introduces the student to the engineering design and problem-solving process through engaging, interdisciplinary, team-based design projects, as well as individual assignments. Professional skills/attributes such as oral and written communication, innovation, tolerance for uncertainty/ambiguity, risk management, social awareness, and professional ethics will be investigated and practiced.

PHYSICS I AND LABORATORY

Kinematics and dynamics of particles, conservation of energy relativistic mechanics, statics and dynamics of rigid bodies. Mechanics of fluids, temperature, heat and thermal energy; laws of thermodynamics; wave motion and sound. Introductory laboratory experiments to complement Physics I.

FOUNDATIONS OF ROBOTICS ENGINEERING

This course introduces the student to the specific discipline of robotics engineering, the areas of study undertaken by robotics engineers, and the related design and problem-solving process used by robotics engineers through engaging, team-based design projects, individual assignments and other class activities. Professional skills/attributes such as oral and written communication, innovation, tolerance for uncertainty/ambiguity, risk management, social awareness, and professional ethics will be investigated and practiced.

CALCULUS I

Topics include, limits and continuity, differentiation of algebraic and transcendental functions, mean value theorem, applications of differentiation, anti-derivatives, indefinite integrals, inverse trigonometric functions, substitutions, definite integrals, the Fundamental Theorem of Calculus, applications of integration. Applications will be emphasized.

CALCULUS II

Hyperbolic functions, L'Hospital's rule, techniques of integration, application to arc length and surface area, polar coordinates, infinite series, Taylor Series.

CALCULUS III

Three-dimensional analytic geometry. Vectors, vector-valued functions, motions in space, functions of several variables, partial differentiation, multiple integration, integration of vector fields, Green's Theorem and Divergence Theorem.

COMPUTER SCIENCE II

Records, advanced file input/output (random access), dynamic memory allocation. Static and dynamic implementation of stacks, linked lists (ordered and unordered), queue (regular and priority), circular queues. Selection and insertion sort, binary search.

MECHANICAL ENGINEERING GRAPHICS

An integrated course in engineering graphics for all students in the College of Engineering. Introduction to graphics in design, fundamentals of orthogonal projection and experience in applying these principles to the solution of space problems. ACAD software.

TECHNICAL AND PROFESSIONAL COMMUNICATION

Training in a systematic method for producing effective technical communication, written reports, letters, and memos as well as oral presentations.

UNIFIED ROBOTICS I

This is the first course in a four course sequence combining both the theory and practice of robotics engineering. The focus of this course is the introduction of robotic system applications, components of a robotic system, basic programming for robotics, object-oriented programming and design, GUI and event driven programming, debugging tools and coding standards, state machines, Introduction to EmguCV, Ohm's law, Kirchhoff's voltage law, Kirchhoff's current law, single loop systems, analog sensors, microprocessors, electromechanical actuators, mechanical actuators, degrees of freedom, spatial rotation matrices, coordinate transformations, constraints, and rigid body forward kinematics. The laboratory sessions consist of hands-on exercises and team projects where students design, build, and program robots and related sub-systems.

PHYSICS II AND LABORATORY

Introductory study of electronics, direct currents, magnetism, electromagnetic waves. Light reflection, refraction, and polarization. Imaging by a mirror and lens.

DYNAMICS

Kinematics and kinetics of particles, rigid bodies, and systems of particles and rigid bodies will be analyzed by the classical methods; vibrations of single degree of freedom systems.

STATICS

A first course in engineering mechanics which covers the following topics: Vector Algebra, resultant of force systems; equilibrium of particles, rigid bodies using free-body diagrams; friction; centroids; moments of inertia.

LEADERSHIP & PROFESSIONAL DEVELOPMENT FOR ENGINEERS

Students will study and apply leadership, ethics, teamwork, and professional development relevant to engineering. The course will introduce frameworks for various leadership concepts and ethical approaches in personal, professional, and organizational settings. Students will develop personal leadership and ethical philosophy through self-reflection and self- and peer assessment of teamwork and problem-solving.

UNIFIED ROBOTICS I

This is the first course in a four-course sequence combining both the theory and practice of robotics engineering. The focus of this course is the introduction of robotic system applications, components of a robotic system, basic programming for robotics, object-oriented programming and design, GUI and event driven programming, debugging tools and coding standards, state machines, Introduction to EmguCV, Ohm's law, Kirchhoff's voltage law, Kirchhoff's current law, single loop systems, analog sensors, microprocessors, electromechanical actuators, mechanical actuators, degrees of freedom, spatial rotation matrices, coordinate transformations, constraints, and rigid body forward kinematics. The laboratory sessions consist of hands-on exercises and team projects where students design, build, and program robots and related sub-systems.

DYNAMICS

Kinematics and kinetics of particles, rigid bodies, and systems of particles and rigid bodies will be analyzed by the classical methods; vibrations of single degree of freedom systems.

CIRCUITS AND ELECTRONICS

Fundamental laws. Circuit parameters, elementary network theory. Forced and transient response, semi-conductor devices, electronic circuits, digital logic and counting circuits. The course includes hands-on experiments.

DIFFERENTIAL EQUATIONS

Topics include, but are not limited to, solving first and second-order differential equations and first-order linear systems of differential equations by various techniques such as separation of variables, integrating factors, substitution methods, variation of parameters, and Laplace Transforms. Emphasis will be placed on applications of differential equations arising from engineering applications and the natural sciences.

PROBABILITY & STATISTICS

Representation of data, probability, random variables, discrete and continuous distributions, sampling theory, central limit theorem, confidence intervals, tests of statistical hypotheses, regression analysis. Lecture 3 hrs.

DIGITAL ELECTRONICS & LAB

The course is an integrated lecture/laboratory course. Logic gates, design and minimization of combinational circuits, MSI and LSI circuits and applications, sequential circuit analysis and design.

MECHANICS OF MATERIALS

Stress as internal force intensity. Stress and deflections due to: axial torsional, and bending; loads and design of an efficient structure.

UNIFIED ROBOTICS II

This is the second course in a four-course sequence combining both the theory and practice of robotics engineering. The focus of this course is the introduction of energy methods, virtual displacement and the virtual work principle, degrees of freedom and generalized coordinates, the principle of D'Alembert, Hamilton's Principle, Lagrange's Equations, constrained systems and Lagrange multipliers, Raspberry PI (RPI) and Linux Bash programming, Python programming, network programming for robotics, image processing and computer vision for

RPI, and software Engineering for robotics. The laboratory sessions consist of hands-on exercises and team projects where students design, build, and program robots and related sub-systems.

MEASUREMENT SYSTEMS

This course introduces students to the use and design of measurement systems for engineering practice. Topics include components of measurement systems, calibration, data acquisition, commonly used sensors, time and frequency domain signal analysis, statistical analysis of data, and data processing and validation. The course concludes with group projects.

LINEAR ALGEBRA

Systems of linear equations, matrices, determinants, eigenvalues, eigenvectors, Finite-dimensional vector spaces, linear transformations and their matrices, Gram-Schmidt orthogonalization, inner product spaces.

INTRODUCTION TO PROJECTS

Introduction to the design process, matching engineering specifications to customer requirements, prototyping, product testing and evaluation, project planning and management. Students will select senior projects, form project teams and submit a project pre-proposal. Orientation to fabrication facilities.

MICROPROCESSORS

Architecture, timing, instruction set, memory and input/output techniques for various microprocessors, design of a microcomputer system.

UNIFIED ROBOTICS III

This is the third course in a four course sequence combining both the theory and practice of robotics engineering. The focus of this course is the introduction of robot inverse kinematics, the Denavit-Hartenberg representation, the inverse kinematics solution, trajectory planning for robots, power sources, programmable logic controllers (PLCs), ladder logic, and digital sensors. The laboratory sessions consist of hands-on exercises and team projects where students design, build, and program robots and related sub-systems.

DISCRETE MATHEMATICS

Number Theory, review of induction and recursion, advanced counting, equivalence, partial ordering, graphs, trees.

SYSTEM MODELING AND CONTROL

This course will cover the fundamentals of mathematically modeling engineering systems along with the corresponding techniques used to control these systems. Topics include mathematical modeling of lumped mechanical, electrical, and electromechanical systems; system modeling; linear time invariant system theory; Laplace transform, transfer functions and block diagrams; introduction to state-space formulation; time response and frequency response analysis; stability analysis; introduction to linear feedback control and system design using root locus and frequency domain techniques.

ENGINEERING COST ANALYSIS

Basic development of the cost consequences of engineering decision-making. Interest calculations, cash flow equivalences, annual cash flows, rates of return, incremental analysis and other analytical approaches. Depreciation, income taxes and replacement analysis.

CAPSTONE PROJECTS I

Topics of mechanical design validation/optimization (structural integrity analysis, robot dynamic validation, actuator testing), sensor systems testing and fusion, control hardware architecture, and experimental system identification are addressed in the context of the specific projects. Students initiate the project's design process, produce a formal engineering proposal, and give oral presentations. Project teams work closely with their faculty advisor.

DISCRETE CONTROL

This course presents the theory and techniques of digital control of engineering systems. Topics include the Z- transform, properties of the z-transform, model description, system discretization, pole placement, feedback using multiple discrete inputs (state space method), discrete observers-controller systems.

