Savitribai Phule Pune University, Pune

Maharashtra, India

Faculty of Science and Technology

National Education Policy (NEP)-2020 Compliant Curriculum

TE - Third Year Engineering (2024 Pattern) in

Robotics and Artificial Intelligence

(With effect from Academic Year 2026-27)

Contents

Nomenclature

AEC	Ability Enhancement Courses
AICTE	All India Council for Technical Education
CO	Course Outcome
CEP	Community Engagement Project
CCE	Comprehensive Continuous Evaluation
HSSM	Humanities, Social Science, and Management
MDM	Multidisciplinary Minor
RAI	Robotics and Artificial Intelligence
MOOC	Massive Open Online Course
NPTEL	National Programme on Technology Enhanced Learning
NEP	National Education Policy
OE	Open Elective
PCC	Program Core Course
PEO	Program Educational Objectives
PSO	Program Specific Objectives
SWAYAM	Study Webs of Active Learning for Young Aspiring Minds
UGC	University Grant Commission
VEC	Value Education Course
VSE	Vocational Skill Course
WK	Knowledge and Attitude Profile

Preface by Board of Studies

Dear Students and Teachers,

We, the members of the Board of Studies – Mechanical Engineering, are very happy to present the Third Year **Robotics and Artificial Intelligence** syllabus, effective from the Academic Year 2026–27 (2024 Pattern). We are confident that you will find this syllabus both interesting and challenging. The present curriculum will be implemented for Third Year Engineering from the academic year 2026–27, and it will be subsequently extended to the Final Year in the academic year 2027–28.

Robotics and Artificial Intelligence is one of the most sought-after branches among engineering students, which necessitates continuous revision and up gradation of the syllabus. The **Robotics and Artificial Intelligence** program equips graduates with strong technical expertise in robotics, AI, and intelligent systems. They develop research-oriented analytical skills, ethical leadership, and lifelong learning abilities to design, simulate, and implement autonomous robotic solutions. The program emphasizes real-world applications, integrating AI with mechanical systems for industrial, healthcare, and automation challenges.

The revised syllabus aligns with the vision of **NEP-2020**, and conforms to the frameworks set by Savitribai Phule Pune University, AICTE New Delhi, UGC, and various accreditation agencies. It takes into account recent technological developments, innovations, and industry needs to ensure students are well-prepared for professional challenges.

Wherever applicable, additional learning resources such as NPTEL and SWAYAM links are provided at the end of each course. Students are encouraged to utilize these platforms for self-learning, engage in online courses, and undertake additional projects to enhance their knowledge and skill set. On successful completion, they are advised to submit their course certifications, which will further support and enrich their academic growth.

This curriculum is the result of collaborative efforts involving academic experts, industry professionals, and alumni to ensure relevance and excellence. It is designed not only to meet current industry expectations but also to prepare students for higher studies, research, and entrepreneurial ventures in the field **Robotics and Artificial Intelligence.**

We hope this curriculum inspires students to become technically competent professionals, responsible citizens, and contributors to the technological and sustainable advancement of society.

Dr. Pradeep A. Patil
Chairman
Board of Studies - Mechanical Engineering

Program Specific Outcomes (PSO)

PSO1: Foundation and Application of Robotics

Graduates will be able to apply kinematics, dynamics, control systems, and mechanical design to model, analyze, and develop robotic systems for industrial and real-world applications.

PSO2: Integration of Robotics with Artificial Intelligence

Graduates will demonstrate the ability to integrate AI techniques (such as machine learning, computer vision, and intelligent control) with mechanical systems to develop autonomous and adaptive robotic solutions.

PSO3: Design and Development of Intelligent Systems

Graduates will possess the skills to design, simulate, and implement intelligent robotic systems by combining mechanical components with embedded systems, sensors, actuators, and AI algorithms to address real-world problems in manufacturing, healthcare, and automation.

Program Educational Objectives (PEO)

Program Educational Objectives (PEOs) are broad statements that describe the career and professional accomplishments that the program is preparing graduates to achieve.

PEO1: **Technical Expertise**	Graduates will demonstrate strong foundational and applied knowledge in Robotics and Artificial Intelligence, enabling them to solve complex engineering problems and contribute effectively to the modern industrial and technological landscape.
PEO2: **Research Orientation and Analytical Skills**	Graduates will be capable of adopting a scientific and research-based approach, employing mathematical models, engineering tools, and simulation techniques to design and evaluate intelligent robotic systems.
PEO3: **Ethical Leadership & Lifelong Growth**	Graduates will exhibit professionalism with ethical integrity, leadership, teamwork, and societal awareness, while demonstrating entrepreneurial thinking, a commitment to lifelong learning, and adaptability to technological changes for sustained career success.

Program Outcomes

Program Outcomes (POs) are statements that articulate what students are expected to know, understand, and be able to do by the time they graduate from the program. These outcomes are aligned with the overall educational objectives of the program and reflect the skills, knowledge, attitudes, and behaviors acquired by students throughout their academic journey. On successful completion of B.E. in **Robotics and Artificial Intelligence**, graduating students/graduates will be able to:

PO No.	Title	Program Outcome Description
PO1	Engineering Knowledge	Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop the solution of complex engineering problems.
PO2	Problem Analysis	Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
PO3	Design / Development of Solutions	Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for public health and safety, whole-life cost, net zero carbon, culture, society and environment. (WK5)
PO4	Conduct Investigations of Complex Problems	Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8)
PO5	Engineering Tool Usage	Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling, recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
PO6	The Engineer and The World	Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7)
PO7	Ethics	Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
PO8	Individual and Collaborative Team Work	Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
PO9	Communication	Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences.
PO10	Project Management and Finance	Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one’s own work, as a member and leader in a team, and to manage projects in multidisciplinary environments.
PO11	Life-Long Learning	Recognize the need for, and have the preparation and ability for: (i) independent and life-long learning, (ii) adaptability to new and emerging technologies, and (iii) critical thinking in the broadest context of technological change. (WK8)

Knowledge and Attitude Profile (WK)

WK No.	Title/Focus Area	Description
WK 1	Natural Sciences and Social Sciences	A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.
WK 2	Mathematics and Data Analysis	Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.
WK 3	Engineering Fundamentals	A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.
WK 4	Engineering Specialist Knowledge	Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.
WK 5	Engineering Design and Environmental Considerations	Knowledge, including efficient resource use, environmental impacts, whole- life cost, re-use of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.
WK 6	Engineering Practice (Technology)	Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.
WK 7	Role of Engineering in Society	Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.
WK8	Research and Critical Thinking	Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.
WK 9	Ethics and Inclusive Behavior	Ethics, inclusive behavior and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability, etc., with mutual respect.

Reference: Self-Assessment Report (SAR) Format Undergraduate Engineering Programs Graduate Attributes and Professional Competencies Version 4.0 (GAPC V4.0) – (August 2024) Page 55-56

General Rules and Guidelines

Term	Definition
Course Outcomes (COs)	Course Outcomes are narrower statements that describe what students are expected to know and be able to do at the end of each course. These relate to the skills, knowledge, and behavior that students acquire throughout the course.
Assessment	Assessment is one or more processes, carried out by the institution, that identify, collect, and prepare data to evaluate the achievement of **Program Educational Objectives (PEOs)** and **Program Outcomes (POs)**.
Evaluation	Evaluation is one or more processes, performed by the **Evaluation Team**, to interpret the data and evidence gathered through assessment practices. It determines how well PEOs or POs are being achieved, and informs decisions for improvement.

Assessment and Evaluation:

Assessment and Evaluation shall be conducted in two parts:
1. Comprehensive Continuous Evaluation (CCE)
2. End-Semester Examination (ESE)

Component	Description	Marks
Comprehensive Continuous Evaluation (CCE)	Conducted at institute level, covering all Units of the syllabus. The design and mark allocation follow the Continuous Assessment Sheet structure.	30
End-Semester Examination (ESE)	Conducted at university level, typically covering the entire syllabus through summative examination.	70
Total Marks per Subject		**100**

A) <u>Comprehensive Continuous Evaluation (CCE)</u>: It can be conducted via Mode 1 or Mode 2

CCE Mode 1:

To design a Comprehensive Continuous Evaluation (CCE) scheme for a theory subject of 30 marks with the specified parameters, the allocation of marks and the structure can be as per continuous assessment sheet;

Savitribai Phule Pune University																					
Board of Studies (Mechanical and Automobile Engineering)																					
Comprehensive Continous Evaluation (CCE) 30 Marks Distribution																					
Class: TE-Mechanical Engineering																Subject: Refrigeration and Air conditioning					
Exam Seat No.	Roll No.	Name of Student	Unit 1		Unit 2		Unit 3		Unit 4		Unit 5		Cumulative Sum				30 Marks Distribution				Marks obtained out
			Field Activity	Quiz	Field Activity	Quiz	Field Activity	Quiz	Field Activity	Quiz	Field Activity	Quiz	Field Activity	Quiz	Internal Test	Attendance	Field Activity	Quiz	Internal Test	Attendance	
			A	B	C	D	E	F	G	H	I	J	SUM(A+C+E+G+I)	SUM(B+D+F+H+J)			P	Q	R	S	SUM(P:Q)
			10	10	10	10	10	10	10	10	10	10	50	50	100	100	15	5	5	5	30
T9028246474	2016	MAAN PANCHAL	8	8	8	8	8	8	8	8	8	8	40	40	60	75	12	4	3	3.8	22.75

Comprehensive Continous Evaluation (CCE) 15 Marks Distribution																					
Exam Seat No.	Roll No.	Name of Student	Unit 1		Unit 2		Unit 3		Unit 4				Cumulative Sum				15 Marks Distribution				Marks obtained out
			Field Activity	Quiz	Field Activity	Quiz	Field Activity	Quiz	Field Activity	Quiz			Field Activity	Quiz	Internal Test		Field Activity	Quiz	Internal Test		
			A	B	C	D	E	F	G	H			SUM(A+C+E+G	SUM(B+D+F+H			P	Q	R	S	SUM(P:Q)
			10	10	10	10	10	10	10	10			40	40	100		5	5	5		15
T9970160753	2020	AMOGH SHINDE	8	8	8	8	8	8	8	8			32	32	60		4	4	3		11

Field Activities / Home Assignments

Field activities and home assignments are essential components of experiential learning. Under this head, course projects, industrial visits, and guest lectures are to be incorporated. For each unit, one such activity should be designed and executed to reinforce theoretical learning through practical exposure.

1. **Course Projects**

   Course Projects should be framed based on real-world problems relevant to the subject. Each course project must be communicated through one of the following modes. It is recommended to complete all the communication modes across different course projects:

   - **Poster Presentation**
   - **PowerPoint Presentation**
   - **Model Making**
   - **Field or Survey Report with Oral Presentation** (e.g., case study)
   - **Submission of Digital Content** (e. g. Video Summary)

   To evaluate these field activities, **assessment rubrics** should be designed. The rubrics should include criteria such as clarity, innovation, subject relevance, presentation skills, and technical content.

   Note: Part of work of any co-curricular activities (relevant to subject contents) like national level project competitions, club activities, paper presentations, startup activities can be accepted as a course projects.

2. **Industrial Visit**

   An industrial visit should be planned in alignment with the subject's scope and should particularly address advancements in the respective field. The purpose is to provide students exposure to actual engineering practices and systems. Assessment of industrial visits should be carried out using any of the following tools:

- **Quiz (based on the visit)**
- **Interactive video or oral discussion**
- **Submission of a detailed visit report**

3. **Guest Lectures**

Guest lectures should be relevant to the course and highlight advanced topics or recent trends in the field. Subject experts from academia or industry may be invited.

Assessment methods for guest lectures may include:

- **Quiz conducted post-lecture**
- **Attendance monitoring**
- **Evaluation of attentiveness and participation**

Rubrics can be developed, if possible, to objectively assess student involvement in guest lectures.

4. **Quiz**

Unit-wise quizzes should be planned and can be conducted either **online** (via LMS, Google Forms) or **offline**. Each quiz should include a **pool of 20 questions**, from which **students are required to attempt any 10**. The quizzes should be diversified across the following question types:

- **Simple Multiple-Choice Questions (MCQs)**
- **Numerical MCQs**
- **Image-based Questions**
- **Match the Following**
- **Fill in the Blanks**
- **Drag and Drop (using images or words)**

This variety ensures the assessment caters to different cognitive skills and learning styles.

5. **Internal Tests**

Two major internal tests should be conducted as follows:

1. **Midterm Examination**: This should cover **Unit I and Unit II**, and should include questions targeting **Bloom's Taxonomy Levels 2, 3, and 4** (UNDERSTAND, APPLY, and ANALYZE).
2. **End term Examination**: This should cover the **remaining units** and should also include questions mapped to **BL Levels 2, 3, and 4**.

CCE Mode 2:

Comprehensive Continuous Evaluation (CCE) scheme in this mode for a theory subject of 30 marks with the specified parameters, the allocation of marks and the structure can be detailed as follows:

Sr. No.	Parameters	Marks	Coverage of Units
1	Unit Test	12 Marks	Unit 1 & Unit 2 (6 Marks/Unit)
2	Assignments/Case Study	12 Marks	Unit 3 & Unit 4 (6 Marks/Unit)
3	Seminar Presentation/ Open Book Test/ Quiz	06 Marks	Unit 5

CCE of 15 marks based on all the Units of course syllabus to be scheduled and conducted at institute level. To design a CCE scheme for a theory subject of 15 marks with the specified parameters, the allocation of marks and the structure can be detailed as follows:

Sr. No.	Parameters	Marks	Coverage of Units
1	Unit Test	10 Marks	Unit 1 & Unit 2 (5 Marks/Unit)
2	Seminar Presentation/ Open Book Test/ Assignments/ Case Studies	05 Marks	Unit 3 & Unit 4

1. **Unit Test:**

   Format: Questions designed as per Bloom's Taxonomy guidelines to assess various cognitive levels (Remember, Understand, Apply, Analyze, Evaluate, and Create).

   Implementation: Schedule the test after completing Units 1 and 2. Ensure the question paper is balanced and covers key concepts and applications.

   - *Sample Question Distribution:*
     1. Remembering (2 Marks): Define key terms related to [Topic from Units 1 and 2].
     2. Understanding (2 Marks): Explain the principle of [Concept] in [Context].
     3. Applying (2 Marks): Demonstrate how [Concept] can be used in [Scenario]
     4. Analyzing (3 Marks): Compare & contrast [Two related concepts] from Units 1 and 2
     5. Evaluating (3 Marks): Evaluate the effectiveness of [Theory/Model] in [Situation].

2. **Assignments / Case Study:** Students should submit one assignment or one Case Study Report based on Unit 3 and one assignment or one Case Study Report based on Unit 4.

   i. ***Format*:** Problem-solving tasks, theoretical questions, practical exercises, or case studies that require in-depth analysis and application of concepts.

ii. ***Implementation:*** Distribute the assignments or case study after covering Units 3 and 4. Provide clear guidelines and a rubric for evaluation.

3. **Seminar Presentation:**

Format: Oral presentation on a topic from Unit 5, followed by a Q&A session.

***Deliverables*:** Presentation slides, a summary report in 2 to 3 pages, and performance during the presentation

Implementation: Schedule the seminar presentations towards the end of the course. Provide students with ample time to prepare and offer guidance on presentation skills.

4. **Open Book Test:**

Format: Analytical and application-based questions to assess depth of understanding. ***Implementation***: Schedule the open book test towards the end of the course, ensuring it covers critical aspects of Unit 5

5. **Quiz:**

Format: Quizzes can help your students practice existing knowledge while stimulating interest in learning about new topic in that course. You can set your quizzes to be completed individually or in small groups.

Implementation: Online tools and software can be used create quiz. Each quiz is made up of a variety of question types including multiple choice, missing words, true or false etc.

- ***Example Timeline for conducting CCE:***

1. Weeks 1-4 : Cover Units 1 and 2
2. Week 5 : Conduct Unit Test (12 marks)
3. Weeks 6-8 : Cover Units 3 and 4
4. Week 9 : Distribute and collect Assignments / Case Study (12 marks)
5. Weeks 10-12 : Cover Unit 5
6. Week 13 : Conduct Seminar Presentations or Open Book Test or Quiz (6 marks)

- ***Evaluation and Feedback:***

1. Unit Test: Evaluate promptly and provide constructive feedback on strengths and areas for improvement.
2. Assignments / Case Study: Assess the quality of submissions based on the provided rubric. Offer feedback to help students understand their performance.
3. Seminar Presentation: Evaluate based on content, delivery, and engagement during the Q&A session. Provide feedback on presentation skills and comprehension of the topic.
4. Open Book Test: Evaluate based on the depth of analysis and application of concepts. Provide feedback on critical thinking and problem-solving skills.

B) **End-Semester Examination (ESE)**

Detailed Scheme for 70 Marks: Unit-Wise Allocation (14 Marks per Unit): Each unit will have a combination of questions designed to assess different cognitive levels. By following this scheme, you can ensure a comprehensive and fair assessment of students' understanding and application of the course material, adhering to Bloom's Taxonomy guidelines for cognitive skills evaluation.

Detailed Scheme for 35 Marks: Unit-Wise Allocation (08 Marks for Unit 1, 09 Marks for Unit 2, Unit 3 and Unit 4): Each unit will have a combination of questions designed to assess different cognitive levels. By following this scheme, you can ensure a comprehensive and fair assessment of students' understanding and application of the course material, adhering to Bloom's Taxonomy guidelines for cognitive skills evaluation **Question Paper Design**.

The following structure is to be followed for designing an ESE for a **theory subject of 70 marks** covering all **5 units** of the syllabus, with **questions set as per Bloom's Taxonomy** guidelines and **14 marks allocated per unit.**

1. Balanced Coverage

Ensure balanced coverage of all units with questions that assess different **cognitive levels of Bloom's Taxonomy**:

a) **Remembering**: Basic recall of facts and concepts.
b) **Understanding**: Explanation of ideas or concepts.
c) **Applying**: Use of information in new situations.
d) **Analyzing**: Drawing connections among ideas.
e) **Evaluating**: Justifying a decision or course of action.
f) **Creating**: Producing new or original work (if applicable).

2. Detailed Scheme

Unit-Wise Allocation: **14 Marks per Unit**

Each unit will have a combination of questions designed to assess different cognitive levels. By following this scheme, you can ensure a comprehensive and fair assessment of students' understanding and application of the course material, adhering to **Bloom's Taxonomy guidelines** for cognitive skills evaluation.