UNIFIED ROBOTICS IV

This is the final course in a four-course sequence combining mechanical engineering, electrical & computer engineering and computer science to develop both the theory and practice of robotics engineering. The focus of this course is navigation, position estimation, simultaneous localization and mapping (SLAM) and communications. Control systems as applied to navigation will be presented. Human robot interaction (HRI) and remote sensing for mobile robots will be introduced. Advanced topics such as cooperative robotics, robot vision, and sensor fusion can also be introduced. The laboratory sessions will be directed towards the solution of an open-ended problem over the course of the entire term.

DATA STRUCTURES

Analysis of algorithms, Big Oh notation, asymptotic behavior. Advanced sorting (heapsort, quicksort), external sorting. Binary, multiway, and AVL trees.

CAPSTONE PROJECTS II

Topics of control strategy selection, software architecture, version control, and control strategy/software troubleshooting and optimization are addressed in the context of the specific projects. Students complete the project's design process, build and validate a prototype, produce a formal final report, and give oral presentations. Project teams work closely with their faculty advisor to meet their project objectives.

EMBEDDED SYSTEMS

Utilization of microcontrollers in design of instruments and embedded controllers. Description of on-chip resources, programming framework, parallel I/O, main timer and real-time interrupt, pulse accumulator, A/D converter and serial communication subsystems. Interfacing techniques.

INTRODUCTION TO ELECTRICAL AND COMPUTER ENGINEERING

This course provides an integrated introduction to electrical engineering and computer science, taught using substantial laboratory experiments with mobile robots. The primary goal is for you to learn to appreciate and use the fundamental design principles of modularity and abstraction in a variety of contexts from electrical engineering and computer science. The second goal is to show you that making mathematical models of real systems can help in the design and analysis of those systems. Finally, we have the more typical goals of teaching exciting and important basic material from electrical engineering and computer science, including modern software engineering, linear systems analysis, electronic circuits, and decision-making.

SYSTEMS PROGRAMMING CONCEPTS

Topics include: program development tools, basic testing, timing, profiling and benchmarking, characteristics of physical devices, memory management, device drivers, pseudo-devices, file structures, file I/O (both buffered and unbuffered), processes, shells, inter-process communications, signals, exceptions, pipes, sockets, shared memory and file and record locking. All topics are viewed from a UNIX system programming perspective. The general objective of this course is to introduce students basic concepts, techniques and skills for systems programming. Specific objectives include: Understand basic concepts in systems programming, understand basic concepts in UNIX file systems and process control, understand UNIX system calls and develop skills to write programs using system services.

OBJECT-ORIENTED DESIGN CONCEPTS

This course introduces students to an object-oriented model of programming. Building from the design methodology covered in Computer Science, this course shows how programs can be decomposed into classes and objects. By emphasizing design, this course shows how to implement small defect-free programs and evaluate design decisions to select an optimal design under specific assumptions. Topics include inheritance, exceptions, interface, design by contract, basic design patterns, and reuse. Students will be expected to design, implement, and debug object-oriented programs composed of multiple classes and over a variety of data structures.

ACCELERATED OBJECT-ORIENTED DESIGN CONCEPTS

This course covers the data structures and general program-design material from object-oriented design concepts but assumes that students have significant prior experience in object-oriented programming. The course covers object-oriented design principles and data structures more deeply and at a faster pace. Students will be expected to design, implement, test, debug, and critique programs both for correctness and adherence to good object-oriented design principles. The course is designed to strengthen both the design skills and algorithmic thinking of students who already have a foundation in object-oriented programming.

ALGORITHMS

Building on a fundamental knowledge of data structures, data abstraction techniques, and mathematical tools, a number of examples of algorithm design and analysis, worst case and average case, will be developed. Topics include greedy algorithms, divide-and-conquer, dynamic programming, heuristics, and probabilistic algorithms. Problems will be drawn from areas such as sorting, graph theory, and string processing. The influence of the computational model on algorithm design will be discussed. Students will be expected to perform analysis on a variety of algorithms.

COURSE OUTLINES

Course Title: Calculus I

Course Description:

Calculus I provides foundational knowledge in differential and integral calculus, essential for understanding dynamic systems, optimization, and control in robotics.

Learning Objectives:

- Master differentiation techniques for functions of one variable.
- Understand integration methods and their applications.
- Apply calculus concepts to robotic modeling and analysis.

Topics Covered:

- **Differentiation**:
  - Limits and continuity.
  - Derivatives of algebraic, trigonometric, exponential, and logarithmic functions.
  - Chain rule and implicit differentiation.
- **Integration**:
  - Definite and indefinite integrals.
  - Techniques of integration (substitution, integration by parts).
  - Applications in robotics (velocity, acceleration, optimization).
- **Series and Sequences**:
  - Convergence and divergence of series.
  - Taylor and Maclaurin series.
- **Multivariable Calculus (Brief Introduction)**:
  - Partial derivatives.
  - Gradient, divergence, and curl.

Assessment and Grading:

- Regular problem sets and quizzes.
- Midterm exams covering differentiation and integration.
- Final exam assessing overall understanding.

Recommended Resources:

- James, G. (2001) “Modern Engineering Mathematics.”
- Kreyszig, E. (1999) “Advanced Engineering Mathematics.”
- Stephenson, G. (1973) “Mathematical Methods for Science Students.”
- Fischer-Cripps, A.C. (2005) “The Mathematics Companion.”

Course Title: Computer Science I

Course Description:

This introductory course provides fundamental knowledge in computer science and programming. It is designed for students with no prior programming experience.

Learning Objectives:

- Understand the basics of computer science.
- Learn programming concepts using Java.
- Explore topics such as object-oriented programming, control statements, and file input/output.

Topics Covered:

- **Introduction**:
  - History of programming languages.
  - Hardware vs. software.
  - Overview of Java.
- **Object-Oriented Programming**:
  - Objects, classes, inheritance, and polymorphism.
  - Java’s primitive data types.
  - Control statements and exception handling.
- **File Input/Output**:
  - Reading and writing files.
  - Handling exceptions during file operations.

Assessment and Grading:

- Regular assignments and quizzes.
- Midterm and final exams.

Recommended Resources:

- No specific textbook required, but consider supplementary materials on Java programming.

Course Title: Physics (Mechanics/Newtonian) and Lab

Course Description:

Physics (Mechanics/Newtonian) introduces fundamental principles of classical mechanics, emphasizing the behavior of physical systems in motion. Students will explore topics essential for understanding robotic dynamics and control.

Learning Objectives:

- Develop a solid foundation in Newtonian mechanics.
- Apply physics concepts to robotic modeling and analysis.
- Enhance problem-solving skills for robotics applications.

Topics Covered:

- **Kinematics**:
  - Position, velocity, and acceleration.
  - Projectile motion and circular motion.
- **Dynamics**:
  - Newton’s laws of motion.
  - Work, energy, and momentum.
  - Rotational dynamics.
- **Applications in Robotics**:
  - Robot arm kinematics.
  - Trajectory planning and control.
  - Stability analysis.

Laboratory Topics:

- Measurement techniques (length, time, mass).
- Free fall experiments.
- Conservation of momentum.
- Rotational motion investigations.
- Oscillations and harmonic motion.

Assessment and Grading:

- Regular problem sets and quizzes.
- Lab reports and data analysis.
- Practical assessments during lab sessions.
- Participation and engagement.

Course Title: General Chemistry

Course Description:

General Chemistry covers the foundational principles of chemistry. Students will explore topics related to matter, atomic structure, chemical reactions, and more.

Learning Objectives:

- Understand the nature of matter and its properties.
- Learn about atomic theory and the periodic table.
- Explore chemical reactions and stoichiometry.

Topics Covered:

- **Foundations of Chemistry**:
  - Scientific measurement and units.
  - States of matter (solid, liquid, gas).
- **Atomic Structure and Bonding**:
  - Subatomic particles (protons, neutrons, electrons).
  - Chemical bonding (ionic, covalent).
- **Chemical Reactions**:
  - Types of reactions (acid-base, redox).
  - Balancing chemical equations.
- **Thermochemistry and Solutions**:
  - Heat transfer, enthalpy, and entropy.
  - Solution properties and concentration.

Assessment and Grading:

- Regular assignments, quizzes, and exams.
- Lab experiments (if applicable).

Recommended Resources:

- Textbooks specific to general chemistry.
- Lecture notes and supplementary materials.

Course Title: Design I

Course Description:

Design I introduces students to fundamental principles of design thinking, creativity, and problem-solving. Through hands-on projects and collaborative activities, students will develop skills necessary for designing innovative solutions.

Learning Objectives:

1. Understand the design process from ideation to implementation.
2. Apply design thinking techniques to real-world challenges.
3. Develop creativity and critical thinking skills.
4. Collaborate effectively in interdisciplinary teams.

Topics Covered:

1. **Introduction to Design Thinking**:
    - Overview of design principles.
    - User-centered design.
    - Brainstorming and ideation.
2. **Problem Definition and Research**:
    - Identifying design problems.
    - Conducting user research.
    - Defining design criteria.
3. **Concept Development and Prototyping**:
    - Sketching and visualization.
    - Creating design concepts.
    - Rapid prototyping techniques.
4. **Materials and Manufacturing Processes**:
    - Understanding materials (metals, plastics, composites).
    - Manufacturing methods (3D printing, CNC machining).
5. **Design Evaluation and Iteration**:
    - Usability testing.
    - Feedback and refinement.
    - Iterative design cycles.

Assessment and Grading:

- Design projects (individual and group).
- Design journals documenting the design process.
- Peer evaluations.
- Final design presentation.

Recommended Resources:

- No specific textbook required, but consider design literature and case studies.

Course Title: Introduction to Electronics

Course Description:

Introduction to Electronics provides foundational knowledge in electronic principles and circuits. Students will explore topics related to electronic components, circuit analysis, and practical applications.

Learning Objectives:

1. Understand the basics of electronic components (resistors, capacitors, diodes, etc.).
2. Analyze simple electronic circuits using Ohm's law and Kirchhoff's laws.
3. Explore DC circuit analysis and magnetism.
4. Introduce AC theory and devices.

Topics Covered:

1. **Fundamentals of Electronics**:
    - Introduction to electronic components.
    - Voltage, current, and resistance.

    - Circuit models and analysis.
2. **DC Circuit Analysis**:
    - Ohm’s law and power calculations.
    - Series and parallel circuits.
    - Voltage dividers and current sources.
3. **Magnetism and Electromagnetic Devices**:
    - Magnetic fields and forces.
    - Inductors and transformers.
    - Applications in electronic systems.
4. **AC Theory and Devices**:
    - Alternating current (AC) circuits.
    - Capacitors and reactance.
    - Introduction to AC devices (e.g., oscillators, filters).

Assessment and Grading:

- Regular assignments and quizzes.
- Practical circuit design and analysis.

Recommended Resources:

- No specific textbook required, but consider supplementary materials on electronics.

Course Title: Introduction to Electrical and Computer Engineering

Course Description:

This course provides an essential foundation in electrical engineering and computer science concepts relevant to robotics engineering. Students will explore fundamental principles, gain hands-on experience, and develop problem-solving skills applicable to robotics systems.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the basics of electrical circuits, digital logic, and computer systems.
2. Apply mathematical modeling techniques to analyze electrical and computer engineering problems.
3. Describe the role of software engineering in robotics applications.
4. Demonstrate knowledge of control systems and their relevance to robotics.
5. Evaluate decision-making algorithms used in autonomous systems.

Topics Covered:

1. **Introduction to Electrical Engineering and Computer Science**
    - Overview of electrical engineering and computer science disciplines
    - Importance of modularity and abstraction in design
    - Mathematical modeling for system analysis
2. **Software Engineering Fundamentals**
    - Basics of software development
    - Algorithms and data structures

    - Object-oriented programming
    - Software testing and debugging
3. **Linear Systems Analysis**
    - Signals and systems
    - Laplace transforms
    - Transfer functions
    - Control systems
4. **Electronic Circuits and Devices**
    - Semiconductor physics
    - Diodes, transistors, and amplifiers
    - Digital logic circuits
    - Analog and digital electronics
5. **Robotics and Control Systems**
    - Robot kinematics and dynamics
    - Sensors and actuators
    - Feedback control
    - Path planning and motion control

Assessment:

- Regular quizzes and assignments
- Midterm and final exams
- Hands-on projects (e.g., building simple circuits, programming microcontrollers)
- Participation in class discussions and labs

Recommended Reading:

1. "Introduction to Robotics: Mechanics and Control" by John J. Craig
2. "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy
3. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein

Course Title: Foundations of Robotics Engineering

Course Description:

This course provides a challenging introduction to basic computational concepts used broadly in robotics. Students will explore key topics related to simulation, kinematics, control, optimization, and probabilistic inference. Emphasis will be placed on the mathematical foundations of each area, with practical applications and programming exercises.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the fundamental principles of robotics and their mathematical underpinnings.
2. Apply kinematics and dynamics concepts to analyze robot motion.
3. Design and control robotic systems using optimization techniques.
4. Implement probabilistic inference methods for uncertainty modeling.

Topics Covered:

1. **Introduction to Robotics**
   - History of robotics
   - Definitions and classifications
   - Typical robot applications
   - Representation of robot function
2. **Simulation and Modeling**
   - Simulation techniques for robotics
   - Kinematic modeling
   - Dynamics modeling
3. **Control Systems for Robots**
   - Feedback control
   - Trajectory generation
   - Motion planning
4. **Optimization in Robotics**
   - Optimization algorithms
   - Parameter tuning
   - Path optimization
5. **Probabilistic Inference**
   - Bayesian methods
   - Markov models
   - Sensor fusion

Assessment:

- Regular assignments and problem sets
- Midterm and final exams
- Hands-on projects (e.g., implementing control algorithms, simulating robot behavior)
- Participation in class discussions and labs

Recommended Reading:

1. "Robot Modeling and Control" by Mark W. Spong, Seth Hutchinson, and M. Vidyasagar
2. "Probabilistic Robotics" by Sebastian Thrun, Wolfram Burgard, and Dieter Fox

Course Title: Calculus II

Course Description:

Calculus II builds upon the concepts introduced in Calculus I. This course delves deeper into integration techniques, differential equations, applications of integrals, parametric equations, polar coordinates, and series.

Learning Outcomes:

By the end of this course, students should be able to:

1. Apply integration techniques to solve complex problems.
2. Understand and analyze differential equations.

3. Apply integrals to real-world scenarios.
4. Work with parametric equations and polar coordinates.
5. Explore infinite series and their convergence.

Topics Covered:

1. **Integrals Review**
   - Accumulations of change
   - Riemann sums
   - Definite integrals
   - Fundamental theorem of calculus
2. **Integration Techniques**
   - u-substitution
   - Trigonometric identities
   - Integration by parts
   - Partial fractions
   - Improper integrals
3. **Differential Equations**
   - Verifying solutions
   - Slope fields
   - Euler's method
   - Separation of variables
4. **Applications of Integrals**
   - Average value of a function
   - Area between curves
   - Volume of solids
   - Arc length
5. **Parametric Equations and Polar Coordinates**
   - Second derivatives of parametric equations
   - Arc length for parametric curves
   - Vector-valued functions
   - Area in polar regions
6. **Series**
   - Convergent and divergent series
   - Geometric series
   - Taylor and Maclaurin polynomials
   - Error bounds

Assessment:

- Regular assignments and quizzes
- Midterm and final exams
- Projects involving real-world applications
- Participation in class discussions and labs

Recommended Reading:

1. "Calculus" by James Stewart
2. "A First Course in Mathematical Analysis" by David Alexander Brannan

Course Title: Computer Science II

Course Description:

Computer Science II builds upon the foundational concepts introduced in Computer Science I. This course focuses on advanced topics related to algorithms, data structures, and software development, with specific applications relevant to robotics engineering.

Learning Outcomes:

By the end of this course, students should be able to:

1. Analyze and design efficient algorithms for solving complex problems.
2. Implement and manipulate data structures commonly used in robotics applications.
3. Apply software engineering principles to develop robust and scalable programs.
4. Understand the role of computer science in robotics research and development.

Topics Covered:

1. **Algorithm Design and Analysis**
   - Asymptotic notation (Big O, Theta, Omega)
   - Sorting algorithms (quicksort, mergesort, heapsort)
   - Graph algorithms (Dijkstra's, Bellman-Ford, minimum spanning trees)
2. **Data Structures**
   - Linked lists, stacks, and queues
   - Trees (binary trees, AVL trees, B-trees)
   - Hash tables and hash functions
   - Graph representations (adjacency lists, adjacency matrices)
3. **Object-Oriented Programming (OOP)**
   - Classes and objects
   - Inheritance and polymorphism
   - Design patterns (singleton, factory, observer)
4. **Software Development Practices**
   - Version control (Git)
   - Testing (unit testing, integration testing)
   - Debugging techniques
   - Code documentation and best practices
5. **Robotics-Specific Applications**
   - Robot control algorithms (PID controllers, path planning)
   - Sensor integration (lidar, cameras, IMUs)
   - Real-time programming considerations
   - Human-robot interaction (HRI) principles

Assessment:

- Regular programming assignments
- Midterm and final exams
- Group projects (implementing algorithms, building robot simulations)
- Participation in coding labs and discussions

Recommended Reading:

1. “Introduction to Algorithms” by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
2. “Clean Code: A Handbook of Agile Software Craftsmanship” by Robert C. Martin

Course Title: Mechanical Engineering Graphics

Course Description:

Mechanical Engineering Graphics introduces fundamental concepts related to engineering sketching, drawing, and graphical representation. Students will learn techniques for creating accurate and precise engineering drawings using both freehand sketching and drawing instruments.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the basics of engineering sketching and drawing.
2. Create orthographic projections, isometric views, and sectional drawings.
3. Apply geometric dimensioning and tolerancing (GD&T) principles.
4. Interpret engineering drawings and assembly diagrams.