NEP 2020 Compliant Curriculum Structure

Third Year Engineering (2024 Pattern)

Robotics & Artificial Intelligence

Course Code	Course Type	Course Name	Teaching Scheme (Hrs./week)			Examination Scheme and Marks						Credits			
			Theory	Tutorial	Practical	CCE	ESE	Term work	Oral	Practical	Total	Theory	Tutorial	Practical	Total
Semester V															
PCC301RAI	Major Course	Dynamics and Control of Robotic Systems	4			30	70				100	4			4
PCC302RAI	Major Course	Machine Learning for Robotics	3			30	70				100	3			3
PCC303RAI	Major Course	Robot Modeling and Simulation	4			30	70				100	4			4
PCC304RAI	Major Course	Advances in Robotics and Artificial Intelligence	3			30	70				100	3			3
PCC305RAI	Major Course	Design of Robot Elements	3			30	70				100	3			3
PCC306RAI	Major Course	Robot Programming Lab			2					50	50			1	1
MDM321RAI	Multidisciplinary Course	Intelligent Systems and Applications			2					50	50			1	1
MDM322RAI	Multidisciplinary Course	Autonomous Navigation using SLAM			2					50	50			1	1
	Open Elective	OE is to be chosen compulsorily from faculty other than that of the Major Discipline.	2			15	35				50	2			2
Total			**19**	**0**	**06**	**165**	**385**	**0**	**0**	**150**	**700**	**19**	**0**	**3**	**22**

CCE*: Comprehensive Continuous Evaluation

Important Note: Min.1 to Max.2 hrs. Per batch of (20-25 students) to be assigned for *CCE and to be considered in teaching load of concerned faculty. (Only applicable for Mode -1)

ESE*: End Semester Examination

Note: Students can opt for Open Electives offered by different faculties such as Arts, Science, Commerce, Management, Humanities, or Inter-Disciplinary Studies. Example – Open Elective III: Courses like Project Management, Business Analytics, or Financial Management from the Commerce and Management faculty can be opted from Inter-Disciplinary Studies, Commerce, and Management faculties, respectively.

NEP 2020 Compliant Curriculum Structure

Third Year Engineering (2024 Pattern)

Robotics & Artificial Intelligence

Course Code	Course Type	Course Name	Teaching Scheme (Hrs./week)			Examination Scheme and Marks						Credits			
			Theory	Tutorial	Practical	CCE	ESE	Term work	Oral	Practical	Total	Theory	Tutorial	Practical	Total
Semester VI															
PCC351RAI	Major Course	Deep Learning and Neutral Networks	4			30	70				100	4			4
PCC352RAI	Major Course	Robotic Process Automation	3			30	70				100	3			3
PCC353RAI	Major Course	Robot Operating System	4			30	70				100	4			4
PCC354RAI	Major Course	Robot Operating and Process Automation Lab			4					50	50			2	2
PEC361RAI	Program Elective I	Select the elective from basket	3			30	70				100	3			3
PEC362RAI	Program Elective II	Select the elective from basket	3			30	70				100	3			3
MDM371RAI	Multidisciplinary Course	Image Processing & Computer Vision			2					50	50			1	1
MDM372RAI	Multidisciplinary Course	Industrial Automation			2					50	50			1	1
VSE381RAI	Vocational Skill Course	Autonomous Drones			2	50					50	1			1
Total			**17**	**0**	**10**	**200**	**350**			**150**	**700**	**18**	**0**	**4**	**22**

Program Elective I		**Program Elective II**	
PEC361ARAI	Augmented Reality and Virtual Reality	PEC362ARAI	Advanced Artificial Intelligence
PEC361BRAI	Mobile and Micro Robotics	PEC362BRAI	Data Analytics
PEC361CRAI	Humanoid Robots	PEC362CRAI	Mechatronics System Design
PEC361DRAI	Agricultural Robotics	PEC362DRAI	AI for Medical Applications

Savitribai Phule Pune University, Pune

Maharashtra, India

TE - Robotics and Artificial Intelligence (2024 Pattern)

Semester V Courses

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PCC-301-RAI: Dynamics and Control of Robotic Systems				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**4 Hours/Week**	**4**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:

Basics of Robotics AI (PCC-201-RAI),
Kinematics of Robot (PCC-252-RAI)

Course Objectives:

The students will be able

1. To apply Lagrangian dynamics, inertia tensors, and dynamic models (Newton-Euler and Lagrange-Euler) for 2-link and 3-link robot manipulators, including trajectory planning and actuator dynamics.
2. To analyze control systems via state space: transfer functions, controllability, observability, state feedback design.
3. To analyze and design linear multivariable control systems using state space and transfer matrix.
4. To implement joint space control methods including computed torque, near minimum time control, feedforward control, and survey existing robot control algorithms.
5. To implement adaptive, resolved motion & inverse dynamics control; program robot trajectories.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: DERIVE and **COMPUTE** dynamic models of serial robot manipulators using Lagrange-Euler and Newton-Euler formulations, and plan joint interpolated trajectories.

CO2: DEVELOP state space representations of given transfer functions, and determine controllability and observability of LTI systems.

CO3: DESIGN state feedback and output feedback controllers using pole placement, and analyze MIMO control systems for stability, controllability, and observability.

CO4: APPLY computed torque control and near minimum time control for joint space motion control of robot manipulators.

CO5: IMPLEMENT advanced control strategies (adaptive, resolved motion, inverse dynamics) and program robot trajectories using industrial robot programming languages.

Course Contents

Unit I	Foundations of Robot Dynamics	(08 Hours)

Lagrangian Dynamics and Inertia Tensors: Introduction to Lagrange's equation, Derivation from Kinetic and potential energy, Link inertia tensor, Jacobian inertia tensor and their role in manipulator dynamics.

Dynamic Models, Trajectories, and Control: Newton-Euler (recursive) and Lagrange-Euler (closed-form) dynamic models, Dynamic modeling of 2-link and 3-link manipulators, Operational space dynamic model, Trajectory planning considerations, Joint interpolated trajectories and interpolation methods, Set point tracking, Introduction to actuator dynamics.

Real World Assignment:

1. Derive Lagrange-Euler dynamics (including inertia tensors) for a 2-link manipulator welding a car chassis, and compute joint torques for a given trajectory.
2. Formulate the recursive Newton-Euler and operational space dynamic models for a 3-link manipulator picking objects from a moving conveyor belt.
3. Design a joint-interpolated trajectory (linear with parabolic blends) and a set-point tracking controller with actuator dynamics for a 2-link manipulator performing a precision drilling operation.

Exemplary: Collaborative robots (cobots) with torque sensing, High-speed delta robots (pick-and-place in food/packaging, Legged robots (e.g., quadruped or biped), Heavy-payload pick-and-place robots, Surgical robotic arms (e.g., da Vinci system), Industrial welding robots.

Unit II	State Space Control Systems Analysis	(08 Hours)

State Space Representation and Properties: Introduction to state variables and state space representation, State-space representations of transfer-function systems, Controllability and observability (definitions, Kalman rank condition tests, importance in control design).

Controller Design in State Space: Design of controllers using root-locus in state space, Pole placement with state feedback (full-state feedback) and with output feedback, Introduction to robust control systems.

Real World Assignment:

1. Check controllability and observability of a two-link robotic manipulator using the Kalman rank condition, and explain why full-state feedback cannot be designed if the system is uncontrollable.
2. Design a pole placement controller for a differential drive mobile robot to regulate position and orientation, then implement output feedback using an observer when only wheel encoder positions are measurable.
3. For a SCARA robot arm, derive the state space model from transfer functions of each joint, verify controllability, and place closed-loop poles to achieve critically damped trajectory tracking under variable payload.
4. Design a state feedback controller using pole placement for a given transfer function, then validate controllability and observability via Kalman rank tests in MATLAB.

Exemplary: Robotic manipulator trajectory tracking, Mobile robot navigation, Quadrotor UAV stability control, Bipedal walking robot balance, Autonomous vehicle lane keeping, Collaborative robot (Cobot) force control.

Unit III	Multivariable Control Systems	(08 Hours)

MIMO Modeling, Analysis, and Properties: Modeling, analysis, and design of linear multi-input, Multi-output (MIMO) control systems, Both state space and transfer matrix (frequency domain) approaches, Stability analysis of MIMO LTI systems, Controllability, Sterilizability (stabilizability), observability in MIMO context.

Realization, Reduction, and Design: Realization theory and model order reduction for MIMO systems, Multivariable control system design methodologies.

Real World Assignment:

1. Derive a MIMO state-space model for a differential-drive robot (position + orientation as outputs, wheel speeds as inputs); check controllability and observability for trajectory tracking.
2. Obtain a minimal realization of a 3-DOF robot arm transfer matrix; reduce model order using balanced truncation while preserving end-effector dynamics.
3. For two UAVs in formation (4 inputs, 4 outputs), design a MIMO decentralized controller and analyze stability and stabilizability under communication delays.

Exemplary: Quadruped robot (e.g., Spot, ANYmal), Industrial robot arm (6-DOF), Soft robotic manipulator, Humanoid robot balancing, SLAM (Simultaneous Localization & Mapping), Drone (UAV) trajectory tracking, Robot digital twin.

Unit IV	Joint Space & Computed Torque Motion Control	(08 Hours)

Joint Space Control Methods: Manipulator control problem (challenges: nonlinearity, coupling, uncertainties), Joint space control, Computed torque techniques, Near minimum time control, Feedforward control.

Existing Control Algorithms for Robots: Survey of existing control algorithms used in controlling robots including PD control with gravity compensation, Inverse dynamics control, Nonlinear decoupled feedback control, Resolved motion control, Adaptive control.

Real World Assignment:

1. Computed Torque Control for a 2-DOF Manipulator – Simulate computed torque vs. PD with gravity compensation, comparing trajectory tracking under payload uncertainty.
2. Inverse Dynamics Control for a Pick-and-Place Arm – Implement inverse dynamics control on a 3-DOF robot, handling nonlinearity and coupling during high-speed motion.
3. Survey & Implementation of Existing Algorithms – Compare PD with gravity compensation, nonlinear decoupled feedback control, and adaptive control on a 6-DOF welding robot for accuracy and robustness.
4. Simulate and compare computed torque control vs. PD with gravity compensation for a 2-link manipulator in MATLAB/Simulink.

Exemplary: Industrial Assembly Robots (Computed Torque & Inverse Dynamics, Surgical Robots (PD Control with Gravity Compensation), Space Robotic Arms (Near Minimum Time Control), Humanoid Walking/Running (Nonlinear Decoupled Feedback Control), Autonomous Underwater Vehicles (Adaptive Control).

Unit V	Advanced Robot Control Strategies & Trajectory Programming	(08 Hours)

Nonlinear and Adaptive Control Strategies: PD control with gravity compensation, Full inverse dynamics (feedback linearization) control, Nonlinear decoupled feedback control, Resolved motion control in operational space, Adaptive control for robots with unknown or varying parameters.
Trajectory Programming and Integration: Robot control of trajectory using programming languages (e.g., VAL, RAPID, KRL), Comparison and integration of all motion control strategies.

Real World Assignment:

1. Implement nonlinear decoupled feedback control in simulation for a 6-DOF industrial robot performing high-speed pick-and-place of unknown payload masses, then compare its tracking error against standard PD control with gravity compensation.
2. Program a trajectory in KRL on a KUKA robot to weld a curved seam under varying friction conditions, and integrate an adaptive control law that updates inertia estimates online while comparing performance with resolved motion control in the operational space.

Exemplary: High-speed pick-and-place with unknown payloads, Robotic arc welding of irregular seams, Force-controlled assembly of precision parts.

Learning Resources

Text Books:

1. Craig, J. J. (2018). *Introduction to robotics: Mechanics and control (4th ed.).* Pearson.
2. Kurdila, A. J., & Ben-Tzvi, P. (2020). *Dynamics and control of robotic systems (1st ed.).* Wiley.
3. Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: Modelling, planning and control.* Springer Science & Business Media.
4. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2020). *Robot modeling and control (2nd ed.).* Wiley.

Reference Books:

1. Behera, L., Kumar, S., Patchaikani, P. K., Nair, R. R., & Dutta, S. (2020). *Intelligent control of robotic systems.* CRC Press.
2. *Billard, A., Figueroa, N., & Mirrazavi, S. (2022). Learning for adaptive and reactive robot control: A dynamical systems approach.* MIT Press.
3. Liu, S., & Chen, G. S. (2019). *Dynamics and control of robotic manipulators with contact and friction.* Wiley.
4. Shah, S. V., Saha, S. K., & Dutt, J. K. (2012). *Dynamics of tree-type robotic systems.* Springer Netherlands.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Robotics and Control: Theory and Practice
   https://onlinecourses.nptel.ac.in/noc26_me72/preview
2. NPTEL Course: Mechanics and Control of Robotic Manipulators
   https://onlinecourses.nptel.ac.in/noc25_me105/preview
3. Coursera Course: Robot Dynamics
   https://www.coursera.org/learn/modernrobotics-course3
4. Coursera Course: Modern Robotics, Course Motion Planning and Control
   https://www.coursera.org/learn/modernrobotics-course4

Savitribai Phule Pune University
Third Year of Robotics and Artificial Intelligence (2024 Pattern)

PCC-302-RAI: Machine Learning for Robotics

Teaching Scheme		Credit	Examination Scheme	
Theory	**3 Hours/Week**	**3**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:
Programming and Problem Solving (PCC-151-ITT),
Basics of Robotics & AI (PCC-201-RAI)

Course Objectives:

The students will be able

1. To introduce fundamentals of ML, learning paradigms, PAC framework, probability, and version spaces.
2. To provide knowledge of regression, Naïve Bayes, and decision trees for supervised learning.
3. To explain advanced classifiers including K-NN, logistic regression, perceptrons, neural networks, and SVM.
4. To explore unsupervised learning methods like clustering, EM algorithm, and PCA.
5. To describe evaluation metrics, ensemble learning, and the complete ML practice process.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: ANALYZE different machine learning paradigms and apply PAC learning concepts along with probability fundamentals to solve basic learning problems.

CO2: IMPLEMENT linear and multilinear regression models, Naïve Bayes classifiers, and decision trees (ID3, CART) for supervised learning tasks with appropriate error analysis.

CO3: DEVELOP and compare advanced classification models including K-NN, logistic regression, perceptrons, multi-layer neural networks, and support vector machines for both linear and nonlinear data.

CO4: APPLY unsupervised learning techniques such as K-Means clustering, hierarchical clustering, DBSCAN, Expectation Maximization, and PCA to discover hidden patterns and reduce data dimensionality.

CO5: EVALUATE models using metrics, implement ensemble methods, and execute end-to-end ML workflows.

Course Contents		
Unit I	**Foundations of Learning & Probability**	**(07 Hours)**

Introduction to Machine Learning and PAC Framework: Definition and scope of Machine Learning, Supervised learning, Unsupervised learning, Semi-supervised learning, Reinforcement learning, PAC (Probably Approximately Correct) Learning framework, Hypothesis space, Sample complexity, Generalization error vs. training error, Bias-variance tradeoff.

Robot Architecture for Learning: Random variables and probability distributions, Conditional probability and Bayes' theorem, Expectation, Variance and covariance, Maximum Likelihood Estimation (MLE) basics, Concept learning and hypothesis spaces, Find-S algorithm, Candidate Elimination algorithm, General-to-specific ordering and boundary sets.

Real World Assignment:

1. PAC Learning: Estimate sample complexity and compare training vs. generalization error for a robot arm learning a grasping threshold.
2. Concept Learning: Implement Find-S and Candidate Elimination to learn obstacle-free zones for a mobile robot from example trajectories.
3. Probabilistic Methods: Apply Bayes' theorem and MLE to fuse noisy sensor data (e.g., camera + LiDAR) and analyze bias-variance tradeoff in robot localization.
4. Compute expectation, variance, and covariance of robot wheel encoder readings to model uncertainty and predict pose estimation error in differential drive robots.

Exemplary: Robot arm grasping threshold learning (PAC learning), Mobile robot obstacle avoidance concept learning, Generalization error in vision-based robotic pick-and-place, Bias-variance tradeoff in robot navigation control.

Unit II	**Regression and Probabilistic Supervised Learning**	**(07 Hours)**

Linear and Multilinear Regression: Predicting continuous outputs using regression, Simple Linear Regression, MSE cost function, Closed-form solution, Multilinear Regression, Gradient Descent (Batch, Stochastic, Mini-batch), Normal equation method, Ridge (L2) regularization, Lasso (L1) regularization, Overfitting and underfitting.

Naïve Bayes and Decision Trees: Conditional independence assumption, Bayes' rule for classification, Gaussian, Multinomial and Bernoulli Naïve Bayes, Laplace smoothing, ID3 algorithm using Information Gain and Entropy, CART algorithm using Gini Index, Pre-pruning and post-pruning, Error bounds and generalization.

Real World Assignment:

1. Predict robot joint angle from sensor data using MSE, gradient descent (batch/SGD/mini-batch), and compare closed-form vs. normal equation.
2. Classify terrain (carpet/grass/concrete) for a mobile robot using Gaussian & Bernoulli Naïve Bayes with Laplace smoothing.
3. Build ID3 (entropy) & CART (Gini) trees for obstacle avoidance actions; apply pre/post-pruning to reduce overfitting.
4. Estimate wheel slip using Ridge (L2) & Lasso (L1); analyze overfitting vs. underfitting.

Exemplary: Robot joint angle prediction, Drone altitude control regression, Warehouse robot path branching, Obstacle avoidance decision tree, Wheel slip estimation, Battery discharge prediction, Robot arm trajectory error prediction.

Unit III	Advanced Classifiers	(07 Hours)

Instance-Based and Linear Classifiers: K-Nearest Neighbors (K-NN), Euclidean, Manhattan and Minkowski distances, Curse of dimensionality, Weighted K-NN, Logistic Regression, Sigmoid function, Cross-entropy loss, Softmax regression, Single-layer perceptron, Perceptron convergence, XOR problem limitation.

Multi-Layer Perceptrons and Support Vector Machines: Multi-Layer Perceptrons (MLP) and hidden layers, Backpropagation algorithm, ReLU, Tanh, Sigmoid and Leaky ReLU activations, Vanishing and exploding gradients, Linear SVM (Hard margin, Soft margin, Hinge loss), Nonlinear SVM using Kernel trick (Polynomial, RBF, Sigmoid kernels), Support vectors.

Real World Assignment:

1. Classify obstacle proximity using LIDAR data with Euclidean/Manhattan distances; analyze curse of dimensionality and weighted K-NN.
2. Implement sigmoid-based logistic regression and single-layer perceptron for line-following robot turns; demonstrate XOR limitation.
3. Implement MLP (backpropagation with ReLU/Tanh) and SVM (RBF kernel) for robotic hand gesture recognition; analyze vanishing gradients.

Exemplary: Obstacle proximity classification, Line-following turn decision, Hand gesture recognition for robotic arm, Object surface type detection, Human-robot collision prediction, Grasp success/failure classification, Multi-class object recognition.