Topics Covered:

1. **Geometric Construction and Sketching**
    - Freehand sketching techniques
    - Geometric shapes and constructions
    - Visualization of three-dimensional objects
2. **Orthographic Projection**
    - Multiview drawings (top, front, side views)
    - Dimensioning conventions
    - Auxiliary views
3. **Isometric Drawing**
    - Isometric projection principles
    - Representation of three-dimensional objects
    - True lengths and angles in isometric views
4. **Sectional Views**
    - Cutting planes and section lines
    - Half-sections and full sections
    - Revolved sections
5. **Geometric Dimensioning and Tolerancing (GD&T)**
    - Symbols and terminology
    - Tolerance zones
    - Position, concentricity, and profile controls

Assessment:

- Regular sketching assignments

- Quizzes on drawing techniques
- Interpretation of complex engineering drawings
- Hands-on projects involving assembly drawings

Recommended Reading:

1. “Engineering Drawing and Design” by David A. Madsen and David P. Madsen
2. “Geometric Dimensioning and Tolerancing” by David A. Madsen

Course Title: Calculus III

Course Description:

Calculus III builds upon the concepts introduced in Calculus I and II. This course extends calculus to functions of several variables, covering topics such as multivariable differentiation, integration, and vector calculus. Students will explore three-dimensional space, vector fields, and applications relevant to engineering and physics.

Learning Outcomes:

By the end of this course, students should be able to:

1. Apply multivariable calculus techniques to solve problems in three dimensions.
2. Understand vector functions, partial derivatives, and multiple integrals.
3. Analyze surfaces, curves, and motion in space.
4. Apply calculus concepts to physical and engineering contexts.

Topics Covered:

1. **3-Dimensional Space**
    - Standard 3D coordinate system
    - Alternative coordinate systems (cylindrical, spherical)
    - Equations of lines and planes in three dimensions
2. **Vector Functions**
    - Tangent and normal vectors
    - Arc length and curvature
    - Velocity and acceleration
3. **Multivariable Differentiation**
    - Partial derivatives
    - Gradient vector
    - Chain rule for multivariable functions
4. **Multiple Integrals**
    - Double integrals (rectangular and polar coordinates)
    - Triple integrals
    - Applications (volume, mass, center of mass)
5. **Vector Calculus**
    - Line integrals
    - Green’s theorem
    - Divergence and curl

Assessment:

- Regular assignments and quizzes
- Midterm and final exams
- Projects involving real-world applications
- Participation in class discussions and labs

Recommended Reading:

1. “Calculus: Early Transcendentals” by James Stewart
2. “Vector Calculus” by Susan Jane Colley

Course Title: Technical and Professional Communication

Course Description:

Technical and Professional Communication equips students with essential skills for effective communication in engineering and technology fields. This course focuses on written and oral communication, emphasizing clarity, precision, and professionalism.

Learning Outcomes:

By the end of this course, students should be able to:

1. **Literature Search and Research Skills:**
   - Conduct effective literature searches
   - Evaluate and cite relevant sources
2. **Technical Writing:**
   - Write clear and concise technical documents (reports, manuals, proposals)
   - Use appropriate language for different audiences
3. **Oral Presentation Skills:**
   - Plan and deliver effective presentations
   - Use visual aids and engage the audience

Topics Covered:

1. **Introduction to Technical Communication**
   - Importance of communication in engineering
   - Ethical considerations
2. **Research and Literature Review**
   - Effective search strategies
   - Evaluating sources
3. **Technical Writing Techniques**
   - Structuring technical documents
   - Writing for clarity and conciseness
   - Case studies and real-world examples
4. **Oral Communication**
   - Planning and organizing presentations

    - Effective slide design
    - Overcoming nervousness
5. **Professionalism and Collaboration**
    - Email etiquette
    - Collaborative writing
    - Team communication

Assessment:

- Written assignments (technical reports, documentation)
- Oral presentations
- Participation in class discussions and peer reviews

Recommended Reading:

1. "Technical Communication" by Mike Markel
2. "Writing in the Sciences" by Ann M. Penrose and Steven B. Katz

Course Title: Physics II and Lab

Course Description:

Physics II is the second installment of introductory calculus-based physics. It focuses on the study of wave motion, electric fields, Coulomb's Law, Gauss' law, capacitors, resistors, DC electric circuits, magnetic fields, induction, radiation, and the basic properties of electromagnetic waves. These topics are reinforced with lab sessions.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand wave phenomena and electromagnetic principles.
2. Analyze electric and magnetic fields.
3. Apply concepts to practical engineering scenarios.

Topics Covered:

1. **Wave Motion and Electromagnetic Waves**
    - Wave properties (frequency, wavelength, amplitude)
    - Electromagnetic spectrum
    - Reflection, refraction, and interference
2. **Electric Fields and Coulomb's Law**
    - Electric field lines
    - Electric potential
    - Capacitors and energy storage
3. **Magnetic Fields and Induction**
    - Magnetic field lines
    - Faraday's law of electromagnetic induction
    - Lenz's law
4. **DC Electric Circuits**
    - Ohm's law
    - Kirchhoff's rules
    - RC and RL circuits

Lab Sessions:

- Hands-on experiments related to electric circuits, magnetic fields, and electromagnetic waves.

Assessment:

- Regular lab reports
- Quizzes and exams
- Participation in lab sessions

Recommended Reading:

1. “University Physics with Modern Physics” by Hugh D. Young and Roger A. Freedman
2. “Introduction to Electrodynamics” by David J. Griffiths

Course Title: Statics

Course Description:

Statics is the study of forces and moments acting on stationary objects. In this course, students will learn principles related to equilibrium, free body diagrams, and the analysis of structures. Topics include vector analysis, moments, and the application of statics to various engineering systems.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the fundamental concepts of static equilibrium.
2. Analyze forces and moments in two and three dimensions.
3. Apply statics principles to solve engineering problems.

Topics Covered:

1. **Introduction to Statics**
   - Overview of static equilibrium
   - Units and problem-solving techniques
2. **Vector Analysis**
   - Vector addition and subtraction
   - Resultant forces
   - Moment vectors
3. **Forces and Moments**
   - Free body diagrams (FBD)
   - Equilibrium conditions
   - Applications to frames, trusses, and machines
4. **Structural Analysis**
   - Analysis of beams and cables
   - Frictional forces
   - Systems with pulleys

Assessment:

- Regular problem-solving assignments
- Quizzes and exams
- Application of statics principles to real-world scenarios

Recommended Reading:

1. “Engineering Mechanics: Statics” by R.C. Hibbeler
2. “Statics and Strength of Materials” by Harold I. Morrow and Robert P. Kokernak

Course Title: Object-Oriented Design Concepts

Course Description:

Object-Oriented Design (OOD) is essential for building robust, maintainable, and efficient software systems. In this course, students will learn principles related to designing software using object-oriented techniques. The focus will be on creating modular, reusable, and well-organized code, with specific applications relevant to robotics engineering.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the fundamental concepts of object-oriented programming.
2. Apply design patterns and principles to solve engineering problems.
3. Develop software components that adhere to OOD best practices.

Topics Covered:

1. **Introduction to Object-Oriented Programming (OOP)**
   - Basics of classes and objects
   - Encapsulation, inheritance, and polymorphism
   - SOLID principles
2. **Design Patterns**
   - Creational patterns (e.g., Singleton, Factory)
   - Structural patterns (e.g., Adapter, Decorator)
   - Behavioral patterns (e.g., Observer, Strategy)
3. **UML (Unified Modeling Language)**
   - Class diagrams
   - Sequence diagrams
   - Use case diagrams
4. **Software Architecture and Modularity**
   - Layered architecture
   - Component-based design
   - Dependency injection
5. **Robotics-Specific Applications**
   - Designing robot control software
   - Sensor integration using OOD
   - Real-time considerations

Assessment:

- Coding assignments (implementing OOD principles)
- Design documentation (UML diagrams, design patterns)
- Participation in class discussions and code reviews

Recommended Reading:

1. “Head First Design Patterns” by Eric Freeman, Elisabeth Robson, Bert Bates, and Kathy Sierra
2. “Clean Code: A Handbook of Agile Software Craftsmanship” by Robert C. Martin

Course Title: Mechanics of Materials

Course Description:

Mechanics of Materials explores the behavior of materials under various loads and stresses. This course provides foundational knowledge essential for designing and analyzing structures, components, and mechanical systems. Students will learn about stress, strain, deformation, and failure mechanisms.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the fundamental concepts of stress and strain.
2. Analyze mechanical components subjected to axial, torsional, and bending loads.
3. Apply material properties to solve engineering problems.

Topics Covered:

1. **Introduction to Mechanics of Materials**
   - Stress and strain definitions
   - Hooke’s law
   - Material properties (elastic modulus, Poisson’s ratio)
2. **Axial Loading and Deformation**
   - Normal stress and strain
   - Thermal stress
   - Stress concentration factors
3. **Torsion and Shear Stress**
   - Torsional deformation
   - Shear stress distribution
   - Power transmission elements (shafts, couplings)
4. **Bending and Flexural Deformation**
   - Bending stress and strain
   - Moment-curvature relationship
   - Deflection calculations
5. **Combined Loading and Mohr’s Circle**
   - Principal stresses
   - Maximum shear stress

- Mohr's circle construction

6. **Failure Theories and Design Considerations**
    - Yield criteria (von Mises, Tresca)
    - Factor of safety
    - Fatigue and fracture mechanics

Assessment:

- Homework assignments and problem-solving exercises
- Quizzes and exams
- Laboratory experiments (testing material properties)

Recommended Reading:

1. "Mechanics of Materials" by Ferdinand P. Beer, E. Russell Johnston Jr., John T. DeWolf
2. "Strength of Materials" by R. Subramanian

Course Title: Unified Robotics I

Course Description:

Unified Robotics I is a foundational course that introduces students to essential concepts in robotics engineering. It provides a comprehensive understanding of various topics related to robotics, laying the groundwork for more advanced courses. Students will explore both theoretical principles and practical applications.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand fundamental robotics concepts, including kinematics, stress and strain, and electrical circuits.
2. Apply knowledge of pneumatics, sensors, and signal conditioning.
3. Gain proficiency in embedded system programming using the C language.

Topics Covered:

1. **Kinematics and Mechanisms**
    - Planar and spatial kinematics
    - Forward and inverse kinematics
    - Robot arm configurations
2. **Stress and Strain Analysis**
    - Basics of material behavior
    - Tensile and compressive stress
    - Deformation and elasticity
3. **Electrical Circuits for Robotics**
    - Circuit components (resistors, capacitors, inductors)
    - Operational amplifiers
    - Motor drive circuits

4. **Sensors and Signal Conditioning**
    - Types of sensors (proximity, ultrasonic, vision)
    - Analog-to-digital conversion
    - Noise reduction techniques
5. **Embedded System Programming**
    - Introduction to C programming
    - Writing code for microcontrollers
    - Interfacing with sensors and actuators

Assessment:

- Homework assignments and programming exercises
- Quizzes and exams
- Hands-on lab sessions (programming microcontrollers, sensor integration)

Recommended Reading:

1. "Robotics: Modelling, Planning and Control" by Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo
2. "Introduction to Robotics: Mechanics and Control" by John J. Craig

Course Title: Dynamics

Course Description:

Dynamics explores the motion of objects and systems under the influence of forces. In this course, students will delve into the principles of kinematics, kinetics, and energy methods. The focus will be on analyzing mechanical systems, including robotic mechanisms, using mathematical tools.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the fundamental concepts of motion and forces.
2. Analyze rigid body dynamics and particle motion.
3. Apply energy methods to solve engineering problems.

Topics Covered:

1. **Kinematics**
    - Particle motion (position, velocity, acceleration)
    - Rigid body motion (translation, rotation)
    - Relative motion analysis
2. **Kinetics of Particles**
    - Newton's laws of motion
    - Work and energy
    - Impulse and momentum
3. **Rigid Body Dynamics**
    - Angular momentum

    - Moment of inertia
    - Equations of motion
4. **Energy Methods**
    - Work-energy principle
    - Conservation of mechanical energy
    - Virtual work

Assessment:

- Homework assignments and problem-solving exercises
- Quizzes and exams
- Application of dynamics principles to robotic systems

Recommended Reading:

1. “Engineering Mechanics: Dynamics” by R.C. Hibbeler
2. “Introduction to Robotics: Mechanics and Control” by John J. Craig

Course Title: Circuits and Electronics

Course Description:

Circuits and Electronics is a fundamental course that covers the principles of electrical circuits, electronic components, and their applications. In this course, students will learn about circuit analysis, semiconductor devices, and practical design techniques relevant to robotics engineering.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the behavior of electrical circuits.
2. Analyze and design basic electronic circuits.
3. Apply knowledge of electronic components to real-world problems.

Topics Covered:

1. **Basic Circuit Analysis**
    - Ohm’s law
    - Kirchhoff’s laws
    - Node and mesh analysis
2. **Semiconductor Devices**
    - Diodes and transistors
    - Operational amplifiers (op-amps)
    - Digital logic gates
3. **Electronic Components**
    - Resistors, capacitors, and inductors
    - Filters and signal conditioning
    - Power supplies
4. **Analog and Digital Electronics**
    - Amplifiers and oscillators
    - Analog-to-digital conversion

- Microcontrollers and embedded systems

Assessment:

- Homework assignments and problem-solving exercises
- Quizzes and exams
- Laboratory sessions (building and testing electronic circuits)

Recommended Reading:

1. “Microelectronic Circuits” by Adel S. Sedra and Kenneth C. Smith
2. “Electronic Devices and Circuit Theory” by Robert L. Boylestad

Course Title: Differential Equations

Course Description:

Differential Equations is a foundational mathematics course that explores the behavior of functions and their derivatives. In this course, students will learn techniques for solving differential equations and their applications in various fields, including science, engineering, and robotics.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the fundamental concepts of differential equations.
2. Solve first-order and higher-order differential equations.
3. Apply differential equations to model real-world phenomena.

Topics Covered:

1. **Basic Concepts**
    - Definitions of differential equations
    - Order and degree of differential equations
    - Initial value problems and boundary value problems
2. **First-Order Differential Equations**
    - Separable equations
    - Linear equations
    - Exact equations
    - Integrating factors
3. **Higher-Order Linear Differential Equations**
    - Homogeneous and nonhomogeneous equations
    - Constant coefficient equations
    - Method of undetermined coefficients
    - Variation of parameters
4. **Systems of Differential Equations**
    - Matrix notation
    - Eigenvalues and eigenvectors
    - Stability analysis
    - Applications in robotics dynamics and control
5. **Special Topics**
    - Laplace transforms

- Fourier series
- Partial differential equations (brief introduction)

Assessment:

- Homework assignments and problem-solving exercises
- Quizzes and exams
- Application of differential equations to modeling robotic systems

Recommended Reading:

1. “Elementary Differential Equations and Boundary Value Problems” by William E. Boyce and Richard C. DiPrima
2. “Differential Equations with Applications and Historical Notes” by George F. Simmons

Course Title: Probability and Statistics

Course Description:

Probability and Statistics is a foundational mathematics course that explores uncertainty, randomness, and data analysis. In this course, students will learn techniques for modeling uncertainty, making informed decisions, and drawing meaningful conclusions from data. The focus will be on both theoretical concepts and practical applications.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the fundamental concepts of probability theory.
2. Apply statistical methods to analyze data and draw valid inferences.
3. Use software tools (such as R) for statistical computations.

Topics Covered:

1. **Introduction to Probability Theory**
    - Basic probability models
    - Combinatorics and counting principles
    - Conditional probability and Bayes’ theorem
2. **Random Variables and Probability Distributions**
    - Discrete and continuous random variables
    - Probability mass functions and probability density functions
    - Expectation and variance
3. **Statistical Inference**
    - Point estimation and confidence intervals
    - Hypothesis testing
    - Maximum likelihood estimation
4. **Regression Analysis**
    - Simple linear regression
    - Multiple regression
    - Model diagnostics
5. **Exploratory Data Analysis**
    - Descriptive statistics

- Data visualization
- Outlier detection

Assessment:

- Homework assignments and problem-solving exercises
- Quizzes and exams
- Data analysis projects using real-world datasets

Recommended Reading:

1. “Introduction to Probability” by Dimitri P. Bertsekas and John N. Tsitsiklis
2. “Statistics” by Robert S. Witte and John S. Witte

Course Title: Digital Electronics and Lab

Course Description:

Digital Electronics and Lab is a foundational course that covers the principles of digital circuits, logic design, and practical implementation. In this course, students will learn about binary representation, combinational and sequential logic, and how to design and analyze digital systems. The lab component provides hands-on experience with electronic components and circuitry.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the basics of digital logic and electronic circuits.
2. Design and analyze combinational and sequential circuits.
3. Implement digital systems using hardware components.

Topics Covered:

1. **Number Systems and Binary Logic**
   - Binary, octal, and hexadecimal representation
   - Boolean algebra and logic gates
   - Karnaugh maps
2. **Combinational Logic Design**
   - Combinational circuits (adders, multiplexers, decoders)
   - Minimization techniques
   - VHDL or Verilog for logic synthesis
3. **Sequential Logic Design**
   - Flip-flops and registers
   - State machines and finite state automata
   - Timing diagrams
4. **Digital System Implementation**
   - Integrated circuits (ICs)
   - Breadboard prototyping
   - Troubleshooting and debugging

Lab Sessions:

- Hands-on experiments with digital components (logic gates, flip-flops, counters)
- Design and testing of digital circuits

Assessment:

- Homework assignments and problem-solving exercises
- Quizzes and exams
- Lab reports and practical assessments

Recommended Reading:

1. “Digital Design” by M. Morris Mano and Michael D. Ciletti
2. “Digital Electronics: Principles and Applications” by Roger L. Tokheim

Course Title: Unified Robotics II

Course Description:

Unified Robotics II is the second part of a two-course sequence that builds upon the foundational concepts introduced in Unified Robotics I. In this course, students will delve deeper into robotics engineering, focusing on advanced topics related to control systems, perception, and autonomous behavior. The course integrates theory, practical applications, and hands-on projects.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand advanced control techniques for robotic systems.
2. Analyze sensor data and implement perception algorithms.
3. Design and develop autonomous robotic behaviors.