Unit IV	Unsupervised Learning & Dimensionality Reduction	(07 Hours)

Clustering Fundamentals and K-Means: Partitional clustering (K-Means, K-Modes), Hierarchical clustering (Agglomerative, Divisive, Dendrograms), Density-based clustering (DBSCAN with eps and minPts), K-Means algorithm steps, Elbow method and Silhouette score, Self-Organizing Maps (SOM) for topological mapping.

Expectation Maximization and Principal Component Analysis: Soft vs. hard clustering, Gaussian Mixture Models (GMM), Expectation Maximization (EM) algorithm (E-step and M-step), Dimensionality reduction using PCA, Eigenvalues and eigenvectors, Principal components and explained variance, Reconstruction error, Applications of PCA.

Real World Assignment:

1. Implement K-Means (elbow method, silhouette score) and DBSCAN (eps, minPts) to segment robot-collected LIDAR points into room clusters; compare partitional vs. density-based clustering.
2. Implement Gaussian Mixture Models with Expectation Maximization (E-step, M-step) to track multiple moving objects from robot camera data; compare soft vs. hard clustering.
3. Apply PCA to reduce high-dimensional joint angle data from a robotic arm; compute eigenvalues, explained variance, and reconstruction error for visualization and faster processing.

Exemplary: LIDAR point cloud room segmentation, Multi-object tracking from camera feed, Robotic arm joint data compression, Topological mapping for warehouse navigation, Anomaly detection in robot sensor readings, Feature reduction for real-time object recognition.

Unit V	Model Evaluation, Ensemble Learning & ML Practice	(07 Hours)

Evaluation Metrics and Significance Tests: Confusion Matrix (TP, TN, FP, FN), Accuracy, Precision, Recall, F1-Score, ROC Curves and AUC, MSE, RMSE, R-squared, k-fold cross-validation, Learning curves.

Ensemble Learning and ML Process in Practice: Bagging and Random Forests (bootstrapping, Out-of-bag error), Boosting (Adaboost, XGBoost), Data preprocessing (handling missing values, Normalization, Encoding), Outlier analysis using Z-Score, Hyperparameter tuning (Grid Search, Random Search), Train-validation-test split, Visualization of results.

Real World Assignment:

1. Evaluation Metrics: Compute precision, recall, F1-score, ROC-AUC, and k-fold cross-validation for a robot obstacle detection model.
2. Ensemble Learning: Implement Random Forest and XGBoost for robot path prediction with data preprocessing (normalization, outlier removal) and hyperparameter tuning (Grid Search).

Exemplary: Obstacle detection model evaluation, Robot path prediction using Random Forest, Hyperparameter tuning for navigation controller, Cross-validation for grasping success prediction (k-fold), Learning curves for drone landing accuracy.

Learning Resources

Text Books:

1. Govers, F. X. (2024). *Artificial intelligence for robotics: Build intelligent robots using ROS 2, Python, OpenCV, and AI/ML techniques for real-world tasks (2nd Ed.).* Packt Publishing Ltd.
2. Roth, A., Manocha, D. N., Sriram, R. D., & Tabassi, E. (2024). *Explainable and interpretable reinforcement learning for robotics*. Springer.
3. Ko Jaulin, L. (2019). *Mobile robotics (2nd Ed.).* Wiley

Reference Books:

1. Imran, A., & Gopalakrishnan, K. (2025). *AI for robotics: Toward embodied and general intelligence in the physical world (1st Ed.).* Apress.
2. Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., & Thrun, S. (2005). *Principles of robot motion: Theory, algorithms, and implementations.* MIT Press.
3. Siciliano, B., & Khatib, O. (Eds.). (2016). *Springer handbook of robotics (2nd Ed.).* Springer International Publishing.
4. Roth, A., Manocha, D. N., Sriram, R. D., & Tabassi, E. (2024). *Explainable and interpretable reinforcement learning for robotics.* Springer.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Introduction to Machine Learning
   https://onlinecourses.nptel.ac.in/noc26_cs74/preview
2. NPTEL Course: Machine Learning
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc20_cs49
3. Coursera Course: Machine Learning Specialization
   https://www.coursera.org/specializations/machine-learning-introduction

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PCC-303-RAI: Robot Modeling and Simulation				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**4 Hours/Week**	**4**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:

Mathematics for Intelligent Systems (MDM-221-RAI),
Kinematics of Robot (PCC-252-RAI)

Course Objectives:

The students will be able

1. To model kinematic structures of robotic manipulators and mobile robots using mathematical representations such as transformation matrices and Jacobians.
2. To analyze differential kinematics, statics, and singularities to understand velocity and force relationships in robot manipulators
3. To formulate dynamic models of robotic systems using Lagrange methods and implement appropriate control architectures.
4. To design motion control and force control for various robotic tasks.
5. To apply visual servoing techniques and motion planning for both manipulators and mobile robots.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: DEVELOP forward and inverse kinematic models for typical robot manipulators using homogeneous transformations and rotation matrices.

CO2: COMPUTE Jacobian matrices, identify kinematic singularities, and solve inverse differential kinematics for velocity and static force analysis.

CO3: DERIVE dynamic equations of motion using Lagrange formulation and distinguish between direct and inverse dynamics.

CO4: IMPLEMENT motion control schemes (joint space and operational space) and force control strategies (impedance, compliance, hybrid) for robot manipulators.

CO5: SIMULATE visual servoing systems and motion planning algorithms (including probabilistic methods) for mobile robots and manipulators in a programming environment.

Course Contents

Unit I	Mathematical Modeling & Kinematics	(08 Hours)

Mathematical Modeling of Robots: Symbolic representation of manipulators, Configuration space, State space, Workspace, Classification of robots (serial, parallel, hybrid), Accuracy and repeatability, Resolution, Wrist and end effectors, Common kinematic arrangements (articulated, SCARA, delta), underactuated and mobile robots.

Kinematics of Rigid Bodies: Pose of a rigid body in 3D, Rotation matrices and composition, Euler angles and gimbal lock, Angle-axis representation, Homogeneous transformations, Direct kinematics for typical manipulators, Joint space vs. operational space, Kinematic calibration.

Real World Assignment:

1. For a 6-DOF articulated manipulator, compute the reachable workspace boundaries from joint limits and identify a singular pose where wrist resolution degrades.
2. Derive the direct kinematics of a SCARA robot using homogeneous transformations, then demonstrate gimbal lock in its Euler-angle wrist and resolve it using angle-axis representation
3. For a delta robot, mathematically model end-effector pose from active joints, then design a calibration routine to measure and distinguish accuracy errors from repeatability errors.

Exemplary: Articulated robot for welding in automotive manufacturing, SCARA robot for PCB assembly in electronics, Delta robot for high-speed food packaging, Mobile robot localization in warehouses (Mobile Robots & Pose), Hybrid robot for medical surgery (Hybrid Kinematics & Wrist), Kinematic calibration of an industrial welding robot.

Unit II	Differential Kinematics, Statics & Trajectory Planning	(08 Hours)

Differential Kinematics and Statics: Geometric Jacobian matrix, Jacobian for typical manipulators, Kinematic singularities, Redundancy analysis, Inverse differential kinematics, Analytical Jacobian, Inverse kinematics algorithms, Statics (force/torque mapping).

Trajectory Planning, Actuators & Sensors: Path vs. trajectory, Operational space and joint space trajectories, Joint actuating systems (electric, hydraulic, pneumatic), Proprioceptive sensors (encoders, tachometers), Exteroceptive sensors (force, tactile, proximity).

Real World Assignment:

1. Implement the geometric Jacobian of a 6-DOF articulated robot in MATLAB/Python. Simulate a trajectory passing through a wrist singularity and apply damped least-squares to avoid high joint velocities. Compare tracking error with and without singularity handling.
2. For a SCARA robot, use the Jacobian transpose to map desired end-effector forces (5 N insertion) to joint torques. Implement a hybrid force-position controller in simulation. Plot contact force vs. time during insertion.
3. Generate a joint space cubic spline trajectory for a delta robot pick-and-place cycle (0.4 sec). Respect joint velocity and torque limits of electric actuators. Specify encoder and proximity sensor feedback for real-time path correction.

Exemplary: Collaborative robot (Cobot) singularity avoidance, Heavy payload hydraulic excavator (Statics), Redundant 7-DOF robot for obstacle avoidance, Pneumatic pick-and-place with trajectory planning, Surgical robot force-sensitive scraping (Statics & Tactile).

Unit III	Dynamics & Control Architecture	(08 Hours)

Dynamics of Robot Manipulators: Lagrange formulation, Properties of dynamic models, Dynamic model of simple manipulators, Parameter identification, Direct and inverse dynamics, Dynamic scaling of trajectories, Operational space dynamic model, Dynamic manipulability ellipsoid.

Control Architecture: Functional architecture (task planning, control loops), Programming environment (simulation vs. real-time), Hardware architecture (microcontrollers, DSPs, industrial PCs).

Real World Assignment:

1. Design hardware (microcontroller for joint control, DSP for inverse dynamics, industrial PC for planning). Implement operational space dynamics for force control. Compare simulation vs. real-time execution.
2. For a delta robot pick-and-place cycle, compute dynamic manipulability ellipsoid along the path. Apply dynamic scaling to reduce peak torque by 25% while holding cycle time. Plot torque profiles before/after.
3. Derive dynamic model of a 2-DOF planar robot using Lagrange formulation. Implement inverse dynamics control in MATLAB to track a circular trajectory. Compare tracking error with/without compensation.

Exemplary: Computed torque control for a welding robot, Dynamic scaling for painter robot speed adjustment, Parameter identification for a collaborative robot payload, Direct dynamics simulation for surgical robot training, Dynamic manipulability for Humanoid Robot walking.

Unit IV	Motion Control & Force Control	(08 Hours)

Motion Control: Control problem (tracking desired trajectories), Joint space control, Independent joint PID control, Decentralized control with gravity compensation, Computed torque feedforward control, Centralized control using full dynamics, Operational space control.

Force Control: Need for force control in assembly and contact tasks, Compliance control (passive vs. active), Impedance control (regulating inertia, damping, and stiffness), Force control with direct feedback, Constrained motion with natural and artificial constraints, Hybrid force/motion control.

Real World Assignment:

1. On a 3-DOF planar robot, implement decentralized joint PID with gravity compensation feedforward. Compare position tracking error and settling time against pure PID during slow vs. fast trajectories.
2. Implement hybrid force/motion control on a SCARA robot inserting a peg. Use motion control in free directions and force control (10 N) in constrained directions. Demonstrate jam-free insertion.
3. Design an impedance controller (M, B, K) for a 6-DOF robot maintaining 5 N contact force on a curved surface. Tune parameters to minimize force ripple.
4. Implement and compare computed torque control vs. independent joint PID for trajectory tracking, and simulate hybrid force/motion control for a peg-in-hole assembly task in MATLAB/Simulink.

Exemplary: Independent Joint PID on SCARA for PCB assembly, Computed torque control on articulated robot for laser welding, Operational space control on Delta Robot for pick-and-place, Direct force feedback on polishing robot, Passive compliance on gripper for bottle capping, Simulation of robotic arm.

Unit V	Visual Servoing, Mobile Robots & Motion Planning	(08 Hours)

Introduction to Visual Servoing: PD control with gravity compensation, Full inverse dynamics (feedback linearization) control, Nonlinear decoupled feedback control, Resolved motion control in operational space, Adaptive control for robots with unknown or varying parameters.

Mobile Robots and Motion Planning: Nonholonomic constraints (no lateral motion), Kinematic models of wheeled robots (differential drive, car-like), Dynamic models, Motion control for trajectory tracking, Odometric localization, Configuration space representation, Planning via retraction, Cell decomposition, Probabilistic planning (RRT, PRM), Planning via artificial potentials.

Real World Assignment:

1. Implement eye-in-hand visual servoing on a 6-DOF robot to track a moving object. Use resolved motion control in operational space. Compare tracking error with/without visual feedback.
2. Model a differential drive robot with nonholonomic constraints. Generate a collision-free path using RRT in a cluttered map. Design a trajectory tracking controller with odometric localization. Validate in simulation.

Exemplary: Nonholonomic and differential drive – Warehouse robot, Shipboard robot, Adaptive control welding with unknown load.

Learning Resources

Text Books:

1. Corke, P. (2023). *Robotics: Fundamental algorithms in Python (1st Ed.).* Springer. Springer.
2. Craig, J. J. (2018). *Introduction to robotics: Mechanics and control (4th Ed.).* Pearson Education.
3. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2020). *Robot modeling and control (2nd Ed.).* Wiley.
4. Lynch, K. M., & Park, F. C. (2017). *Modern robotics: Mechanics, planning, and control.* Cambridge University Press.

Reference Books:

1. Siciliano, B., & Khatib, O. (Eds.). (2016). *Springer handbook of robotics (2nd Ed.).* Springer.
2. Noda, I., Ando, N., Brugali, D., & Kuffner, J. J. (Eds.). (2012). *Simulation, modeling, and programming for autonomous robots.* Springer.
3. Megahed, S. M. (1993). *Principles of robot modelling and simulation.* Wiley.
4. Khalil, W., & Dombre, E. (2004). *Modeling, identification and control of robots. Butterworth-Heinemann.* ScienceDirect.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Modelling and Simulation of Dynamic Systems
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_me18
2. NPTEL Course: Intelligent Control of Robotics Systems
   https://onlinecourses.nptel.ac.in/noc26_ee89/preview
3. NPTEL Course: Robotics: Basic and Selected Advanced Concepts
   https://onlinecourses.nptel.ac.in/noc23_me67/preview

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PCC-304-RAI: Advances in Robotics and Artificial Intelligence				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**3 Hours/Week**	**3**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:

Basics of Robotics & AI (PCC-201-RAI)

Course Objectives:

The students will be able

1. To introduce humanoid robot technologies (sensors, control, actuation, and system integration), social robots, and case studies of Atlas, ASIMO, and Pepper.
2. To provide knowledge of swarm robotics (basics, systems, tasks, real-world applications), smart robots, and warfare robotics.
3. To explain HRI fundamentals (scope, need, types, ethics, architecture), collaborative robots, and Cobot applications and technology.
4. To describe Industry 4.0 evolution, IoRT fundamentals, cloud robotics vs. IoRT, and applications across industrial, healthcare, agriculture, smart city, and consumer sectors.
5. To introduce NLP stages (preprocessing, lexical analysis, parsing, semantic analysis, NLG) and automated reasoning methods and types in AI.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: ANALYZE humanoid robot technologies, social robot applications, and case studies of Atlas, ASIMO, and Pepper.

CO2: EXPLAIN swarm basics, smart robots and their applications, and robotics for warfare.

CO3: DESCRIBE HRI fundamentals, collaborative robots, and Cobot applications and technology.

CO4: IDENTIFY Industry 4.0 fundamentals, IoRT, and its applications across industrial, healthcare, agriculture, smart city, and consumer sectors.

CO5: CLASSIFY NLP techniques and automated reasoning methods according to their functions and applications.

Course Contents		
Unit I	**Humanoid Robotics and Social Robots**	**(07 Hours)**

Humanoid Robot Technologies: Sensors in Humanoid Robot (vision, tactile, proprioceptive, proximity, auditory, force), Control of Humanoid Robot (open-loop, closed-loop, balance, gait, joint position, whole-body), Actuation types for Humanoid Robot (electric, hydraulic, pneumatic, series elastic, shape memory alloy, magnetic), System integration in Humanoid Robot (hardware, software, power, communication, sensor-actuator, real-time).

Social Robots: Social Robot (definition, characteristics, types, social interaction, emotional expression, and communication), Need of Social Robots, Assistive and Social Robots in healthcare and others (education, retail, hospitality, defense, agriculture).

Case Study on Humanoid Robot: Boston Dynamics Atlas, Honda ASIMO, SoftBank Pepper.

Real World Assignment:

1. Watch ASIMO/Atlas video; list sensors, control methods, actuation types, and integration features.
2. Compare sensor placement in NAO vs. Atlas using product datasheets; note three differences.
3. Choose one robot (Atlas, ASIMO, Pepper, Valkyrie, Tesla Bot, and NAO); prepare one-page case study on technologies, applications, and limitations.

Exemplary:

1. ASIMO's floor surface detection sensors detect floor type and irregularities to adapt walking gait for stable navigation.
2. Pepper robot's emotional recognition via cameras uses facial expression and posture analysis to identify human emotions for social interaction.
3. Atlas robot's IMU for balance measures acceleration and angular rate to maintain equilibrium during dynamic movements like running and jumping.

Unit II	**Swarm Robotics and Smart Robots**	**(07 Hours)**

Swarm Robotics: Characteristics, Swarm Robotics and multi-Robotic Systems, Experimental platforms in Swarm Robotics.

Swarm Tasks and Real-World Use: Tasks in Swarm Robotics, Swarm robots used in real world applications.

Smart and Warfare Robots: Smart Robots, Smart Robots applications, Robotics for warfare applications.

Real World Assignment:

1. Visit a nearby warehouse or watch a warehouse video; differentiate between swarm robotics and multi-robot system; justify your classification with one reason.
2. Research and recall 2 real-world applications of swarm robots in agriculture (e.g., Ecorobotix, SWARMIX); write them with brief explanations.
3. Suggest smart robot applications for an Indian airport (e.g., baggage handling, passenger guidance); propose at least 2 innovative use cases with reasoning.

Exemplary:

1. Ant-inspired mining swarms use pheromone-based communication for underground resource exploration.
2. PackBot for IED disposal provides remote reconnaissance and bomb defuse in combat zones.
3. Daksh bomb disposal robot (Indian Army) handles hazardous materials with 6-degree-of-freedom arm.

Unit III	Human-Robot Interaction (HRI) and Collaborative Robots	(07 Hours)

HRI Fundamentals: Introduction to HRI, Scope, Need, Types, HRI vs. HCI, Ethical issues in HRI, Multi-modal perception, HRI Architecture (perception, decision-making, action execution, feedback loop, ROS/NAOqi middleware).
Collaborative Robots (Cobots): Definition and characteristics of Cobots, Difference between Cobots and industrial robots, Safety standards, Types of collaboration.
Cobot Applications and Technology: Applications of collaborative robots (assembly lines, pick and place, quality inspection, packaging), Collaborative robot technology (torque and force sensors, vision-guided Cobots, end-effector design for safety, programming methods).

Real World Assignment:

1. Visit a nearby industry, identify any collaborative or industrial robot, and explain how collaborative robots differ from industrial robots in terms of safety standards.
2. Interview five workers in a small manufacturing unit or watch factory safety videos; explain how collaborative robots differ from industrial robots in safety standards.
3. Propose a new HRI architecture for a social robot at an Indian railway station (guiding, announcements, and queries); include perception, decision, action, feedback, and middleware (ROS/NAOqi).

Exemplary:

1. KUKA LBR iiwa (2010s) introduced collaborative interaction with joint torque sensors for safe human proximity.
2. Rethink Robotics Baxter uses visual cues (screen face) and compliant arms for intuitive human-cobot collaboration.
3. Techman cobot with built-in vision uses integrated camera for workpiece alignment without external sensors.

Unit IV	Industry 4.0 and Internet of Robotic Things (IoRT)	(07 Hours)

IoRT Fundamentals: Introduction to Industry 4.0, Cyber-Physical Systems (CPS), Industrial Internet of Things (IIoT), Evolution from Industry 1.0 to 4.0, Key enabling technologies (AI, Cloud, Big Data, 5G, Edge Computing), Definition of Internet of Robotic Things (IoRT), Integration of IoT with Robotics, Cloud robotics vs. IoRT.
Applications & Developments of IoRT: Industrial applications (predictive maintenance, smart manufacturing, warehouse automation), Healthcare and Agriculture applications, Smart cities and Consumer applications, Recent developments (5G-enabled IoRT, AI-driven robot fleets, digital twins).

Real World Assignment:

1. Draw a poster showing the evolution from Industry 1.0 to Industry 4.0. Display it in your classroom and explain how each era enabled the next.
2. Compare two warehouse automation systems (Amazon Kiva vs. Flipkart's robotic system). Contrast their robot navigation methods, cloud coordination, and scalability.
3. Compare two real-world IoRT platforms (e.g., AWS RoboMaker vs. Azure IoT with robots), Differentiate their features, cloud integration, and ease of use.

Exemplary:

1. Bosch Industry 4.0 factory (India) uses connected sensors and predictive maintenance for manufacturing efficiency.
2. Starship delivery robots fleet uses IoRT for real-time tracking, remote monitoring, and route optimization.
3. Azure IoT with autonomous mobile robots coordinates warehouse robots through cloud-based task allocation.

Unit V	Natural Language Processing (NLP) and Automated Reasoning	(07 Hours)

Natural Language Processing (NLP): Introduction, Classical approaches to Natural Language Processing, Text preprocessing, Lexical analysis, Syntactic parsing, Semantic analysis, Natural language generation, Applications.

Automated Reasoning: Introduction, Methods of reasoning, Reasoning types, Use of automated reasoning in AI, Reasoning and its types, Applications for Automated Reasoning, Mathematical consideration.

Real World Assignment:

1. Take any 5 customer reviews from Amazon or Flipkart. Manually perform sentiment analysis (positive, negative, neutral) and compare result with the product's actual rating.
2. Compare two NLP tools (Google Translate vs. Microsoft Translator) by translating the same 5 sentences; differentiate their accuracy, grammar handling, and language coverage.
3. Compare two applications of automated reasoning (e.g., hardware verification at Intel vs. software bug detection using Facebook Infer). Contrast their methods, scale, and real-world impact.

Exemplary:

1. ChatGPT response generation applies transformer-based natural language generation for conversational AI.
2. Google Translate integrates syntactic parsing and semantic transfer for cross-language translation.
3. Facebook Infer uses automated reasoning to detect null pointer exceptions and memory leaks.
4. GPT-4 for code generation combines semantic understanding and logical reasoning for programming tasks.

Learning Resources

Text Books:

1. Soloman, S. (2023). *Advanced robotics (design & applications)*. Khanna Book Publishing Co.
2. Trivedi, M. C. (2023). *A classical approach to artificial intelligence*. Khanna Book Publishing.
3. Luger, G. F. (2008). *Artificial intelligence (5th Ed.)*. Pearson.
4. Herbrick, R., & Graepel, T. (2010). *A handbook on natural language processing (2nd Ed.)*. CRC Press.

Reference Books:

1. Mukherjee, S. (2021). *Robotics process automation*. Khanna Book Publishing.
2. Chopra, R. (2023). *Data science with AI, ML, DL*. Khanna Book Publishing.
3. Dadios, E. P., Biliran, J. J. C., Garcia, R. G., Johnson, D., & Valencia, A. R. B. (2012). *Humanoid robot: Design and fuzzy logic control technique for its intelligent behaviors*. IntechOpen.
4. Navarro, I., & Matía, F. (2013). *An introduction to swarm robotics*. *ISRN Robotics*.
5. Faneuff, J., & Follett, J. (2016). *Designing for collaborative robotics*. O'Reilly Media.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Collaborative Robots
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc25_me86

2. NPTEL Course: Introduction To Industry 4.0 And Industrial Internet of Things
   https://nptel.ac.in/courses/106105195

3. NPTEL Course: Natural Language Processing
   https://nptel.ac.in/courses/106105158

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PCC-305-RAI: Design of Robot Elements				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**3 Hours/Week**	**3**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:

Materials and Manufacturing (PCC-202-RAI)
Mechanics of Materials (PCC-253-RAI)

Course Objectives:

The students will be able

1. To differentiate static and fluctuating loads in robots and justify fatigue failure theories for infinite life design.
2. To evaluate shafts for rigidity as per ASME code and select bearings for robot joints.
3. To design gears using beam strength, compute power screw efficiency, and specify timing belts for robotic drives.
4. To classify gripper types and construct jaw mechanisms with appropriate actuation force and materials.
5. To determine bolt preload, choose locking devices, and formulate spring rates for robot applications.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: ANALYZE static and fluctuating loads in robots and apply fatigue failure theories (Soderberg, Goodman, and Gerber) to design for infinite life.

CO2: EVALUATE shafts for torsional and lateral rigidity as per ASME code and select suitable bearings for robot joints based on load, speed, and alignment.

CO3: DESIGN gears using Lewis beam strength, compute torque and efficiency of power screws, and select timing belts for robotic drives.

CO4: COMPARE mechanical, vacuum, magnetic, and soft grippers and design jaw mechanisms with appropriate actuation force and materials.

CO5: DETERMINE bolt preload and locking devices for vibrating robots and calculate compression/torsion spring rates for gravity compensation.

Course Contents		
Unit I	**Design against Static and Fluctuating Loads**	**(07 Hours)**

Introduction to Loads: Static, Fluctuating (repeated, reversed, and random), and impact loads, Load cycles in robot operation.

Static Load Design: Factor of safety (FoS), Design for static strength, Stress Concentration: Causes, stress concentration factors, Failure theories applications.

Fatigue Load Design: Cyclic stress, Endurance limit, S-N curve, Fatigue failure stages (crack initiation, propagation), Fatigue strength modifying factors, Fatigue failure theories: Soderberg, Goodman, Gerber criteria, Combined mean and alternating stress, Design for infinite life.

Real World Assignment:

1. Conduct a load cycle analysis on a pick-and-place robot arm by identifying static, repeated, reversed, and random loads during a typical operation cycle.
2. Calculate the required factor of safety (FoS) for a critical robot link using static strength design and accounting for stress concentration factors from bolted joints or sharp corners.
3. Generate an S-N curve for a small aluminum or steel specimen using a rotating bending fatigue tester and visually identify crack initiation and propagation stages.

Exemplary: Static Load Design: Design robot gripper fingers, Calculate lifting arm FoS, Fluctuating Load Design: Monitor conveyor robot wrist torque, Classify pick-and-place load cycles, and assess repeated bending in linkage.

Unit II	**Design of Shaft and Bearings**	**(07 Hours)**

Design of Shaft: Introduction and Robot specific requirements**,** Shaft design on the strength basis, Torsional rigidity basis and lateral rigidity basis, Hollow shaft design, Design of shaft as per ASME code, Shaft connections for robots.

Design of Bearings: Introduction and Robot specific requirements, Classification of bearings for robotics, Bearing selection criteria: load, speed, temperature, and alignment, Static and dynamic load carrying capacities, Lubrication, maintenance, and failure prevention.

Real World Assignment:

1. Measure torsional deflection of a solid vs. hollow robot drive shaft under rated torque to verify lateral and torsional rigidity requirements.
2. Select suitable rolling element bearings for a robot elbow joint based on load, speed, temperature, and alignment data from a duty cycle.
3. Perform lubrication inspection and failure analysis on worn robot arm bearings, documenting common failure modes.

Exemplary: Shaft Design: Robot wrist joint shaft, Mobile robot wheel axle, and Industrial robot elbow shaft, Shaft Connections for Robots: Keyed joint in robotic gripper, Pin connection in articulated arm, Bearing Design: Deep groove ball bearing in SCARA robot, Angular contact bearing in robot wrist, Thrust bearing in vertical lift arm.

Unit III	Design of Power Transmission Elements	(07 Hours)

Introduction & Terminology: Introduction to Gears in Robotics, Gear Terminology & Geometry.
Gear Design: Beam strength of gear tooth, Lewis form factor, Allowable stress, Design for safe bending stress. Numerical on finding module, Harmonic drive (concept only).
Power Screws Design: Types of screw threads, Lead screw vs. ball screw, Torque analysis with square & trapezoidal threads, Efficiency, Self-locking condition, Stresses in power screws, Torque to raise/lower load, Robot applications (prismatic joints, SCARA Z-axis).
Timing Belts: Timing belts in robots (GT2, GT3, HTD), Geometrical relations (pitch length, center distance), Analysis of belt tensions, Selection of timing belts (pitch, width, material).

Real World Assignment:

1. Compute Lewis form factor and design module for a robot wrist gear tooth using beam strength with allowable bending stress.
2. Measure torque to raise and lower a SCARA robot Z-axis lead screw, calculate efficiency, and verify self-locking condition.
3. Compare ball screw vs. trapezoidal lead screw in a prismatic joint of a linear robot for friction and stiffness.

Exemplary: Gear Design in Robotics: Robot joint harmonic drive, Spur gear in gripper mechanism, Helical gear in mobile robot wheel, Power Screws in Robotics: SCARA robot Z-axis ball screw, Lead screw in 3D printer robot, Timing Belts in Robotics: Delta robot arm drive (GT2 belt), SCARA robot secondary joint (HTD belt), 3D printer-based pick robot (GT3 belt).

Unit IV	Design of Robot End Effectors	(07 Hours)

Introduction & Types: Introduction to end effectors, Types: grippers vs. tools, Considerations for gripper selection: part geometry, environment, cycle time, grip type.
Design of Mechanical Grippers: Force analysis, Mechanism: Parallel jaw gripper (rack & pinion, double pivoting, toggle linkage), Jaw stroke, Actuation force (pneumatic or electric), Material for jaws (hardened steel, aluminum with rubber pad).
Other Types of Grippers: Vacuum grippers, Magnetic grippers, Soft/ compliant grippers.
Tools as End Effector & Robot Interface: Tool types with applications, Mechanical interface: ISO 9409-1, Physical support of the end effector.

Real World Assignment:

1. Select an appropriate gripper for three different parts (sphere, cube, thin sheet) based on geometry, cycle time, and environment.
2. Design and test jaw materials (hardened steel vs. aluminum with rubber pad) for gripping fragile vs. heavy objects.
3. Compare vacuum gripper, magnetic gripper, and soft compliant gripper on porous, ferrous, and delicate surfaces.

Exemplary: Gripper selection: Soft gripper for apple picking, Parallel jaw for bin picking, Vacuum for sheet metal, Gripper comparison: Vacuum for cardboard box, Magnetic for steel sheet, Soft gripper for tomato, Vacuum for silicon wafer, Magnetic for scrap metal, Soft gripper for eggshell.

Unit V	Design of Fasteners and Springs	(07 Hours)

Fasteners for Robots: Introduction to Threaded Joints, Bolt types (hex head, socket cap), Bolt grades (8.8, 10.9, and 12.9), Preload, and Torque tightening formula.
Locking devices: Need for locking in vibrating robots, Spring washer, Nord-Lock wedge washer, Loctite thread locker (blue 242, red 271), Nyloc nut, Safety wire, and Selection guide.
Springs in robots: Need for springs (gravity compensation, compliance, and return mechanism), Compression spring, Torsion spring, Spring rate selection for gravity counterbalance.

Real World Assignment:

1. Measure compression spring rate of a gravity compensation spring from an articulated robot and calculate required spring rate for counterbalancing.
2. Design a torsion spring for a gripper return mechanism and validate compliance under cyclic operation.
3. Prepare a selection guide for locking devices based on robot type (collaborative arm, delta robot, heavy payload robot, cleanroom robot).

Exemplary: Bolt grade selection for FANUC base mount, Compression spring for ABB IRB 120 counterbalance, Torsion spring for apple picking gripper, Nyloc nut for delta robot arm.

Learning Resources

Text Books:

1. Bhandari, V.B. (2007). *Design of Machine Elements (2nd Ed.)*. Tata McGraw Hill Publication Co. Ltd.
2. Pandya and Shah (2015). *Machine Design (20th Ed.)*. Charotar Publishing.
3. Shigley, J.E. and Mischke, C.R. (2009). *Mechanical Engineering Design*. McGraw Hill Publication Co. Ltd.
4. Juvinall, R.C. and Marshek, K.M. (2011). Fundamentals of Machine Component Design. Wiley India.
5. Sclater, N. (2011). *Mechanisms and Mechanical Devices Sourcebook*. McGraw Hill.

Reference Books:

1. Norton, R.L. (2001). *Machine Design*. Pearson Education Asia.
2. Dieter, G.E. and Schmidt, L.C. (2013). *Engineering Design*. McGraw Hill Education, Indian Edition.
3. Pilli, S.C. and Patil, H.G. (2017). *Design of Machined Elements*. I. K. International Publisher.
4. Hall, Holowenko, and Laughlin (2008). *Machine Design* (Schaum's Outline series), adapted by S.K. Somani. Tata McGraw Hill Publishing Company Ltd., New Delhi, Special Indian Edition.
5. Craig, J.J. (2005). *Introduction to Robotics: Mechanics and Control*. Pearson Education.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Design of Mechanical Transmission Systems
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc24_me71
2. NPTEL Course: Design of Machine Elements - I
   http://www.digimat.in/nptel/courses/video/112105124/L01.html
4. NPTEL Course: Gear And Gear Unit Design: Theory and Practice
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_me62
5. Coursera Course: Machine Design Part I
   https://www.coursera.org/learn/machine-design1

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)					
PCC-306-RAI: Robot Programming Lab					
Teaching Scheme		**Credits**	**Examination Scheme**		
Theory	**NA**	**1**	**Term work**	**NA**	
Practical	**2 Hours/Week**		**Practical**	**50 Marks**	

Prerequisite:
Fundamentals of Programming Languages (ESC-105-COM),
Kinematics of Robot (PCC-252-RAI)

Course Objectives:

The student will be able

1. To derive Lagrange-Euler dynamics for a 2-link planar manipulator and implement recursive Newton-Euler algorithm for inverse dynamics of a 3-link manipulator.
2. To design state-space, pole placement, computed torque, and PD and gravity controllers.
3. To perform kinematic analysis (forward kinematics, Jacobian, singularities) for 6-DOF robots.
4. To implement inverse dynamics and hybrid force/motion control.
5. To apply visual servoing and RRT motion planning.

Course Outcomes:

After successful completion of the course, learner will be able to:

CO1: COMPUTE joint torques for given trajectories using Lagrange-Euler formulation and recursive Newton-Euler algorithm.

CO2: DESIGN full-state feedback controllers using pole placement after verifying controllability and observability from state-space models.

CO3: EVALUATE trajectory tracking performance of computed torque control and PD with gravity compensation under payload uncertainty through simulation.

CO4: PERFORM forward kinematics, workspace analysis, and singularity identification, and implement damped least-squares to avoid kinematic singularities.

CO5: DEVELOP and simulate inverse dynamics control, hybrid force/motion control for constrained assembly, and eye-in-hand visual servoing with RRT motion planning.

Laboratory Experiments/Assignments	
The student must complete the following activity for their Term Work Journal	
Part A	**Dynamics and Control of Robotic Systems**
Practical 1	**Lagrangian Dynamics & Joint Torque Computation** Derive Lagrange-Euler dynamics for a 2-link planar manipulator and compute joint torques for a given trajectory.
Practical 2	**Newton-Euler Recursive Dynamics for 3-Link Manipulator** Implement recursive Newton-Euler algorithm for inverse dynamics.
Practical 3	**State Space Representation & Controllability/Observability** Convert transfer functions to state space and test Kalman rank conditions.
Practical 4	**Pole Placement Controller Design for a Robotic Joint** Design full-state feedback controller using pole placement.
Practical 5	**Computed Torque Control vs. PD with Gravity Compensation** Simulate and compare trajectory tracking performance under payload uncertainty.
Part B	**Robot Modeling and Simulation**
Practical 1	**Direct Kinematics & Workspace Analysis of a 6-DOF Articulated Robot** Compute forward kinematics and identify singular poses.
Practical 2	**Geometric Jacobian & Singularity Avoidance** Compute Jacobian and implement damped least-squares to avoid singularities.
Practical 3	**Lagrange Dynamics & Inverse Dynamics Control** Derive dynamic model and implement inverse dynamics control.
Practical 4	**Hybrid Force/Motion Control for Peg-in-Hole Assembly** Implement hybrid force/motion control for constrained assembly task.
Practical 5	**Visual Servoing & RRT Motion Planning for Mobile Robot** Implement eye-in-hand visual servoing and RRT path planning.

Learning Resources

Text Books:

1. Kurdila, A. J., & Ben-Tzvi, P. (2020). *Dynamics and control of robotic systems (1st Ed.)*. Wiley.
2. Rico, F. M. (2022). *A concise introduction to robot programming with ROS 2*. Chapman & Hall/CRC.
3. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2020). *Robot modeling and control (2nd Ed.)*. Wiley.
4. Corke, P. (2023). *Robotics: Fundamental algorithms in Python (1st Ed.)*. Springer.

Reference Books:

1. Liu, S., & Chen, G. S. (2019). *Dynamics and control of robotic manipulators with contact and friction.* Wiley.
2. Megahed, S. M. (1993). *Principles of robot modelling and simulation.* Wiley.
3. Khalil, W., & Dombre, E. (2004). *Modeling, identification and control of robots*. Butterworth-Heinemann. ScienceDirect.
4. Noda, I., Ando, N., Brugali, D., & Kuffner, J. J. (Eds.). (2012). *Simulation, modeling, and programming for autonomous robots.* Springer.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Robotics and Control: Theory and Practice
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc20_me03

2. NPTEL Course: Robotics: Basics and Selected Advanced Concepts
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_me01

3. Coursera Course: Modern Robotics, Course Motion Planning and Control
   https://www.coursera.org/learn/modernrobotics-course4

Real World Assignments:

1. Lagrangian Dynamics & Joint Torque Computation: Derive the Lagrange-Euler dynamics for a 2-link planar manipulator and compute required joint torques for a minimum-jerk trajectory from (0°, 0°) to (60°, 45°).
2. Newton-Euler Recursive Dynamics for 3-Link Manipulator: Implement the recursive Newton-Euler inverse dynamics algorithm for a 3-link planar manipulator to compute joint torques under a given end-effector load.
3. State Space Representation & Controllability/Observability: Convert a DC motor-driven robotic joint transfer function to state space form and test controllability and observability using Kalman's rank condition.
4. Pole Placement Controller Design for a Robotic Joint: Design a full-state feedback controller using Ackermann's formula to place poles for a single robotic joint with given inertia and damping.
5. Computed Torque Control vs. PD with Gravity Compensation: Simulate trajectory tracking of a 2-link planar arm under 30% payload uncertainty comparing computed torque control and PD plus gravity compensation.