Topics Covered:

1. **Advanced Control Systems**
   - PID controllers and tuning
   - State-space representation
   - Adaptive control
2. **Robot Perception**
   - Sensor fusion (lidar, cameras, IMUs)
   - Object detection and tracking
   - Simultaneous localization and mapping (SLAM)
3. **Path Planning and Navigation**
   - Motion planning algorithms (A*, RRT, D* Lite)
   - Trajectory generation
   - Obstacle avoidance
4. **Autonomous Behavior Design**
   - Behavior trees
   - Finite state machines
   - Reinforcement learning for robotics

Lab Sessions:

- Implementation of control algorithms on robotic platforms
- Sensor integration and data processing
- Autonomous navigation experiments

Assessment:

- Project-based assignments (e.g., designing a robot controller, implementing SLAM)
- Quizzes and exams
- Participation in lab activities and demonstrations

Recommended Reading:

1. "Robotics, Vision and Control: Fundamental Algorithms in MATLAB" by Peter Corke
2. Research papers and case studies on specific robotics applications

Course Title: Measurement Systems

Course Description:

Measurement Systems provides essential knowledge and skills related to sensors, instrumentation, and data acquisition. In this course, students will learn about measurement principles, calibration techniques, and the design of measurement systems. The focus will be on practical applications relevant to robotics engineering.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the fundamental concepts of measurement theory.
2. Select appropriate sensors for specific applications.
3. Design and implement measurement systems for robotic devices.

Topics Covered:

1. **Introduction to Measurement Systems**
    - Measurement standards and units
    - Accuracy, precision, and resolution
    - Error analysis and uncertainty
2. **Sensors and Transducers**
    - Types of sensors (temperature, pressure, displacement, etc.)
    - Sensor characteristics (linearity, sensitivity, hysteresis)
    - Signal conditioning and amplification
3. **Instrumentation and Data Acquisition**
    - Analog and digital data acquisition systems
    - Sampling rates and aliasing
    - LabVIEW or Python for data acquisition
4. **Calibration and Metrology**
    - Calibration methods (static, dynamic, traceability)
    - Measurement traceability and standards
    - Measurement uncertainty estimation
5. **Measurement System Design**

- Sensor selection for robotics applications
- Integration of sensors into robotic platforms
- Real-time measurement and control

Assessment:

- Homework assignments and problem-solving exercises
- Quizzes and exams
- Hands-on lab sessions (sensor calibration, data acquisition)

Recommended Reading:

1. “Introduction to Measurement Systems” by Ernest O. Doebelin
2. Research papers and case studies on sensor applications in robotics

Course Title: Linear Algebra

Course Description:

Linear Algebra is a foundational mathematics course that explores vector spaces, linear transformations, and matrix algebra. In this course, students will learn techniques for solving systems of linear equations, understanding geometric transformations, and applying linear algebra concepts to robotics and engineering problems.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the fundamental concepts of vector spaces and linear transformations.
2. Analyze matrices, determinants, and eigenvalues.
3. Apply linear algebra techniques to solve engineering problems.

Topics Covered:

1. **Vectors and Vector Spaces**
   - Vector operations (addition, scalar multiplication)
   - Linear independence and basis
   - Subspaces and dimension
2. **Matrix Algebra**
   - Matrix operations (addition, multiplication)
   - Inverse matrices
   - Rank and nullity
3. **Systems of Linear Equations**
   - Gaussian elimination
   - Row echelon form
   - Homogeneous and nonhomogeneous systems
4. **Eigenvalues and Eigenvectors**
   - Diagonalization
   - Applications in robotics (e.g., stability analysis)
5. **Geometric Transformations**

- Linear mappings
- Rotation, reflection, and scaling
- Affine transformations

Assessment:

- Homework assignments and problem-solving exercises
- Quizzes and exams
- Application of linear algebra concepts to robotic systems

Recommended Reading:

1. “Linear Algebra and Its Applications” by Gilbert Strang
2. “Introduction to Linear Algebra” by Serge Lang

Course Title: Introduction to Projects

Course Description:

The “Introduction to Projects” course provides students with practical experience in planning, executing, and managing engineering projects. Students will work on real-world projects related to robotics and automation, applying their knowledge from various disciplines. The course emphasizes teamwork, problem-solving, and project management skills.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the project lifecycle and project management principles.
2. Apply engineering concepts to solve project-specific challenges.
3. Collaborate effectively in multidisciplinary teams.

Topics Covered:

1. **Project Initiation and Planning**
    - Defining project scope and objectives
    - Stakeholder analysis
    - Work breakdown structure (WBS)
2. **Project Execution and Monitoring**
    - Task scheduling and resource allocation
    - Risk assessment and mitigation
    - Progress tracking and reporting
3. **Teamwork and Communication**
    - Effective team dynamics
    - Conflict resolution
    - Presenting project updates
4. **Robotics and Automation Projects**
    - Designing and building robotic prototypes
    - Implementing control algorithms
    - Testing and validation

Assessment:

- Project proposals and feasibility studies
- Regular progress reports
- Final project presentation and documentation

Recommended Reading:

1. “Project Management: A Systems Approach to Planning, Scheduling, and Controlling” by Harold Kerzner
2. Research papers and case studies on robotics projects

Course Title: Microprocessors

Course Description:

The Microprocessors course provides foundational knowledge about microcontrollers, programming, and interfacing. In this course, students will learn about the architecture, operation, and applications of microprocessors commonly used in robotics and automation systems.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the fundamental concepts of microprocessors and microcontrollers.
2. Program and interface with microcontrollers for real-world applications.
3. Apply microprocessor knowledge to robotics projects.

Topics Covered:

1. **Introduction to Microprocessors**
    - Historical context and evolution
    - Basic components (CPU, memory, I/O)
    - Instruction set architecture
2. **Microcontroller Architecture**
    - Harvard vs. von Neumann architecture
    - Registers and memory organization
    - Addressing modes
3. **Assembly Language Programming**
    - Instruction formats
    - Data movement and arithmetic instructions
    - Conditional branching
4. **Peripheral Interfacing**
    - Input/output ports
    - Timers and counters
    - Analog-to-digital conversion
5. **Real-world Applications**
    - Embedded systems design
    - Sensor integration
    - Motor control and robotics

Assessment:

- Programming assignments (assembly language programming)
- Lab sessions (microcontroller interfacing and projects)
- Quizzes and exams

Recommended Reading:

1. "The 8051 Microcontroller and Embedded Systems" by Muhammad Ali Mazidi, Janice Gillispie Mazidi, and Rolin D. McKinlay
2. Research papers and case studies on microcontroller applications in robotics

Course Title: Unified Robotics III

Course Description:

Unified Robotics III builds upon the foundational knowledge gained in Unified Robotics I and II. In this advanced course, students will explore specialized topics related to robotics engineering, focusing on manipulation, computer vision, and autonomous behavior. The course integrates theory, practical applications, and hands-on projects.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand advanced control techniques for robotic manipulation.
2. Analyze and implement computer vision algorithms for perception.
3. Design and develop autonomous robotic behaviors in complex environments.

Topics Covered:

1. **Robotic Manipulation**
   - Kinematics of robotic arms
   - End-effector design and grippers
   - Trajectory planning for precise manipulation
2. **Computer Vision for Robotics**
   - Image processing techniques
   - Object detection and recognition
   - Camera calibration and 3D reconstruction
3. **Perception and Sensing**
   - Sensor fusion (lidar, cameras, depth sensors)
   - Simultaneous localization and mapping (SLAM)
   - Perception-driven control
4. **Autonomous Behavior in Unstructured Environments**
   - Path planning in cluttered spaces
   - Obstacle avoidance and dynamic environments
   - Human-robot interaction

Lab Sessions:

- Implementation of manipulation algorithms on robotic arms

- Computer vision experiments using real-world data
- Autonomous navigation and interaction scenarios

Assessment:

- Project-based assignments (e.g., designing a pick-and-place robot, implementing SLAM)
- Quizzes and exams
- Participation in lab activities and demonstrations

Recommended Reading:

1. “Robotics, Vision and Control: Fundamental Algorithms in MATLAB” by Peter Corke
2. Research papers and case studies on advanced robotics applications

Course Title: Discrete Mathematics

Course Description:

Discrete Mathematics provides essential mathematical foundations for computer science, engineering, and robotics. In this course, students will explore topics related to logic, set theory, graph theory, combinatorics, and algorithms. The focus will be on discrete structures and their applications in solving real-world problems.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand fundamental concepts in discrete mathematics.
2. Apply discrete structures to algorithm design and problem-solving.
3. Analyze and reason about discrete systems.

Topics Covered:

1. **Logic and Proof Techniques**
   - Propositional logic
   - Predicate logic
   - Mathematical induction
2. **Set Theory and Relations**
   - Sets, subsets, and power sets
   - Cartesian products
   - Equivalence relations
3. **Graph Theory**
   - Graph terminology (vertices, edges, paths)
   - Graph representations (adjacency matrix, adjacency list)
   - Graph algorithms (BFS, DFS, shortest paths)
4. **Combinatorics**
   - Counting principles (permutations, combinations)
   - Pigeonhole principle
   - Generating functions
5. **Algorithms and Complexity**
   - Big O notation
   - Algorithm analysis

- NP-completeness

Assessment:

- Homework assignments and problem-solving exercises
- Quizzes and exams
- Application of discrete mathematics concepts to algorithmic problems

Recommended Reading:

1. "Discrete Mathematics and Its Applications" by Kenneth H. Rosen
2. Research papers and case studies on discrete structures in robotics and automation

Course Title: System Modeling and Control

Course Description:

System Modeling and Control provides essential knowledge about mathematical modeling, system analysis, and control techniques. In this course, students will learn how to represent dynamic systems, design controllers, and optimize performance. The focus will be on applications relevant to robotics and automation.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the principles of system modeling and control theory.
2. Analyze linear and nonlinear systems using mathematical tools.
3. Design and implement control strategies for robotic systems.

Topics Covered:

1. **Mathematical Modeling of Dynamic Systems**
   - Differential equations and transfer functions
   - State-space representation
   - Linearization techniques
2. **System Analysis and Stability**
   - Frequency response analysis
   - Bode plots and Nyquist diagrams
   - Routh-Hurwitz stability criterion
3. **Control System Design**
   - PID controllers
   - Root locus analysis
   - Pole placement techniques
4. **Advanced Control Strategies**
   - State feedback and observer design
   - Optimal control (LQR)
   - Adaptive and robust control
5. **Applications in Robotics and Automation**
   - Robot arm control

- Trajectory tracking
- Motion planning

Assessment:

- Homework assignments and problem-solving exercises
- Quizzes and exams
- Control system design projects

Recommended Reading:

1. "Feedback Control of Dynamic Systems" by Gene Franklin and J. Da Powell
2. Research papers and case studies on control applications in robotics

Title: Engineering Cost Analysis

Course Description:

Engineering Cost Analysis provides students with the knowledge and skills necessary to evaluate costs associated with engineering projects. In this course, students will learn techniques for estimating, analyzing, and managing costs throughout the project lifecycle. The focus will be on practical applications relevant to robotics engineering.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the principles of cost estimation and analysis.
2. Apply cost management techniques to engineering projects.
3. Make informed decisions based on cost considerations.

Topics Covered:

1. **Cost Estimation Methods**
    - Historical data analysis
    - Parametric models
    - Expert judgment
2. **Cost Components**
    - Direct costs (materials, labor, equipment)
    - Indirect costs (overhead, administrative expenses)
    - Contingency and risk allowances
3. **Life Cycle Cost Analysis**
    - Total cost of ownership
    - Discounted cash flow analysis
    - Sensitivity analysis
4. **Cost Control and Monitoring**
    - Earned value management
    - Change order management
    - Cost tracking and reporting
5. **Case Studies in Robotics Engineering**
    - Cost analysis for robot development
    - Budgeting for automation projects

    - Cost-benefit analysis of robotic systems

Assessment:

- Cost estimation exercises and case studies
- Quizzes and exams
- Project-based assignments related to cost management

Recommended Reading:

1. "Project Management: A Systems Approach to Planning, Scheduling, and Controlling" by Harold Kerzner
2. Research papers and industry reports on cost analysis in robotics and automation

Course Title: Robotics Capstone Project I

Course Description:

The Robotics Capstone Project I is an advanced course where students apply their knowledge and skills acquired throughout their robotics engineering program. In this project-based course, students work in teams to design, develop, and implement a real-world robotics project. The focus is on integrating various aspects of robotics, including hardware, software, control, and perception.

Learning Outcomes:

By the end of this course, students should be able to:

1. **Apply Integrated Knowledge:** Integrate concepts from various robotics disciplines to solve complex problems.
2. **Project Management:** Plan, execute, and manage a robotics project within specified constraints.
3. **Collaborate Effectively:** Work collaboratively in multidisciplinary teams to achieve project goals.

Topics Covered:

1. **Project Proposal and Scope Definition**
    - Identifying project objectives
    - Defining project scope and deliverables
    - Forming project teams
2. **System Requirements and Design**
    - Identifying system components (sensors, actuators, controllers)
    - Creating system architecture
    - Developing detailed design specifications
3. **Hardware and Software Integration**
    - Selecting and procuring necessary components
    - Integrating sensors and actuators
    - Developing control algorithms
4. **Testing and Validation**
    - Conducting functional testing
    - Validating system performance

    - Iterative refinement based on test results
5. **Documentation and Presentation**
    - Creating project documentation (design documents, user manuals)
    - Preparing project presentations
    - Demonstrating project outcomes

Assessment:

- Project proposal and design documentation
- Regular progress reports and team meetings
- Final project demonstration and presentation

Recommended Reading:

1. "Robotics: Modelling, Planning and Control" by Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo
2. Research papers and case studies related to robotics projects

Course Title: Discrete Control

Course Description:

Discrete Control provides students with essential knowledge about control systems, feedback loops, and digital control techniques. In this course, students will learn how to design, analyze, and implement control strategies for discrete-time systems. The focus will be on practical applications relevant to robotics and automation.

Learning Outcomes:

By the end of this course, students should be able to:

1. Understand the principles of discrete-time control systems.
2. Analyze stability, performance, and robustness of digital controllers.
3. Apply control techniques to real-world robotic systems.

Topics Covered:

1. **Introduction to Discrete Control Systems**
    - Difference equations and z-transform
    - Sampled-data systems
    - Digital vs. analog control
2. **Discrete-Time Modeling and Analysis**
    - State-space representation
    - Transfer functions in the z-domain
    - Stability analysis (BIBO stability, Routh-Hurwitz criteria)
3. **Digital Controller Design**
    - PID controllers in discrete-time
    - Pole placement methods
    - Dead-time compensation
4. **Performance Specifications and Robustness**
    - Settling time, overshoot, and steady-state error
    - Sensitivity functions

    - Controller tuning
5. **Applications in Robotics and Automation**
    - Motor control using digital controllers
    - Digital filters for sensor signal processing
    - Adaptive and predictive control

Assessment:

- Homework assignments and problem-solving exercises
- Quizzes and exams
- Control system design projects with simulation or hardware implementation

Recommended Reading:

1. "Digital Control of Dynamic Systems" by Gene Franklin and J. Da Powell
2. Research papers and case studies on digital control applications in robotics

Course Title: Unified Robotics IV

Course Description:

Unified Robotics IV is an advanced course that builds upon the foundational knowledge gained in Unified Robotics I, II, and III. In this capstone project course, students work collaboratively on complex robotics projects, integrating theory, practical skills, and interdisciplinary expertise. The focus is on solving real-world challenges using robotics technology.

Learning Outcomes:

By the end of this course, students should be able to:

1. **Project Leadership:** Lead and manage multidisciplinary robotics projects effectively.
2. **Advanced Problem-Solving:** Apply advanced robotics concepts to address complex engineering problems.
3. **Innovation and Creativity:** Design novel solutions and demonstrate innovation in robotics applications.

Topics Covered:

1. **Project Proposal and Planning**
    - Defining project scope and objectives
    - Identifying key deliverables and milestones
    - Allocating resources and forming project teams
2. **Advanced Robotics Algorithms and Techniques**
    - Reinforcement learning for robotics
    - Computer vision for perception
    - Advanced control strategies (adaptive, optimal)
3. **Hardware Integration and Prototyping**
    - Selecting and integrating sensors and actuators
    - Rapid prototyping techniques
    - Safety considerations
4. **Testing, Validation, and Optimization**
    - Rigorous testing protocols

    - Performance evaluation and optimization
    - Iterative refinement based on test results
5. **Documentation and Final Presentation**
    - Comprehensive project documentation
    - Final project demonstration and presentation
    - Lessons learned and future recommendations

Assessment:

- Project proposal and detailed project plan
- Regular progress reports and team meetings
- Final project demonstration and comprehensive documentation

Recommended Reading:

1. "Robotics: Modelling, Planning and Control" by Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo
2. Research papers and case studies on advanced robotics applications

Course Title: Data Structures

Course Description:

This course introduces fundamental data structures used in computer science and robotics. Students will learn how to design, implement, and analyze data structures to solve real-world problems. Topics include arrays, linked lists, trees, graphs, sorting algorithms, and their applications in robotics.

Learning Objectives:

By the end of this course, students should be able to:

- Understand the importance of data structures in programming.
- Analyze time and space complexity of algorithms.
- Implement and manipulate arrays, linked lists, stacks, and queues.
- Apply data structures to robotics problems.

Topics Covered:

1. **Introduction to Data Structures**
    - Overview of data structures.
    - Basic terminology.
    - Time and space complexity analysis.
2. **Arrays and Linked Lists**
    - Array implementation.
    - Dynamic arrays.
    - Singly and doubly linked lists.
3. **Stacks and Queues**
    - Stack (LIFO).
    - Queue (FIFO).
    - Applications in robotics algorithms.
4. **Trees and Graphs**

- Binary trees.
    - Binary search trees.
    - Graph representation.
5. **Sorting and Searching Algorithms**
    - Bubble sort, selection sort, insertion sort.
    - Quick sort, merge sort, heap sort.
    - Binary search and linear search.
6. **Hashing and Hash Tables**
    - Hash functions.
    - Collision resolution.
    - Applications in robot path planning.
7. **Advanced Topics**
    - Priority queues.
    - Graph algorithms (Dijkstra's, A*, Floyd-Warshall).
    - Space-efficient data structures.
8. **Robotics-Specific Applications**
    - Data structures for robot kinematics, sensor fusion, and motion planning.
    - Real-world examples and case studies.

Assessment:

- Assignments, quizzes, and exams.
- Practical implementation of data structures in robotics projects.

Recommended Reading:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. MIT Press.
2. Weiss, M. A. (2014). *Data Structures and Algorithm Analysis in C++*. Pearson.

Course Title: Robotics Capstone Project II

Course Description:

In this capstone course, students will apply their knowledge and skills acquired throughout their robotics specialization to solve real-world problems. Working in teams, students will design, implement, and test a robotics project, addressing specific challenges related to robotics systems, control, perception, and autonomy.