6. Robot Modelling and Simulation - Direct Kinematics & Workspace Analysis: Compute forward kinematics for a 6-DOF articulated robot (e.g., PUMA 560) and identify singular poses within its workspace.
7. Geometric Jacobian & Singularity Avoidance: Compute the geometric Jacobian for a 2-link planar arm and implement damped least-squares (DLS) to avoid singularities during trajectory execution.
8. Lagrange Dynamics & Inverse Dynamics Control: Derive the full Lagrangian dynamic model of a 2-link manipulator and implement inverse dynamics control for tracking a circular trajectory.
9. Hybrid Force/Motion Control for Peg-in-Hole Assembly: Implement a hybrid force/motion controller for a 3-DOF planar robot performing a peg-in-hole assembly task with environmental contact constraints.
10. Visual Servoing & RRT Motion Planning for Mobile Robot: Implement eye-in-hand visual servoing combined with RRT path planning for a differential drive mobile robot to navigate to and grasp a target object.

Exemplary:

Autonomous vehicle lane keeping and obstacle avoidance, Space robot manipulator capture of floating objects, Surgical robot teleoperation and tremor filtering, Warehouse mobile robot navigation and pallet picking, Drone trajectory tracking and payload stabilization.

Savitribai Phule Pune University
Third Year of Robotics and Artificial Intelligence (2024 Pattern)

MDM-321-RAI: Intelligent Systems and Applications

Teaching Scheme		Credits	Examination Scheme	
Theory	**NA**	**1**	**CCE**	**NA**
Practical	**2 Hours/Week**		**Practical**	**50 Marks**

Prerequisite:
Programming and Problem Solving (PCC-151-ITT)

Course Objectives:

The student will be able

1. To implement classical sense-plan-act architecture and model predictive control for autonomous robotics.
2. To apply regression and classification techniques for robot dynamics modeling and sensor data interpretation.
3. To design MLPs for inverse kinematics and CNNs for vision-based robotic tasks.
4. To implement decision-making algorithms using MDPs, DQN, and policy gradient methods.
5. To integrate imitation learning with control systems for complex automation tasks.

Course Outcomes:

After successful completion of the course, learner will be able to:

CO1: ANALYZE and compare supervised learning models for robotic perception and control problems.

CO2: DEVELOP neural network architectures for kinematic approximation and visual recognition tasks.

CO3: IMPLEMENT reinforcement learning algorithms for sequential decision-making in dynamic environments.

CO4: EVALUATE classical and learning-based control strategies for mobile robots and manipulators.

CO5: DESIGN integrated robotic systems combining sensing, learning, and planning components.

Laboratory Experiments/Assignments	
The student must complete the following activity for their Term Work Journal (Any 10)	
Practical 1	Implement the classical Sense-Plan-Act (SPA) architecture for a simple task.
Practical 2	Model the relationship between a robot arm's desired and actual position using regression
Practical 3	Apply Ridge and Lasso regression to predict a robot's future trajectory and analyze feature selection.
Practical 4	Use logistic regression to classify objects based on simple geometric features.
Practical 5	Train an SVM to classify LIDAR scan data as "obstacle" or "free space".
Practical 6	Train a simple MLP to approximate the inverse kinematics of a 2-link planar arm.
Practical 7	Implement a simple CNN to classify images from a robotics-relevant dataset.
Practical 8	Implement a dynamic programming algorithm to solve a simple Markov Decision Process (MDP).
Practical 9	Train a DQN agent to balance a pole on a cart using the OpenAI Gym environment.
Practical 10	Implement a simple policy gradient algorithm to learn a policy for a continuous control problem.
Practical 11	Implement a simple model predictive controller for a simulated mobile robot.
Practical 12	Train a neural network to mimic an expert policy for a simple driving task.

Learning Resources

Text Books:

1. Govers, F. X. (2024). *Artificial intelligence for robotics: Build intelligent robots using ROS 2, Python, OpenCV, and AI/ML techniques for real-world tasks (2nd Ed.).* Packt Publishing Ltd.
2. Roth, A., Manocha, D. N., Sriram, R. D., & Tabassi, E. (2024). *Explainable and interpretable reinforcement learning for robotics.* Springer.
3. Ko Jaulin, L. (2019). *Mobile robotics (2nd Ed.).* Wiley.

Reference Books:

1. Imran, A., & Gopalakrishnan, K. (2025). AI for robotics: *Toward embodied and general intelligence in the physical world (1st Ed.).* Apress.
2. Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., & Thrun, S. (2005). *Principles of robot motion: Theory, algorithms, and implementations.* MIT Press.
3. Siciliano, B., & Khatib, O. (2016). *Springer handbook of robotics (2nd Ed.).* Springer International Publishing.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Introduction to Machine Learning
   https://onlinecourses.nptel.ac.in/e-learning/preview/
2. NPTEL Course: Machine Learning
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc20_cs49
3. Coursera Course: Machine Learning Specialization
   https://www.coursera.org/specializations/machine-learning-introduction

Real World Assignment:

1. Implement Sense-Plan-Act for a simulated vacuum robot that senses dirt, plans cleaning path, and acts by moving and cleaning.
2. Model the relationship between commanded and actual end-effector positions of a robotic arm using linear regression to improve accuracy.
3. Apply Ridge and Lasso regression to predict a drone's future flight path from noisy sensor data and analyze which features (velocity, acceleration) matter most.
4. Use logistic regression to classify incoming objects as "box" or "cylinder" based on height-to-width ratio and other geometric features.
5. Train an SVM on LIDAR point cloud data to classify regions as obstacle or free space for safe navigation.
6. Train an MLP to compute joint angles required to reach a given (x, y) coordinate for a pick-and-place task.
7. Implement a CNN to classify images as "graspable" or "non-graspable" for a robotic arm in a cluttered environment.

8. Solve an MDP using dynamic programming to find optimal routes for an autonomous forklift in a warehouse grid.
9. Train a DQN agent to balance a pole on a cart, simulating a self-balancing robot or inverted pendulum system.
10. Implement REINFORCE policy gradient to learn smooth continuous joint movements for a robotic arm reaching task.
11. Implement Model Predictive Control for a simulated self-driving car to navigate through a track with obstacles.

Exemplary: Industrial robotic arm calibration, Autonomous vehicle obstacle detection, Pick-and-place robot, Warehouse sorting robot, Self-balancing scooter, Autonomous Lane following.

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
MDM-322-RAI: Autonomous Navigation using SLAM				
Teaching Scheme		**Credits**	**Examination Scheme**	
Theory	**NA**	**1**	**CCE**	**NA**
Practical	**2 Hours/Week**		**Practical**	**25 Marks**

Prerequisite:

Fundamentals of Programming Languages (ESC-105-COM),
Programming and Problem Solving (PCC-151-ITT)

Course Objectives:

The student will be able

1. To implement EKF SLAM and Fast SLAM 1.0 for 2D robot localization and mapping.
2. To build occupancy grids using log-odds and align LIDAR scans using ICP and NDT.
3. To construct and optimize pose graphs using Gauss-Newton or Levenberg-Marquardt.
4. To estimate camera motion via feature matching (ORB/SIFT) and build sparse 3D feature maps.
5. To localize robots using UWB, IMU, and multi-resolution scan matching (Hector SLAM style).

Course Outcomes:

After successful completion of the course, learner will be able to:

CO1: IMPLEMENT and compare EKF SLAM and Fast SLAM 1.0, analyzing the trade-offs between computational complexity, data association, and map consistency in 2D environments.

CO2: BUILD and update occupancy grid maps using log-odds formulation from ground-truth poses, and estimate odometry by sequentially aligning LIDAR point clouds using ICP and NDT.

CO3: OPTIMIZE robot trajectories by constructing pose graphs with loop closure constraints and applying nonlinear least squares optimization (Gauss-Newton / Levenberg-Marquardt).

CO4: ESTIMATE camera motion from feature correspondences and generate sparse 3D feature maps using ORB features tracked across RGB-D or stereo frames.

CO5: INTEGRATE range and inertial data (UWB and IMU) for indoor localization, and implement scan-to-map matching with multi-resolution map pyramids using gradient descent (Hector SLAM style) without ROS dependencies.

Laboratory Experiments/Assignments	
The student must complete the following activity for their Term Work Journal (Any 8).	
Practical 1	Implement Extended Kalman Filter SLAM in a 2D environment.
Practical 2	Implement Fast SLAM 1.0 using a set of particles, each carrying its own map.
Practical 3	Given ground-truth robot poses and simulated 2D LIDAR scans, build an occupancy grid map using log-odds update rules.
Practical 4	Align two 2D point clouds using ICP. Apply it sequentially to LIDAR frames to estimate odometry (Scan-to-Scan matching).
Practical 5	Build a pose graph from odometry and loop closure constraints, then optimize it using Gauss-Newton or Levenberg-Marquardt.
Practical 6	Estimate camera motion between frames using feature matching (ORB/SIFT) and the Essential/Fundamental matrix.
Practical 7	Implement NDT as an alternative to ICP for aligning 2D LIDAR scans.
Practical 8	Replicate the core of Hector SLAM scan to map matching using a multi-resolution map pyramid and gradient descent without any ROS dependencies.
Practical 9	Use Ultra-Wideband (UWB) range measurements and IMU data to localize a robot in an indoor environment.
Practical 10	Build a sparse 3D feature map using ORB features tracked across frames from an RGB-D camera or stereo pair.

Learning Resources

Text Books:

1. Hughes, C., & Hughes, T. (2016). *Robot programming: A guide to controlling autonomous robots*. Que Publishing.
2. Quigley, M., Gerkey, B., & Smart, W. D. (2015). *Programming robots with ROS: A practical introduction to the Robot Operating System*. O'Reilly Media, Inc.
3. Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: Modelling, planning and control*. Springer Publishing Company.

Reference Books:

1. Quigley, M., Gerkey, B., & Smart, W. D. (2016). *Programming robots with ROS*. SPD Shroff Publishers and Distributors Pvt. Ltd.
2. Joseph, L. (2013). *Mastering ROS for robotics programming: Design, build and simulate complex robots using ROS*. Packt Publishing.
3. Joseph, L. (2019). *ROS robotics projects: Build and control robots powered by the Robot Operating System, machine learning, and virtual reality*. Packt Publishing.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Wheeled Mobile Robots
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc21_me44

2. Coursera Course: Robot Manipulation and Wheeled Mobile Robots
   https://www.coursera.org/learn/modernrobotics-course5?utm_source=chatgpt.com

Real World Assignment:

1. Implement EKF with prediction from odometry and update from simulated LiDAR landmarks to estimate robot pose and map.
2. Code a particle filter where each particle maintains its own feature map using EKFs for landmark positions
3. Given ground-truth poses and LiDAR data, compute log-odds for each grid cell and threshold to build a binary map.
4. Write point-to-point ICP to align consecutive 2D LiDAR scans and compute odometry from scratch.
5. Build a pose graph with odometry constraints and loop closures, then optimize using Gauss-Newton or Levenberg-Marquardt.
6. Extract ORB features from two images, match them, compute Essential matrix, and recover relative camera pose.
7. Implement Normal Distributions Transform by gridding points into Gaussians and optimize alignment via Newton's method.
8. Perform scan-to-map matching using a multi-resolution map pyramid with gradient descent to estimate pose.
9. Integrate UWB ranges (from anchors) and IMU accelerometer/gyro in an EKF for indoor robot localization.

10. Track ORB features across RGB-D frames, triangulate 3D points using depth, and build a sparse feature map.
Exemplary: Autonomous warehouse forklifts, Quadruped robots, Last-mile delivery robots, Legged rescue robots, Grocery inventory robots, Visual SLAM drones, Underwater AUVs, Hospital service robots.

Savitribai Phule Pune University, Pune

Maharashtra, India

TE - Robotics and Artificial Intelligence (2024 Pattern)

Semester VI Courses

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PCC-351-RAI: Deep Learning and Neural Networks				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**4 Hours/Week**	**4**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:

Machine Learning for Robotics (PCC-302-RAI)

Course Objectives:

The students will be able

1. To understand fundamental neuron models, perceptron learning, and the need for multilayer perceptrons.
2. To derive the backpropagation algorithm mathematically and compare various gradient descent optimization methods (SGD, Momentum, Adam) with their advantages.
3. To study bias-variance tradeoff and apply regularization techniques for stable training.
4. To explore autoencoders for unsupervised feature learning and representation.
5. To understand advanced architectures including CNNs, RNNs, LSTMs, and attention mechanisms.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: IMPLEMENT and analyze fundamental neuron models (McCulloch–Pitts, perceptron) and explain the necessity of multilayer perceptrons and non-linear activation functions using the XOR problem.

CO2: APPLY the backpropagation algorithm for feedforward networks, and compare different gradient descent variants (SGD, Momentum, Adam) to mitigate optimization challenges like saddle points.

CO3: DIFFERENTIATE between underfitting and overfitting using bias-variance analysis, and justify the selection of appropriate regularization to enhance training stability.

CO4: DESIGN and train various autoencoder architectures (denoising, sparse, contractive) to perform representation learning, dimensionality reduction, and feature extraction from unlabeled data.

CO5: EVALUATE advanced deep learning models, including CNNs for image recognition tasks and RNNs/LSTMs with attention mechanisms for sequence-to-sequence problems.

Course Contents		
Unit I	**Foundations of Neural Networks Analytics**	**(08 Hours)**

Fundamental Neuron Models: Biological neuron vs. Computational units, McCulloch–Pitts unit and thresholding logic, Linear perceptron and perceptron learning algorithm (Architecture, weight update rule, and learning process).

Multilayer Perceptrons and Convergence: Linear separability and convergence theorem for perceptron learning algorithm (Geometric interpretation and XOR problem), Introduction to multilayer perceptrons (MLPs) (Need for hidden layers and forward pass), Representation power of MLPs (Universal approximation theorem), Sigmoid neurons (Non-linearity, Mathematical formulation, and Comparison with threshold units).

Real World Assignment:

1. Design a McCulloch-Pitts logic gate network for a line-following robot's turn-left/turn-right decisions.
2. Train a single-layer perceptron on a two-wheeled robot's sensor data and prove its failure on the XOR problem using geometric interpretation
3. Build a multilayer perceptron (one hidden layer) for a robotic arm to solve the XOR-based gripper open/close coordination task.
4. Use sigmoid neurons in an MLP to learn inverse kinematics of a 2-DOF robot arm from joint angle–end effector position pairs.

Exemplary: Threshold logic for collision detection in swarm robotics, MLP for end-effector trajectory prediction, Sigmoid neuron for smooth inverse kinematics of a robotic arm, Perceptron learning algorithm for wall-following behavior in mobile robots.

Unit II	**Training and Backpropagation**	**(08 Hours)**

Feedforward and Backpropagation: Feedforward neural networks representation (input, hidden, and output layers), Backpropagation algorithm (mathematical derivation using chain rule, error calculation, and step-by-step working mechanism).

Gradient Descent Variants and Challenges: Gradient descent (batch GD), Stochastic gradient descent (SGD), Momentum based GD, Nesterov accelerated GD, AdaGrad, RMSProp, Adam (brief working and advantages of each), Saddle point problem in neural networks (Difference from local minima and mitigation strategies).

Real World Assignment:

1. Implement feedforward backpropagation for a 3-layer neural network to predict a mobile robot's wheel velocities from sensor inputs
2. Apply momentum-based GD to smooth a robotic gripper's force closure learning and reduce oscillations.
3. Analyze saddle point issues in a 2-link robotic arm's inverse kinematics network and apply gradient noise or dropout as mitigation.
4. Implement Adam optimizer for a differential drive robot's end-to-end steering angle prediction from camera images.

Exemplary: Backpropagation for robot arm inverse kinematics, Batch GD for offline path planning, Momentum GD for smoother biped walking, Momentum GD for smoother biped walking, SGD for real-time drone stabilization, Saddle point avoidance in robot localization networks.

Unit III	**Regularization and Training Stability**	**(08 Hours)**

Bias-Variance and Basic Regularization: Bias-variance tradeoff (underfitting vs. Overfitting), L2 regularization (weight decay), Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods (bagging and boosting).

Advanced Regularization and Training Enhancements: Dropout and drop connect, Batch normalization, Better activation functions (ReLU, Leaky ReLU, and ELU), Better weight initialization methods (Xavier and He initialization), Greedy layer wise pre-training.

Real World Assignment:

1. Apply L2 regularization to reduce overfitting in a robot arm's inverse kinematics model trained on limited joint angle data.
2. Implement early stopping to prevent overfitting in a mobile robot's LiDAR-based obstacle avoidance neural network.
3. Implement batch normalization to stabilize training of a self-balancing robot's tilt angle prediction network.

Exemplary: L2 for arm inverse kinematics, Early stopping for obstacle avoidance, Data augmentation for gait classification, Dropout for gripper control, Batch norm for self-balancing robot, Xavier init for vision navigation, Leaky ReLU for drone control, Ensemble bagging for swarm collision.

Unit IV	**Autoencoders**	**(08 Hours)**

Basic Autoencoders and Regularization: Autoencoders (definition and basic architecture: Encoder, Bottleneck, Decoder), Regularization in autoencoders, Denoising autoencoders.

Specialized Autoencoders and Training Overview: Sparse autoencoders, Contractive autoencoders, Overview of autoencoder training: Problem definition of reconstruction, Data encoding and decoding, Interpretation of latent representations, Decision-making for feature learning.

Real World Assignment:

1. Implement a basic autoencoder to compress and reconstruct a robot arm's joint angle data for dimensionality reduction.
2. Apply a sparse autoencoder to learn sparse features from a quadruped robot's foot pressure sensors for gait analysis
3. Implement a contractive autoencoder to make a drone's visual odometry network robust to small input perturbations.
4. Use a denoising autoencoder to remove sensor noise from a mobile robot's LiDAR scans for cleaner obstacle mapping.

Exemplary: Denoising autoencoder for LiDAR noise removal, Sparse autoencoder for foot pressure features, Contractive autoencoder for drone odometry, Basic autoencoder for joint compression, Reconstruction error for motor anomaly detection.

Unit V	Advanced Architectures	(08 Hours)

Convolutional Neural Networks: Introduction to CNN and building blocks of CNN (convolution, pooling, and fully connected layers), Overview of CNN architectures (LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, ResNet), Visualizing CNNs and guided backpropagation, Fooling convolutional neural networks, Transfer learning.

Recurrent Neural Networks and Attention: Introduction to RNN and backpropagation through time (BPTT), Vanishing and exploding gradients and truncated BPTT, Long short term memory (LSTM) and gated recurrent units (GRU), Bidirectional RNNs and bidirectional LSTMs, Encoder-decoder models and attention mechanism.

Real World Assignment:

1. Implement a CNN (LeNet) on a mobile robot to classify floor surface types from camera images for adaptive traction control.
2. Implement a bidirectional LSTM on a robotic gripper to capture past and future force sensor context for smoother grasping.
3. Implement an RNN with BPTT to predict a quadruped robot's next footstep position from past joint angle sequences.

Exemplary: Transfer learning with VGGNet for robotic arm object recognition, Guided backpropagation for drone CNN adversarial visualization, RNN with BPTT for quadruped footstep prediction, LSTM for self-balancing robot tilt stabilization.

Learning Resources

Text Books:

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
2. Aggarwal, C. C. (2023). *Neural networks and deep learning: A textbook* (2nd Ed.). Springer.
3. Bishop, C. M., & Bishop, H. (2024). *Deep learning: Foundations and concepts*. Springer.
4. Haykin, S. (2009). *Neural networks and learning machines* (3rd Ed.). Pearson Prentice Hall.

Reference Books:

1. Chollet, F. (2021). *Deep learning with Python* (2nd Ed.). Manning Publications.
2. Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (3rd Ed.). O'Reilly Media.
3. Raschka, S. (2023). *Machine learning with PyTorch and Scikit-Learn* (2nd Ed.). Packt Publishing.
4. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). *Dive into deep learning*. Cambridge University Press.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Deep Learning
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc20_cs62
2. NPTEL Course: Introduction to Artificial Neural Networks
   https://nptel.ac.in/courses/117105084
3. Coursera Course: Neural Networks and Deep Learning
   https://www.coursera.org/learn/neural-networks-deep-learning
4. Coursera Course: Deep Learning Specialization
   https://www.coursera.org/specializations/deep-learning

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PCC-352-RAI: Robotic Process Automation				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**3 Hours/Week**	**3**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:

Computer Programming: JAVA Lab (MDM-222-RAI),
Advances in Robotics and Artificial Intelligence (PCC-304-RAI)

Course Objectives:

The students will be able

1. To explain insights on Robotic Process Automation (RPA) technology.
2. To illustrate basic bot development concepts including activities, sequences, flowcharts, variables, control flow, loops, and decision making using an RPA platform.
3. To describe data manipulation techniques and UI control automation.
4. To explain handling of user events, assistant bots, exception handling strategies, and logging mechanisms for debugging.
5. To design and manage scalable, secure, and maintainable RPA code while analyzing contemporary trends such as AI integration, governance, and hyperautomation.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: ANALYZE Robotic Process Automation technology, its value proposition, types of bots, application areas, and popular RPA tools.

CO2: DEMONSTRATE bot development using sequences, flowcharts, variables, control flow, loops, and decision-making constructs in an RPA platform like UiPath.

CO3: DIFFERENTIATE between various data manipulation techniques (data tables, file operations, CSV/Excel transfer) and UI control automation methods (recording, scraping, mouse, keyboard activities).

CO4: DESIGN automation solutions that handle user events through assistant bots, manage exceptions using try-catch and retry mechanisms, and implement logging with screenshots.

CO5: EVALUATE effective code organization, workflow structure selection, and contemporary RPA trends including security, governance, attended/unattended automation, and AI integration.

Course Contents

Unit I	Introduction to RPA & Bot Development Basics	(07 Hours)

Foundations of RPA: Emergence and definition of Robotic Process Automation (RPA), Benefits, Types of Bots (attended, unattended, hybrid), Application areas of RPA (finance, healthcare, HR, customer service), RPA development methodology and key considerations (process selection, security, governance), Overview of RPA tools (UiPath, Blue Prism, Automation Anywhere, Power Automate).

Bot Development Fundamentals: Activities, Flowcharts and Sequences, Sequencing the workflow, Log Message (information, warning, error), Variables, Control flow, Various types of loops (For Each, While, Do While), Decision making with conditions, Best practices for Bot Development, Step-by-step example using Sequence and Flowchart, Step-by-step example using Sequence and Control flow.

Real World Assignment:

1. Develop a sequence where a robotic arm inspects parts on a conveyor belt using variables, decision-making (If-Else), and Log Messages (Info, Warning, Error) to record pass/fail status and increment a counter.
2. Create a flowchart where a mobile robot picks multiple boxes using a For Each loop, implements retry logic with a While loop for failed grasps, and makes decisions to skip blocked locations.

Exemplary: Automated quality inspection collaborative robots (Cobots), Pick-and-Place operations articulated robotic arms, Emergency stop & safety monitoring collaborative robots (Cobots), Machine tending industrial robotic arms.

Unit II	Data Manipulation & Control Interaction	(07 Hours)

Data Handling Techniques: Data table usage in RPA (build, add row, filter, sort, loop through rows), Clipboard management (copy to clipboard, get clipboard text), File operations (copy, move, delete, read, write text file), Data transfer between CSV/Excel and data table (read CSV, write CSV, read range, write range, append range).

UI Control Automation: Finding the control (selectors, UI explorer), Techniques for waiting for a control (wait for ready, element exists, attribute equals), Act on controls – Mouse and keyboard activities (click, type into, hover, hotkeys), Handling events (click trigger, key trigger), Recording and scraping (basic recording, desktop recording, web recording, data scraping).

Real World Assignment:

1. Read robot sensor logs from Excel into DataTable, filter errors (temp > 75°C or error code $\neq 0$), sort by timestamp, write to CSV, archive original file.
2. Automate robot control panel using selectors, wait for controls ready, click start, type coordinates, hover over e-stop, use hotkeys (Ctrl+S), handle mode selection triggers.
3. Scrape robot part numbers and stock levels from a warehouse management screen using desktop recording and data scraping, copy scraped data to clipboard, paste into Excel using append range, and save the file.
4. Read robot maintenance logs from a text file, append timestamp and robot ID to each line, write to a new CSV file, copy the CSV to a backup folder, and delete the original text file after successful backup.

Exemplary: Robot log data processing, Robot control panel automation, Clipboard-based fault reporting, Inventory data scraping, CSV to robot configuration transfer.

Unit III	Handling User Events & Assistant Bots	(07 Hours)

Assistant Bots and Event Monitoring: Assistant bots (concept, use cases, attended automation scenarios), Monitoring system event triggers (system idle, process started, window opened), Monitoring image and element triggers (image appears, element disappears, attribute changes).

Keyboard Event Automation: Launching an assistant bot on a keyboard event (global hotkeys, shortcut keys), Practical examples (opening bot on Ctrl+Shift+B, triggering automation on function keys), Managing multiple keyboard triggers without conflicts.

Real World Assignment:

1. Develop an assistant bot that monitors a system event (global hotkey Ctrl+Shift+E) to trigger an emergency stop procedure on a collaborative robot, logging the event and displaying a warning notification.
2. Create an attended automation bot that monitors an image trigger (robot gripper holding a part appears on camera feed) and automatically saves the robot's current position and part ID to a log file.
3. Build an assistant bot that monitors system idle time (no user input for 5 minutes) and automatically triggers a safe shutdown sequence for a robotic arm, including moving to home position and disabling motors.

Exemplary: Robot part detection assistant, Robot control panel window bot, Robot status element disappears trigger, Robot temperature attribute monitor, Function key automation for robotics, Emergency stop hotkey for robotics.

Unit IV	Exception Handling & Logging	(07 Hours)

Exception Handling Strategies: Common exceptions in RPA (selector not found, timeout, application not responding, argument null), Strategies for exception handling (try-catch, throw, retry scope, continue on error), Using finally block for cleanup activities.

Logging and Debugging Mechanisms: Logging mechanisms – Client logging (local log files, debug output), Server logging (Orchestrator logs, cloud logging), Taking screenshots for error debugging (capture screenshot on exception, save to log), Best practices for logging (log levels, message formatting, avoiding sensitive data).

Real World Assignment:

1. Handle "selector not found" exception when a robot control panel button (e.g., "Start") fails to appear. Use try-catch, implement retry scope, capture screenshot on exception, and save to local log file.
2. Handle "timeout exception" when waiting for a robotic arm to reach target position within 10 seconds. Use throw to re-throw critical errors, finally block to reset robot to home position, and log error to cloud/Orchestrator logs.
3. Handle "application not responding" exception when robot controller freezes. Use continue on error for non-critical steps, finally block to close robot process gracefully, capture screenshot, and write to client logs with Info, Warn, and Error levels while avoiding sensitive data.

Exemplary: Automated welding robot, Mobile inventory robot, Surgical robot, Logistics robot, Painting robot, Packaging robot arm, Logistics robot, Inspection robot, Palletizing robot, Cleaning robot.

Unit V	Code Management & Contemporary Issues	(07 Hours)

Code Organization and Maintenance: Project organization best practices (folders for workflows, libraries, and data files), Choosing between Flowcharts, State Machines, or Sequences, Maintaining and scaling RPA code (modular design, reusable libraries, version control).

Current Trends and Future Directions: Security, Governance and compliance in RPA (role-based access, audit trails), Attended vs. unattended automation (differences and use cases), AI integration with RPA (Intelligent Process Automation, OCR), Future trends in industrial process automation using RPA (hyperautomation, cloud RPA).

Real World Assignment:

1. Implement role-based access for robot control (operator, supervisor, admin), maintain audit trails, and demonstrate attended vs. unattended robot automation.
2. Integrate OCR to read part labels from images, automate robot task scheduling, and explore hyperautomation and cloud RPA for remote robot monitoring.
3. Organize robot automation project with folders for workflows, libraries, and data files. Use modular design and reusable libraries for common robot functions (move, pick, place, etc.).

Exemplary: Modular robot control library, State machine for robot workflow, Version control for robot code, Audit trails for robot actions, Cloud RPA for remote robot monitoring.

Learning Resources

Text Books:

1. Srivastava, V. (2021). *Getting started with RPA using Automation Anywhere: Automate your day-to-day business processes using Automation Anywhere.* BPB Publications.
2. Tripathi, A. M. (2018). *Learning robotic process automation: Create software robots and automate business processes with the leading RPA tool – UiPath.* Packt Publishing.
3. Dasgupta, D. (2022). *Intelligent automation simplified: Learn enterprise automation, AI-led automation, and robotic process automation with use-cases (1st Ed.)*. BPB Publications.

Reference Books:

1. Lacity, M., & Willcocks, L. (2018). *Robotic process and cognitive automation: The next phase.* Steve Brookes Publishing.
2. Surdak, C. (2020). *The care and feeding of bots: An owner's manual for robotic process automation*. Independently Published.
3. Taulli, T. (2020). *The robotic process automation handbook: A guide to implementing RPA systems (1st Ed.).* Apress.
4. Willcocks, L., Hindle, J., & Lacity, M. (2019). *Becoming strategic with robotic process automation*. SB Publishing.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Industrial Robotics: Theories for Implementation
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc24_me117
2. Coursera Course: Robotic Process Automation (RPA) Specialization
   https://www.coursera.org/specializations/roboticprocessautomation
3. Coursera Course: Making the Case for Robotic Process Automation
   https://www.coursera.org/learn/robotic-process-automation

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PCC-353-RAI: Robot Operating System				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**4 Hours/Week**	**4**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:

Robot Modeling and Simulation (PCC-303-RAI),
Robot Programming Lab (PCC-306-RAI)

Course Objectives:

The students will be able

1. To explain Robot Operating System (ROS) fundamentals, evolution, Master/Node architecture.
2. To describe asynchronous topics, synchronous services, long-running actions, and parameter management for node communication.
3. To create, debug, and execute nodes with publishers, subscribers, service servers/clients, timers, and exception handling in both languages.
4. To comprehend mapping, localization, path planning, obstacle avoidance, and sensor-based object detection using ROS frameworks.
5. To use simulation, visualization, robot description, transforms, and interface real sensors/actuators via serial protocols and micro-ROS.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: DESCRIBE ROS architecture (Master, Nodes, Parameter Server), history, pros/cons, and real-world applications.

CO2: EXPLAIN ROS communication (topics, services, actions, parameters), naming conventions, and rosbag logging.

CO3: APPLY programming to develop ROS nodes in Python/C++ with publishers, subscribers, services, timers, and error handling.

CO4: ANALYZE navigation (SLAM, AMCL, path planning) and perception (detection, segmentation) using ROS frameworks.

CO5: CREATE ROS solutions using Gazebo, rviz, URDF, TF2, and hardware interfacing.

Course Contents

Unit I	Introduction to ROS & Environment Setup	(08 Hours)

ROS Overview: Introduction to robot software development, History and evolution of ROS, Comparison of ROS with other robotics frameworks (Including Webots and CoppeliaSim**)**, Overview of ROS architecture (Master, Nodes, Parameter Server, ROS Core), Applications of ROS in industry and research.

Environment Setup: Setting up a ROS development environment on Linux (Ubuntu), installing and configuring ROS distributions, creating and managing ROS workspaces (catkin and colcon), introduction to ROS packages and package management (package.xml, CMakeLists.txt), environment variable setup, basic ROS command-line tools (roscd, rosls, rospack, rosrun, rosnode, rostopic, rosservice, rosparam).

Real World Assignment:

1. Simulate a mobile robot in Gazebo using ROS, control its movement via keyboard teleoperation using rostopic pub, and visualize its real-time position and orientation in Rviz.
2. Write a ROS node that subscribes to laser scan data, implements obstacle avoidance, and publishes velocity commands to navigate a simulated robot through a cluttered environment.

Exemplary: Autonomous Mobile Robots (AMRs): ROS-based navigation stacks (move_base, SLAM) used in warehouse robots (e.g., Amazon, Fetch Robotics), Industrial Robot Arms: ROS-Industrial framework for controlling ABB, Fanuc, and Universal Robots in pick-and-place, welding, and assembly lines, Service Robots, Self-Driving Cars, Medical Robotics.

Unit II	ROS Fundamentals & Communication	(08 Hours)

Core Concepts: Nodes, Topics, Messages and services in ROS, ROS Communication: Publishers and subscribers (asynchronous), ROS services (synchronous request-reply), ROS actions (long-running tasks), ROS parameters and parameter server, ROS master and roscore functionality.

Messages and Data Definition: ROS standard message types, Data types and custom message definition with ROS messages (.msg), Service definition files (.srv), Action definition files (. action), introduction to ROS names and naming conventions, Remapping of topics and nodes, ROS bags for data logging and playback (Advanced rosbag analysis workflows – e.g., filtering, time-syncing, extracting data for ML).

Real World Assignment:

1. Write a ROS publisher node that simulates a LiDAR sensor publishing distance readings to /scan topic, and a subscriber node that stops the robot (publishes zero velocity) when an obstacle is detected within 0.5 meters.
2. Create a ROS service that takes a target joint angle as a request and returns the current joint position as a response, simulating a robotic arm joint controller for precise positioning.
3. Implement a ROS action server for a delivery robot that navigates to a goal warehouse location (long-running), provides real-time feedback (distance remaining), and allows preemption for urgent new orders.

Exemplary: Teleoperated robot control, Robot arm pose service, Obstacle avoidance, Long-duration Navigation, Parameter tuning, Data logging for debugging.

Unit III	Programming with ROS	(08 Hours)

Language and Node Creation: Introduction to ROS with Python (rospy) and C++ (roscpp), Creating simple ROS nodes in both languages, Node initialization and shutdown, Rate control and sleep functions, Logging and debugging output.

Communication Implementation: Publishing and subscribing to topics (Publisher, Subscriber, callback functions), Sending and receiving ROS service requests (ServiceClient and ServiceServer), Working with ROS libraries and APIs, Time handling in ROS (ros::Time, rospy. Time), Timers and periodic callbacks, Exception handling in ROS nodes.

Real World Assignment:

1. Write a Python (rospy) subscriber node for a mobile robot that monitors /battery status topic and stops the robot (/cmd_vel) with a log warning if voltage drops below 20%.
2. Write a C++ (roscpp) service server for a robotic arm that receives a target joint angle, simulates motion using a ros::Timer, returns time taken, and handles out-of-range exceptions.

Exemplary: Mobile Robot Emergency Stop, Robotic Arm Joint Service, Robot Odometry Logger, Battery Safety Node, Pick-and-Place Client, Sensor Fault Handler.

Unit IV	Robot Navigation & Perception	(08 Hours)

Robot Navigation: Robot navigation challenges and solutions (SLAM, path planning, localization, obstacle avoidance), ROS navigation frameworks (ROS Navigation Stack, MoveIt, Cartographer), Overview of costmaps, Odometry and sensor fusion, AMCL for localization, ROS enabled autonomous robot movement.

Robot Perception: Importance of robot perception for navigation and interaction, ROS integration with sensors like LiDAR, cameras, IMUs, and radar, Robot perception (object detection, segmentation, classification), How robots "see" and interpret their environment, Introduction to computer vision libraries in ROS (OpenCV, PCL), Camera calibration.

Real World Assignment:

1. Configure the ROS Navigation Stack on a simulated mobile robot, set up costmaps (local and global), and use AMCL for localization to autonomously navigate from a start pose to a goal pose while avoiding static obstacles.
2. Configure the ROS Navigation Stack on a simulated mobile robot, set up costmaps (local and global), and use AMCL for localization to autonomously navigate from a start pose to a goal pose while avoiding static obstacles.
3. Integrate a USB camera with ROS using usb_cam driver, perform camera calibration using camera_ calibration package, and write a node that uses OpenCV to detect and classify colored objects (e.g., red balls) for a pick-and-place robot.

Exemplary: Warehouse autonomous mobile robots, Self-driving cars, Service robots (tiago/care-o-bot), Agricultural drones, Robotic vacuum cleaners, Autonomous forklifts, Search & rescue robots.

Unit V	Advanced Tools, Simulation & Hardware Interfacing	(08 Hours)

Tools and Simulation: Introduction to ROS tools (rviz, rqt, rqt_graph, rqt_console, rqt_plot, rosbag, Foxglove Studio) for visualization, Debugging and logging, Working with robot simulations in ROS (Gazebo, comparison with Webots and CoppeliaSim), URDF for robot description, TF transform library (tf2) for coordinate frames, Introduction to robot navigation frameworks (MoveIt) and perception pipelines.

Hardware Interfacing: Introduction to robot sensors (proximity, light, contact) and actuators (DC motors, servo motors, stepper motors), Interfacing sensors and actuators with ROS drivers (UART, I2C, SPI, USB), Reading sensor data and controlling actuators through ROS nodes, Introduction to micro-ROS and rosserial for embedded systems, NVIDIA Jetson and TensorRT for edge inference, CI/CD basics using GitHub Actions / GitLab CI for robotics projects, Docker + ROS for reproducible setups, ROS 2 basics.