Learning Objectives:

By the end of this course, students should be able to:

- Apply mathematical and programming methods used in robotics research.
- Collaborate effectively in a team-based project.
- Design and implement a functional robotics system.
- Evaluate and optimize their project based on performance metrics.

Topics Covered:

1. **Project Selection and Proposal**
    - Identifying a robotics problem or application.

    - Writing a project proposal.
    - Defining project scope and objectives.
2. **System Architecture and Design**
    - High-level system architecture.
    - Component selection (sensors, actuators, controllers).
    - Safety considerations.
3. **Perception and Sensing**
    - Sensor integration (lidar, cameras, IMUs).
    - Data fusion and filtering.
    - Object detection and tracking.
4. **Control and Actuation**
    - PID control.
    - Trajectory planning.
    - Motor control and actuator calibration.
5. **Localization and Mapping**
    - Simultaneous Localization and Mapping (SLAM).
    - Odometry and dead reckoning.
    - Map representation (grid-based, feature-based).
6. **Autonomy and Decision-Making**
    - Path planning algorithms (A*, RRT).
    - Behavior-based control.
    - Collision avoidance.
7. **Testing and Evaluation**
    - Real-world testing.
    - Performance metrics (accuracy, speed, robustness).
    - Iterative improvements.
8. **Documentation and Presentation**
    - Technical report.
    - Project demonstration.
    - Lessons learned and future work.

Assessment:

- Project milestones (proposal, design, implementation, testing).
- Final project demonstration and presentation.
- Peer evaluations.

Recommended Reading:

1. Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics*. MIT Press.
2. Siciliano, B., & Khatib, O. (2008). *Springer Handbook of Robotics*. Springer.

Course Title: Embedded Systems

Course Description:

This course focuses on the design, implementation, and optimization of embedded systems used in robotics. Students will learn how to develop software and hardware solutions for real-time control, sensing, and communication in robotic applications.

Learning Objectives:

By the end of this course, students should be able to:

- Understand the fundamentals of embedded systems.
- Design and program microcontrollers and microprocessors.
- Interface sensors, actuators, and communication modules.
- Apply real-time operating systems (RTOS) for robotics.

Topics Covered:

1. **Introduction to Embedded Systems**
    - Definition and characteristics of embedded systems.
    - Hardware components (microcontrollers, sensors, actuators).
    - Software development for embedded platforms.
2. **Microcontroller Programming**
    - Embedded C programming.
    - GPIO (General Purpose Input/Output) control.
    - Interrupt handling.
3. **Real-Time Operating Systems (RTOS)**
    - Basics of RTOS.
    - Task scheduling and synchronization.
    - Case studies with RTOS in robotics.
4. **Sensor Interfacing**
    - Analog and digital sensors (IR, ultrasonic, encoders).
    - ADC (Analog-to-Digital Conversion).
    - Sensor fusion techniques.
5. **Actuator Control**
    - Motor control (DC, servo, stepper).
    - PWM (Pulse Width Modulation).
    - Feedback control loops.
6. **Communication Protocols**
    - UART, SPI, I2C.
    - Wireless communication (Bluetooth, Wi-Fi).
    - Network protocols for distributed systems.
7. **Embedded System Design**
    - System architecture.
    - Power management.
    - Safety considerations.
8. **Robotics Applications**
    - Implementing embedded systems in robot control.
    - Real-world examples (robot arms, drones, autonomous vehicles).

Assessment:

- Lab assignments (programming microcontrollers).
- Embedded system project (design and implementation).
- Written exams.

Recommended Reading:

1. Lee, E. A., & Seshia, S. A. (2011). *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. MIT Press.

2. Valvano, J. W. (2012). *Embedded Systems: Introduction to ARM Cortex-M Microcontrollers*. Cengage Learning.

Course Title: Algorithms

Course Description:

This course focuses on the study of algorithms, their design, analysis, and implementation. Students will learn fundamental algorithms used in robotics, optimization, and problem-solving. The course emphasizes both theoretical understanding and practical application.

Learning Objectives:

By the end of this course, students should be able to:

- Understand algorithmic paradigms (greedy, divide and conquer, dynamic programming).
- Analyze time and space complexity.
- Implement algorithms in programming languages.
- Apply algorithms to robotics problems.

Topics Covered:

1. **Introduction to Algorithms**
    - Importance of algorithms in computer science and robotics.
    - Algorithmic problem-solving approaches.
2. **Algorithm Design Techniques**
    - Greedy algorithms.
    - Divide and conquer.
    - Dynamic programming.
3. **Sorting and Searching Algorithms**
    - Bubble sort, insertion sort, merge sort.
    - Binary search, linear search.
4. **Graph Algorithms**
    - Breadth-First Search (BFS).
    - Depth-First Search (DFS).
    - Shortest path algorithms (Dijkstra's, Bellman-Ford).
5. **Dynamic Programming**
    - Memoization.
    - Optimal substructure.
6. **Data Structures and Algorithms for Robotics**
    - Data structures (arrays, linked lists, trees).
    - Path planning algorithms.
    - Sensor data processing.
7. **Optimization Algorithms**
    - Genetic algorithms.
    - Simulated annealing.
    - Particle swarm optimization.
8. **Real-world Applications**
    - Robot motion planning.
    - Image processing.
    - Swarm robotics.

Assessment:

- Assignments and coding exercises.
- Algorithm analysis and implementation projects.
- Written exams.

Recommended Reading:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. MIT Press.
2. Skiena, S. S. (2008). *The Algorithm Design Manual*. Springer.

Course Title: Accelerated Object-Oriented Design Concepts

Course Description:

This course covers advanced topics related to object-oriented design principles and data structures. It assumes that students have significant prior experience in object-oriented programming. The course aims to deepen understanding and enhance algorithmic thinking in the context of robotics.

Learning Objectives:

By the end of this course, students should be able to:

- Apply object-oriented design principles effectively.
- Analyze and optimize data structures for robotics applications.
- Develop robust and modular software solutions.

Topics Covered:

1. **Review of Object-Oriented Programming**
   - Recap of key OOP concepts (classes, inheritance, polymorphism).
   - Advanced Java programming techniques.
2. **Design Patterns**
   - Creational patterns (singleton, factory, builder).
   - Structural patterns (adapter, decorator, composite).
   - Behavioral patterns (observer, strategy, command).
3. **Advanced Data Structures**
   - Trees (binary trees, AVL trees).
   - Graphs (graph traversal, shortest paths).
   - Hashing and collision resolution.
4. **Software Architecture**
   - Layered architecture.
   - Model-View-Controller (MVC) pattern.
   - Microservices and modularity.
5. **Testing and Debugging**
   - Unit testing.
   - Debugging strategies.
   - Code quality metrics.
6. **Real-world Case Studies**
   - Robotics software design examples.

    - Performance optimization.
    - Scalability considerations.

Assessment:

- Coding assignments (design and implementation).
- Design critiques and code reviews.
- Final project applying design principles to a robotics problem.

Recommended Background:

- Proficiency in Java programming.
- Experience writing object-oriented programs from scratch.

Course Title: Leadership and Professional Development for Engineers

Course Description:

This course focuses on developing essential leadership skills and enhancing professional competencies for engineers in the field of robotics. Students will learn effective communication, teamwork, project management, and ethical decision-making, preparing them for successful careers in engineering leadership.

Learning Objectives:

By the end of this course, students should be able to:

- Understand the role of leadership in engineering organizations.
- Communicate effectively with diverse teams.
- Apply project management principles.
- Navigate ethical challenges in engineering practice.

Topics Covered:

1. **Introduction to Engineering Leadership**
    - Importance of leadership skills for engineers.
    - Leadership styles and self-awareness.
2. **Effective Communication**
    - Written and oral communication.
    - Presentations and technical reports.
    - Cross-cultural communication.
3. **Teamwork and Collaboration**
    - Building and leading effective teams.
    - Conflict resolution.
    - Interdisciplinary collaboration.
4. **Project Management**
    - Project planning and scheduling.
    - Risk management.
    - Budgeting and resource allocation.
5. **Ethics and Professional Responsibility**
    - Engineering codes of ethics.
    - Case studies in ethical decision-making.
    - Social and environmental impact considerations.
6. **Career Development**

- Resume writing and job search strategies.
    - Interview skills.
    - Lifelong learning and professional growth.
7. **Industry Insights**
    - Guest lectures from industry professionals.
    - Site visits to robotics companies.
    - Networking opportunities.

Assessment: Group projects (leadership simulations, case studies). Individual assignments (communication exercises, ethical dilemmas).

Recommended Reading:

1. Covey, S. R. (2004). *The 7 Habits of Highly Effective People*. Free Press.
2. Goleman, D. (1995). *Emotional Intelligence*. Bantam.

Join Our Celestial Journey

Illuminating Minds, Igniting Innovation. Be Part of the Spark as we Unlock the Universe's Secrets, One Equation at a Time

Office of Institutional Planning and Effectiveness
The Knowledge Park I, Examination Council of Eswatini
P.O. Box D61, Ezulwini
Eswatini

www.springfieldresearch.university
frontdesk@springfieldresearch.university

+268 7619-2898
+268 2417-1634

Research Beyond Boundaries

SPRINGFIELD RESEARCH UNIVERSITY