Real World Assignment:

1. Create a URDF model of a differential drive mobile robot, visualize it in RViz, simulate it in Gazebo with realistic physics, and use tf2 to publish coordinate transforms between robot base and laser scanner.
2. Interface an ultrasonic distance sensor (HC-SR04) with an Arduino, send sensor readings to ROS using rosserial, and write a ROS node that subscribes to this data to stop a DC motor (actuator) when an obstacle is detected within 20 cm.

Exemplary: Warehouse robot simulation, Surgical robot visualization, Low-cost educational robots, Industrial robotic arm programming, Agricultural drone mapping.

Learning Resources

Text Books:

1. Hughes, C., & Hughes, T. (2016). *Robot programming: A guide to controlling autonomous robots*. Que Publishing.
2. Quigley, M., Gerkey, B., & Smart, W. D. (2015). *Programming robots with ROS: A practical introduction to the Robot Operating System*. O'Reilly Media, Inc.
3. Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: Modelling, planning and control*. Springer Publishing Company.

Reference Books:

1. Martinez, A., & Fernandez, E. (2013). *Learning ROS for robotic programming*. Packt Publishing.
2. O'Kane, J. M. (2014). *A gentle introduction to ROS*. CreateSpace Independent Publishing Platform.
3. Joseph, L. (2013). *Mastering ROS for robotics programming: Design, build and simulate complex robots using ROS*. Packt Publishing.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Robotics and Control: Theory and Practice
   https://onlinecourses.nptel.ac.in/noc26_me72/preview

2. NPTEL Course: Robotics
   https://onlinecourses.nptel.ac.in/noc21_me76/preview

3. NPTEL Course: Introduction to Robotics
   https://nptel.ac.in/courses/107106090

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)					
PCC-354-RAI: Robot Operating and Process Automation Lab					
Teaching Scheme		**Credits**	**Examination Scheme**		
Theory	**NA**	**2**	**CCE**	**NA**	
Practical	**4 Hours/Week**		**Practical**	**50 Marks**	

Prerequisite:

Autonomous Navigation using SLAM (MDM-322-RAI)

Course Objectives:

The student will be able

1. To build ROS-based robotic systems using publishers, subscribers, services, and actions.
2. To simulate, localize, and navigate robots using ROS Navigation Stack, AMCL, and Gazebo.
3. To integrate USB cameras, calibration, OpenCV object detection, and TF2 transforms.
4. To learn RPA fundamentals and develop bots using UiPath (sequences, control flow, selectors).
5. To build advanced RPA bots with exception handling, OCR, modular design, and event monitoring.

Course Outcomes:

After successful completion of the course, learner will be able to:

CO1: IMPLEMENT publisher-subscriber for obstacle avoidance and service-client for parameter management in ROS.

CO2: EXECUTE long-duration navigation tasks using ROS action servers and robust nodes in Python/C++.

CO3: CONFIGURE ROS Navigation Stack with costmaps/AMCL and simulate URDF models in Gazebo.

CO4: DEVELOP RPA bots using DataTables, CSV/Excel, file operations, and control loops.

CO5: APPLY selectors, mouse/keyboard automation, exception handling, logging, and OCR in RPA.

Laboratory Experiments/Assignments	
The student must complete the following activity for their Term Work Journal	
Part A	**Robot Operating System**
Practical 1	Install ROS, create a ROS workspace, and practice basic command-line tools for robot simulation and teleoperation.
Practical 2	Implement asynchronous topic communication using ROS publishers and subscribers for obstacle avoidance in a simulated robot.
Practical 3	Implement synchronous request-reply communication using ROS services and manage robot parameters using the parameter server.
Practical 4	Implement a preemptable ROS action server and client for long-duration navigation tasks with real-time feedback.
Practical 5	Develop, debug, and execute ROS nodes in both Python (rospy) and C++ (roscpp) with timers and exception handling.
Practical 6	Configure the ROS Navigation Stack with costmaps and AMCL for autonomous mobile robot localization and path planning.
Practical 7	Integrate a USB camera with ROS, perform camera calibration, and implement object detection using OpenCV.
Practical 8	Create a URDF robot model, simulate it in Gazebo with TF2 transforms, and interface real sensors/actuators using rosserial.
Part B	**Robotic Process Automation**
Practical 1	Understand RPA fundamentals and develop basic bots using sequences, variables, control flow, loops, and decision-making constructs in an RPA platform (e.g., UiPath).
Practical 2	Implement data manipulation techniques including Data Table operations (build, filter, sort) and data transfer between CSV/Excel files for robot log processing.
Practical 3	Perform file operations (copy, move, delete, read/write text files) and clipboard management for automated data backup and reporting.
Practical 4	Automate robot control panels using selectors, mouse activities (click, hover), keyboard activities (type, hotkeys), and event triggers.
Practical 5	Use recording techniques (desktop, web) and data scraping to extract inventory or sensor data from legacy applications and transfer to structured formats.
Practical 6	Develop assistant bots that monitor system events (hotkeys, idle time, window triggers) and image/element triggers for attended automation scenarios.
Practical 7	Implement exception handling strategies (try-catch, retry scope, throw, finally) and logging mechanisms with screenshot capture for debugging robot failures.
Practical 8	Organize RPA projects using modular design and reusable libraries, implement role-based access and audit trails, and integrate OCR for intelligent process automation.

Learning Resources

Text Books:

1. Kumar, B. (2018). *Robot operating system cookbook: Over 70 recipes to help you master advanced ROS concepts (1st Ed.).* Packt Publishing.
2. *Govers, F. X. (2024). Artificial intelligence for robotics: Build intelligent robots using ROS 2, Python, OpenCV, and AI/ML techniques for real-world tasks (2nd Ed.). Packt Publishing.*
3. Quigley, M., Gerkey, B., & Smart, W. D. (2015). *Programming robots with ROS: A practical introduction to the Robot Operating System*. O'Reilly Media.

Reference Books:

1. Renard, E. (2024). *ROS 2 from scratch: Get started with ROS 2 and create robotics applications with Python and C++*. Packt Publishing.
2. Joseph, L., & Cacace, J. (2025). *Mastering ROS 2 for robotics programming: Design, build, simulate, and prototype complex robots using the Robot Operating System 2 (4th Ed.).* Packt Publishing.
3. Koubaa, A. (2021). *Robot Operating System (ROS): The complete reference* (*Volume 6*).Springer Nature.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Wheeled Mobile Robots
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc21_me44
2. NPTEL Course: Industrial Robotics: Theories for Implementation
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc24_me117
3. Coursera Course: Industrial Robotics: Robotic Process Automation (RPA) Specialization
   https://www.coursera.org/specializations/roboticprocessautomation

Real World Assignment:

1. Install ROS (Noetic/Humble), create a workspace, launch a simulated robot (TurtleBot3/Gazebo), and teleoperate it using keyboard or joystick commands
2. Implement a publisher-subscriber system where LiDAR scan data triggers velocity commands, enabling a simulated robot to avoid obstacles autonomously.
3. Create a ROS service to change robot speed, acceleration, and PID values at runtime, storing parameters on the parameter server for persistent tuning.
4. Build a preemptable action server that navigates a robot through multiple waypoints, providing real-time distance/feedback to the client.

5. Develop identical obstacle-detection nodes in Python (rospy) and C++ (roscpp) using timers, implementing try-catch for sensor read failures.
6. Configure ROS Navigation Stack, costmaps, and AMCL to enable a robot to localize and plan paths from point A to point B in a known map.
7. Integrate a USB camera into ROS, perform camera calibration, and use OpenCV to detect colored balls or faces in real-time.
8. Create a URDF model, simulate it in Gazebo with TF2 transforms, and interface an Arduino with ultrasonic sensor using rosserial.
9. Develop a bot in UiPath that logs into a robot dashboard, checks system status, and sends an email alert if and error is detected.
10. Read robot log CSV files, filter rows with "ERROR" status, sort by timestamp, and export cleaned data to a new Excel file.
11. Create a bot that copies daily robot logs to a backup folder, renames files, reads error codes from clipboard, and writes them to a summary text file.
12. Automate a legacy robot control panel using selectors, mouse clicks, keyboard shortcuts, and event triggers to change operating modes.
13. Use desktop/web recording and data scraping to extract sensor readings from a legacy web dashboard and transfer them to a structured Excel sheet.
14. Implement try-catch, retry scope, and finally blocks; capture screenshots on failure and write detailed error logs for debugging bot failures.
15. Organize bot code into reusable libraries, implement role-based access, maintain audit logs, and integrate OCR to read serial numbers from robot labels.

Exemplary: Warehouse Autonomous Mobile Robot (AMR), Hospital delivery robot, agricultural field robot, Security patrol robot, Autonomous vacuum cleaner, restaurant serving robot, Industrial robotic arm, Autonomous underwater vehicle.

Savitribai Phule Pune University
Third Year of Robotics and Artificial Intelligence (2024 Pattern)

PEC-361A-RAI: Augmented Reality and Virtual Reality

Teaching Scheme		Credit	Examination Scheme	
Theory	**3 Hours/Week**	**3**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite: NA

Course Objectives:

The students will be able

1. To understand the definition, history, working principle (5 stages), and core ingredients of augmented reality, and classify applications on Milgram's Reality-Virtuality Continuum.
2. To learn AR display types, processor architectures, tracking technologies, sensors, calibration, and sensor fusion for real-time performance.
3. To explain marker-based and marker-less tracking methods, including pose estimation, feature-based tracking, hybrid tracking, and recovery techniques.
4. To describe VR fundamentals, system components (visual, aural, haptic displays), human perception, vestibular system, and causes of vection and motion mismatch.
5. To understand geometric transformations (axis-angle, chaining) for tracking position/orientation in VR, and the impact of human eye movements on user experience.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: CLASSIFY real-world AR applications on Milgram's Reality-Virtuality Continuum and explain the 5 working stages and 5 ingredients of AR.

CO2: DIFFERENTIATE between various AR displays, processors, and tracking technologies including calibration, mobile sensors, optical tracking, and sensor fusion.

CO3: DEMONSTRATE marker-based tracking using printed markers and **COMPARE** it with marker-less tracking methods such as feature-based and hybrid tracking.

CO4: EXPLAIN VR fundamentals, system components (visual, aural, haptic displays), human perception, vestibular system, mismatched motion, and vection.

CO5: APPLY geometric transformations (axis-angle, viewing transformations, chaining) to track position, orientation, and attached bodies in VR.

Course Contents

Unit I	Introduction to Augmented Reality	(07 Hours)

Definition & History: Definition of augmented reality, History of augmented reality (1968 to 2016).
Relationship with Other Technologies: Relationship between AR and other technologies (media, technologies, and concepts related to the spectrum between real and virtual worlds as per Milgram's Reality-Virtuality Continuum).
Applications & Working: Applications of augmented reality, Working principle of AR (5 stages: camera capture, pose estimation, projection, rendering, display overlay).
Core Concepts & Ingredients: Core concepts related to augmented reality, Ingredients of an augmented reality experience (scene generator, tracking system, display, sensors, and processor).

Real World Assignment:

1. Identify and classify 5 real-world applications (e.g., Snapchat filters, Pokemon GO, Google Maps AR) on the Reality-Virtuality spectrum.
2. Use any smartphone AR app (Google Measure or IKEA Place) and record a 2-minute video explaining the 5 stages (capture, pose estimation, projection, rendering, and overlay).
3. Open any AR app (Pokemon GO or Google Maps Live View), identify and photograph or write down all 5 ingredients (scene generator, tracking system, display, sensors, and processor) with one sentence function for each.

Exemplary: ARToolKit 1999, Google Glass 2013, HoloLens 2016, Applications: Medical – AR overlays patient scans on body during surgery, Retail – Virtual try-on for glasses, clothes, or furniture (e.g., IKEA Place), Navigation – AR arrows on live camera view (e.g., Google Maps Live View).

Unit II	Augmented Reality Architecture	(07 Hours)

AR Displays: Audio displays, Haptic displays, Visual displays, and other sensory displays, Visual perception, Requirements and characteristics, Spatial display model.
Processors: Role of processors, Processor system architecture, Processor specifications.
Tracking & Sensors: Tracking, Calibration, and registration, Characteristics of tracking technology, Stationary tracking systems, Mobile sensors, Optical tracking, Sensor fusion.

Real World Assignment:

1. Visit a smartphone store, test 3 different phones with AR apps, and compare visual display quality (brightness, sharpness, color accuracy) in the same lighting condition.
2. Research and compare processor specifications of HoloLens 2 (Snapdragon 850) vs. Magic Leap 2 (AMD/ARM), and write a one-page report on how each affects AR performance.
3. Use Google Maps Live View AR at 3 locations (under trees, inside building, open ground), note tracking stability at each spot, and explain which sensor failed.

Exemplary: Google Maps Live View uses visual displays for navigation arrows, Snapdragon XR2 powers HoloLens 2 for real-time tracking, NVIDIA Jetson runs AR for warehouse robots, HoloLens uses optical tracking for hands-free operation, and Surgical AR uses stationary tracking systems for tool navigation.

Unit III	AR Techniques & Tracking Methods	(07 Hours)

Marker Based Tracking: Introduction to Marker-based tracking, Types of markers, Marker camera pose and identification, Visual tracking, Mathematical representation of matrix multiplication, template markers, 2D barcode markers, Imperceptible markers.
Marker-Less Tracking: Localization based augmentation, Real world examples.
Tracking Methods: Visual tracking, Feature based tracking, Hybrid tracking, Initialization and recovery.

Real World Assignment:

1. Use AR app with printed marker, move marker in 6 directions, record 1-min video showing virtual object locked to marker.
2. Use Google Maps Live View at 3 locations, click screenshots, and explain how app knows position without marker.
3. Identify 3 marker-less AR apps (Pokémon GO, Snapchat, Google Lens) with one sentence explanation of tracking method.

Exemplary: ArUco markers track warehouse robots for navigation, Medical surgery uses marker tracking for tool navigation, Amazon AR View shows products in living rooms using feature tracking, Snapchat uses visual tracking for face filters.

Unit IV	Introduction to Virtual Reality & Human Perception	(07 Hours)

VR Fundamentals: Defining virtual reality, History of VR, Human physiology and perception, Key elements of VR experience, Virtual reality system, Interface to virtual world (input and output including visual, aural and haptic displays), Applications of VR.
VR Representation: Representation of virtual world, Visual representation in VR, Aural representation in VR, Haptic representation in VR.
Motion & Perception in VR: Motion in real and virtual worlds (velocities and accelerations), Vestibular system, Physics in virtual world, Mismatched motion and vection.

Real World Assignment:

1. Visit VR gaming zone, try VR headset, and write report on definition, history, human perception, key elements, system, I/O displays, and applications.
2. Use any VR painting app (Tilt Brush or Open Brush), draw a simple 3D object, and note how visual depth, audio feedback, and haptic vibration help you create.
3. Use VR roller coaster app for 5 minutes, note dizziness or vection, explain mismatched motion and vestibular system response.

Exemplary: VR helps stroke patients regain arm movement, surgical simulators use haptic feedback to feel tissue resistance, driving simulators show realistic roads and traffic sounds, VR racing games use mismatched motion to simulate speed.

Unit V	**Virtual World Motion Tracking & Human Vision**	**(07 Hours)**

Tracking: Tracking 2D and 3D orientation, Tracking position and orientation, and tracking attached bodies.
Geometric Transformation: Geometric models, Changing position and orientation, Axis-angle representations of rotation, Viewing transformations, Chaining the transformations.
Human Vision for VR: Human eye, Eye movements and implications for VR.

Real World Assignment:

1. Wear VR headset, move head and hands, note how virtual objects follow real movements.
2. Use Blender or VR painting app, rotate and move a 3D object, observe how transformations chain together.
3. Wear VR headset for 10 minutes, note eye strain and how eyes move to track fast objects.

Exemplary: ArUco markers track warehouse robots for navigation, Medical surgery uses marker tracking for tool navigation, Amazon AR View shows products in living rooms using feature tracking, Snapchat uses visual tracking for face filters.

Learning Resources

Text Books:

1. Chopra, R. (2021). *Virtual and augmented reality*. Khanna Book Publishing.
2. LaValle, S. M. (2016). *Virtual reality*. Cambridge University Press.
3. Sherman, W. R., & Craig, A. B. (2002). *Understanding virtual reality: Interface, application and design (the morgan kaufmann series in computer graphics)*. Morgan Kaufmann Publishers.
4. Craig, A. B., Sherman, W. R., & Will, J. D. (2009). *Developing virtual reality applications: Foundations of effective design*. Morgan Kaufmann.

Reference Books:

1. Kim, G. J. (2005). *Designing virtual systems: The structured approach*. Springer-Verlag London 2005.
2. Bowman, D. A., Kruijff, E., LaViola Jr., J. J., & Poupyrev, I. (2005). *3D user interfaces: Theory and practice*. Addison Wesley.
3. Bimber, O., & Raskar, R. (2005). *Spatial augmented reality: Merging real and virtual worlds*. A. K. Peters, Ltd.
4. Burdea, G. C., & Coiffet, P. (2003). *Virtual reality technology*. Wiley Interscience.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Foundation for Virtual and Augmented Reality System
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs03

2. NPTEL Course: Augmented & Virtual Reality – Foundations and Applications
   https://onlinecourses.swayam2.ac.in/e-learning/preview/ntr26_ed41

3. Coursera Course: Virtual Reality Specialization
   https://www.coursera.org/specializations/virtual-reality

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PEC-361B-RAI: Mobile and Micro Robotics				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**3 Hours/Week**	**3**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:

Kinematics of Robot (PCC-252-RAI),
Dynamics and Control of Robotic Systems (PCC-301-RAI),
Autonomous Navigation using SLAM (MDM-322-RAI)

Course Objectives:

The students will be able

1. To introduce the fundamentals of mobile robots and their applications in various domains.
2. To understand the kinematics, dynamics, and sensor systems used in mobile robots.
3. To study localization, mapping, navigation, and motion control techniques in mobile robotics.
4. To explore concepts of micro-robotics, MEMS-based robotic systems, and fabrication methods.
5. To examine advanced applications of mobile manipulators, cooperative robots, and micro-robotic devices.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: UNDERSTAND the fundamental concepts, classifications, architectures, and real-world applications of mobile and micro robotic systems.

CO2: APPLY kinematic and dynamic modeling techniques for analysis and motion planning of different mobile robotic platforms.

CO3: ANALYZE localization, mapping, navigation, and sensor integration in autonomous mobile robots.

CO4: EVALUATE motion control strategies, cooperative robotic systems, and performance limitations of mobile robotic applications.

CO5: DEVELOP conceptual designs and solutions related to MEMS-based micro-robots, micro-actuators, and advanced micro-robotic devices for engineering applications.

Course Contents		
Unit I	**Introduction to Mobile Robots**	**(07 Hours)**

Introduction to Mobile Robots: Tasks of mobile robots, Tele-robotics, Service robotics, Robotics philosophy, Challenges and obstacles in mobile robotics.
Types of Mobile Robots: Wheeled robots, Legged robots, Aerial robots, Underwater robots, Surface robots.
Environment Representation: Structured and unstructured environments, Mapping concepts, Navigation challenges.
Applications of Mobile Robots: Industrial automation, Healthcare, Agriculture, Surveillance, Logistics and autonomous systems.

Real World Assignment:

1. Study and compare different types of mobile robots used in industrial and service applications.
2. Prepare a report on applications of autonomous mobile robots in smart agriculture or healthcare.
3. Analyze environmental challenges faced by mobile robots in real-world scenarios.

Exemplary:

1. Mobile robots are widely used in logistics, healthcare, and autonomous transportation systems.
2. Environment representation helps robots perform navigation and obstacle avoidance effectively.

Unit II	**Kinematics, Dynamics and Sensors for Mobile Robots**	**(07 Hours)**

Kinematics of Mobile Robots: Two-wheel, Three-wheel and Four-wheel mobile robots, Differential drive systems, Omni-directional and Mecanum wheel robots.
Dynamics of Mobile Robots: Motion modeling, Stability and mobility analysis.
Localization Sensors: Magnetic and optical position sensors, Gyroscope, Accelerometer, Magnetic compass, Inclinometer, GNSS.
Navigation Sensors: Tactile sensors, Proximity sensors, Ultrasonic rangefinder, Laser scanner, Infrared sensors, Vision systems.

Real World Assignment:

1. Develop kinematic models for differential drive and omni-directional robots.
2. Interface ultrasonic and infrared sensors for obstacle detection in mobile robots.
3. Compare localization sensors used in autonomous robotic systems.

Exemplary:

1. Sensor fusion improves the localization and navigation performance of autonomous robots.
2. Kinematic models are essential for controlling the movement of mobile robots.

Unit III	**Localization, Mapping and Motion Control**	**(07 Hours)**

Localization in Mobile Robotics: Odometry, Dead reckoning methods, Map-based localization techniques.
Mapping Techniques: Environment mapping, Grid mapping, Feature-based mapping, Simultaneous Localization and Mapping (SLAM) concepts for autonomous navigation.

Motion Control of Mobile Robots: Model-based control, Motion-based controllers, Feedback control systems, Lyapunov-based motion control techniques and stability analysis.
Advanced Mobile Robotics: Mobile manipulators, Cooperative mobile robots, Applications and limitations of autonomous robots.

Real World Assignment:

1. Implement odometry-based localization for a mobile robot platform.
2. Study SLAM algorithms used in autonomous navigation systems.
3. Design a simple motion controller for trajectory tracking in mobile robots.

Exemplary:

1. Localization and mapping are essential for autonomous navigation in unknown environments.
2. Motion control algorithms improve robotic stability and path-following accuracy.

Unit IV	Fundamentals of Micro Robotics	(07 Hours)

Introduction to Micro Robotics: Definition, Characteristics and importance of micro robots, Scope and challenges in micro robotics.
Classification of Micro Robots: Task-specific micro robots, Size-based and fabrication-based classifications, Mobility and functional classifications.
MEMS-based Micro Robotics: Fundamentals of MEMS technology, Principles of MEMS-based micro-robotics, Micro sensors and micro actuators used in robotic systems.
Applications of Micro Robots: Biomedical applications, Micro manipulation systems, Inspection and surveillance systems.

Real World Assignment:

1. Study MEMS-based micro robot applications in biomedical engineering.
2. Compare different fabrication techniques used in micro-robotics.
3. Prepare a case study on micro-robotic systems used in healthcare applications.

Exemplary:

1. MEMS technology enables compact and efficient micro robotic systems.
2. Micro-robots are widely used in biomedical and precision engineering applications.

Unit V	Micro Robotic Devices and Future Trends	(07 Hours)

Micro Robotic Actuators: Arrayed actuator principles, Electrostatic, Piezoelectric and Thermal micro-actuation mechanisms, Micro motors and smart material-based actuators.
Micro Robotic Devices: Micro grippers, Micro tools, Micro conveyors, Walking MEMS micro robots.
Multi Robot Systems: Cooperative micro robotics, Swarm-based micro robot systems, Micro robot powering methods, Wireless communication systems for micro robots.
Micro fabrication and Future Trends: Micro fabrication principles, Micromachining, Packaging and integration, Micro assembly platforms and manipulators, Emerging trends in micro robotics.

Real World Assignment:

1. Design a conceptual micro gripper for biomedical applications.
2. Analyze communication and coordination methods used in cooperative micro robotic systems.
3. Study future trends in intelligent micro-robotics and nano robotic technologies.

Exemplary:

1. Micro robotic devices enable high-precision manipulation in biomedical and industrial applications.
2. Cooperative and swarm-based micro-robot systems improve efficiency in complex micro-scale operations.

Learning Resources

Text Books:

1. Soloman, S. (2023). *Advanced robotics: Design & applications*. Khanna Book Publishing.
2. Siegwart, R., Nourbakhsh, I. R., & Sacramuzza, D. (2011). *Introduction to autonomous mobile robots* (2nd Ed.). MIT Press.
3. Tzafestas, S. G. (2021). *Introduction to mobile robot control*. Elsevier.
4. Mukherjee, S. (2023). *Essentials of robotics process automation*. Khanna Publishing House.

Reference Books:

1. Baesens, B. (2014). *Analytics in a big data world: The essential guide to data science and its applications*. Wiley.
2. Franks, B. (2012). *Taming the big data tidal wave: Finding opportunities in huge data streams with advanced analytics*. John Wiley & Sons.
3. Provost, F., & Fawcett, T. (2013). *Data science for business*. O'Reilly Media.
4. Schutt, R., & O'Neil, C. (2013). *Doing data science*. O'Reilly Media.
5. Rajaraman, A., & Ullman, J. D. (2012). *Mining of massive datasets*. Cambridge University Press.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Introduction to Mobile Robotics
   https://onlinecourses.nptel.ac.in/noc24_cs81/preview
2. NPTEL Course: Introduction to MEMS and Microsystems
   https://onlinecourses.nptel.ac.in/noc25_me69/preview
3. Coursera Course: Robot Manipulation and Wheeled Mobile Robots
   https://www.coursera.org/learn/modernrobotics-course5

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PEC-361C-RAI: Humanoid Robots				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**3 Hours/Week**	**3**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:

Kinematics of Robot (PCC-252-RAI),
Dynamics and Control of Robotic Systems (PCC-301-RAI),
Advances in Robotics and Artificial Intelligence (PCC-304-RAI)

Course Objectives:

The student will be able

1. To introduce the fundamentals of humanoid robotics and robot kinematics.
2. To understand the concepts of humanoid robot dynamics and Zero Moment Point (ZMP).
3. To explore biped walking pattern generation and control techniques.
4. To study whole-body motion generation and humanoid robot stability.
5. To examine dynamic simulation methods and applications of humanoid robots.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: UNDERSTAND the fundamentals of humanoid robots, coordinate transformations, and robot kinematics.

CO2: APPLY the concepts of ZMP and dynamics in humanoid robot motion analysis.

CO3: ANALYZE different biped walking pattern generation and stabilization techniques.

CO4: CLASSIFY various methods for whole-body motion generation and humanoid robot stability control.

CO5: APPLY dynamic simulation techniques for humanoid robotic systems and analyze rigid body dynamics.

Course Contents

Unit I	Introduction to Humanoid Robotics and Kinematics	(07 Hours)

Introduction to Humanoid Robots: Definition and classification of humanoid robots, Applications of humanoid robots in healthcare, Service, Defence, and Industrial sectors.
Coordinate Transformations and Kinematics: Characteristics of rotational motion, Velocity in three-dimensional space, Coordinate transformations in robotics, Robot data structures and programming.
Kinematics of Humanoid Robots: Forward and inverse kinematics, Joint motion and trajectory generation, Kinematic modeling of humanoid robots.

Real World Assignment:

1. Model a humanoid robot arm using forward kinematics in MATLAB/Python.
2. Implement coordinate transformation for a robotic leg movement simulation.
3. Study the motion of a humanoid robot performing object pick-and-place tasks.

Exemplary:

1. Humanoid robots are used in healthcare assistance and elderly care for human-like interaction and mobility.
2. Kinematic modeling enables humanoid robots to perform precise movement and posture control in industrial applications.

Unit II	Zero Moment Point (ZMP) and Humanoid Robot Dynamics	(07 Hours)

Introduction to ZMP: Concept of Zero Moment Point, Ground reaction forces, Stability criteria in humanoid robots, Importance of balance control in humanoid locomotion.
Measurement and Calculation of ZMP: Methods for measuring ZMP, Sensors used for balance control, Calculation of ZMP from robot motion, Real-time monitoring of stability during walking and running.
Dynamics of Humanoid Robots: Dynamic equations of motion, Center of mass and balance, Stability during walking and motion execution, Inverted pendulum model for humanoid balance analysis.
Control Techniques for Dynamic Stability: Feedback and feedforward control methods, Dynamic posture correction, Applications of gyroscope and accelerometer sensors in balance control.

Real World Assignment:

1. Measure ZMP using force sensors mounted on a humanoid robot foot.
2. Simulate humanoid robot balance using inverted pendulum models.
3. Analyze center of mass movement during robot walking cycles.

Exemplary:

1. ZMP-based balance control helps humanoid robots maintain stability during walking and climbing stairs.
2. Dynamic analysis enables robots to safely operate in uneven and human-centered environments.

Unit III	Biped Walking and Motion Control	(07 Hours)

Introduction to Biped Walking: Fundamentals of biped locomotion, Human-inspired walking mechanisms, Challenges in stable walking, Comparison between static and dynamic walking.

Walking Pattern Generation: Two-dimensional walking pattern generation, Three-dimensional walking pattern generation, ZMP-based walking pattern generation, Foot trajectory planning and gait cycle generation.

Stabilization and Control: Stabilizers in humanoid robots, Dynamic biped walking technologies, Additional methods for biped motion control, Real-time balance adjustment using sensor feedback.

Trajectory Planning and Motion Coordination: Step planning, Turning and obstacle avoidance during locomotion, Coordination between joints for smooth walking motion.

Real World Assignment:

1. Develop a simple gait generation algorithm for biped walking.
2. Simulate 2D and 3D walking trajectories for a humanoid robot.
3. Implement balance correction during walking using sensor feedback.

Exemplary:

1. Biped walking algorithms enable humanoid robots to navigate stairs, ramps, and uneven terrain.
2. Stabilization techniques improve robot mobility in rescue and service applications.

Unit IV	Whole-Body Motion Generation and Stability	(07 Hours)

Whole-Body Motion Generation: Methods for generating whole-body motion, Motion planning and coordination of multiple joints, Synchronization of upper and lower body movements.

Dynamic Stability in Motion: Conversion of motion patterns into dynamically stable motion, Real-time balance correction techniques, Stability enhancement during complex motions such as lifting and turning.

Remote Operation and Human-Robot Interaction: Remote operation of humanoid robots, Human-robot interaction methods, Gesture and voice-based robot control systems.

Fall Detection and Recovery Mechanisms: Reducing the impact of robot falls, Collision handling, Self-recovery and posture correction techniques after disturbances.

Real World Assignment:

1. Generate coordinated arm-leg movement for a humanoid robot.
2. Implement motion planning for object carrying tasks.
3. Study fall detection and recovery mechanisms in humanoid robots.

Exemplary:

1. Whole-body motion generation allows humanoid robots to perform complex tasks such as lifting and climbing.
2. Fall prevention systems improve safety and reliability in real-world humanoid robot applications.

Unit V	Dynamic Simulation and Emerging Applications	(07 Hours)

Dynamics of Rigid Bodies: Dynamics of rotating rigid bodies, Spatial velocity and acceleration, Dynamics of rigid body systems, Motion analysis using Newton-Euler formulations.
Dynamics of Link Systems: Forward and inverse dynamics, Dynamics of multi-link robotic systems, Featherstone's method for efficient dynamic computation, Joint torque analysis in humanoid robots.
Simulation Techniques for Humanoid Robots: Simulation tools for humanoid robots, Modeling using ROS, Gazebo, and MATLAB, Virtual testing and validation of robot motions.
Emerging Applications and Trends: AI integration in humanoid systems, Machine learning for adaptive robot behaviour, Future trends, Autonomous interaction and collaborative robots in healthcare and industry.

Real World Assignment:

1. Simulate humanoid robot dynamics using ROS/Gazebo or MATLAB.
2. Perform forward and inverse dynamics analysis of a robotic leg.
3. Develop a simulation of autonomous humanoid robot navigation.

Exemplary:

1. Dynamic simulation tools help engineers test humanoid robots before real-world deployment.
2. AI-integrated humanoid robots are widely used in healthcare, customer service, and collaborative industrial applications.

Learning Resources

Text Books:

1. Kajita, S., Hirukawa, H., Harada, K., & Yokoi, K. (2014). *Introduction to humanoid robotics*. Springer.
2. Nenchev, D. N., Konno, A., & Tsujita, T. (2019). *Humanoid robots: Modelling and control*. Butterworth-Heinemann.
3. Hackel, M. (2007). *Humanoid robots: Human-like machines*. I-Tech Education and Publishing.
4. Choi, B. (2019). *Humanoid robots*. InTech Publications.

Reference Books:

1. Siciliano, B., & Khatib, O. (2016). *Springer handbook of robotics* (2nd Ed.). Springer.
2. Craig, J. J. (2005). *Introduction to robotics: Mechanics and control* (3rd Ed.). Pearson Education.
3. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot modeling and control*. Wiley.
4. Niku, S. B. (2020). *Introduction to robotics: Analysis, control, applications* (3rd Ed.). Wiley.
5. Groover, M. P. (2019). *Automation, production systems, and computer-integrated manufacturing* (4th Ed.). Pearson.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Robotics and Control: Theory and Practice
   https://onlinecourses.nptel.ac.in/noc20_me03/preview
2. NPTEL Course: Experimental Robotics
   https://onlinecourses.nptel.ac.in/elearning/preview/noc24_ge31
3. NPTEL Course: Foundations of Cognitive Robotics:
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc24_me82

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PEC-361D-RAI: Agricultural Robotics				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**3 Hours/Week**	**3**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:

Sensors for Industrial Robotics (PCC-251-RAI),
Autonomous Navigation using SLAM (MDM-322-RAI)

Course Objectives:

The student will be able

1. To introduce the concepts and applications of agricultural robotics in modern farming.
2. To understand sensing and perception techniques used in agricultural robotic systems.
3. To explore navigation, localization, and autonomous operation in agricultural environments.
4. To study robotic manipulation and automation techniques for agricultural tasks.
5. To examine precision farming, data analytics, and emerging trends in agricultural robotics.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: UNDERSTAND the role and significance of agricultural robotics in improving agricultural productivity and sustainability.

CO2: IDENTIFY various sensing and perception techniques used for monitoring crops, soil, and environmental conditions.

CO3: ANALYZE navigation and localization methods for autonomous agricultural robots.

CO4: APPLY robotic manipulation and automation techniques for agricultural operations such as harvesting and planting.

CO5: EVALUATE precision farming technologies and emerging trends in agricultural robotics.

Course Contents		
Unit I	**Introduction to Agricultural Robotics**	**(07 Hours)**

Introduction to Agricultural Robotics: Overview and importance of agricultural robotics, Evolution and trends in agricultural automation, Applications in precision farming and smart agriculture.

Components of Agricultural Robots: Sensors, Actuators, Controllers, Power systems, Communication systems and embedded platforms.

Role of Robotics in Agriculture: Improvement in productivity and sustainability, Reduction in labour dependency, Automation in farming operations.

Challenges and Considerations: Ethical and environmental considerations, Economic challenges, Future scope of agricultural robotics.

Real World Assignment:

1. Study and compare different agricultural robots used in smart farming applications.
2. Prepare a case study on the use of autonomous tractors in precision farming.
3. Identify challenges faced in deploying robots in Indian agricultural environments.

Exemplary:

1. Agricultural robots improve crop productivity through automated seeding, spraying, and harvesting operations.
2. Smart robotic systems reduce labour dependency and enhance sustainable farming practices using AI and automation technologies.

Unit II	Sensing and Perception in Agricultural Robotics	(07 Hours)

Sensing Technologies: Soil moisture sensors, Temperature and humidity sensors, Crop health monitoring sensors, Environmental sensing systems.

Vision and Image Processing: Image acquisition techniques, Computer vision for crop monitoring, Disease and pest detection, Yield estimation methods.

Sensor Fusion Techniques: Multi-modal data integration, Real-time sensing and monitoring, AI-based perception systems.

Challenges in Perception: Outdoor environmental challenges, Noise and lighting variations, Robust sensing solutions for agricultural robots.

Real World Assignment:

1. Use soil moisture sensors to monitor irrigation requirements in farmland.
2. Develop a basic crop disease detection system using image processing techniques.
3. Compare RGB and multispectral imaging techniques for crop health analysis.

Exemplary:

1. Vision-based agricultural robots perform automated crop inspection and disease detection.
2. Sensor fusion enables accurate environmental monitoring for precision farming applications.

Unit III	Navigation and Localization in Agricultural Environments	(07 Hours)

Navigation Systems: GPS, GNSS, and RTK-based navigation, Autonomous navigation techniques, Mobile robot navigation in farms.
Localization Techniques: Indoor and outdoor localization methods, Mapping and positioning systems, Localization algorithms for field robots.
Path Planning and Obstacle Avoidance: Route optimization techniques, Obstacle detection using sensors, Safe navigation strategies.

Integration for Autonomous Operation: Perception and mapping integration, Autonomous field operations, Real-world navigation challenges in agriculture.

Real World Assignment:

1. Simulate GPS-based navigation for an agricultural mobile robot.
2. Develop an obstacle avoidance system using ultrasonic or LiDAR sensors.
3. Perform path planning for autonomous crop monitoring robots in a greenhouse environment.

Exemplary:

1. Autonomous agricultural vehicles use RTK-GPS for accurate field navigation and precision farming operations.
2. Path planning and localization techniques improve operational efficiency in autonomous farming systems.

Unit IV	Robotic Manipulation and Automation in Agriculture	(07 Hours)

Robotic Manipulators and Grippers: Agricultural robotic arms, End-effectors for harvesting and planting, Fruit and vegetable handling mechanisms.
Automation in Agricultural Tasks: Automated harvesting systems, Pruning and seeding robots, Spraying and sorting applications.
Control and Decision-Making: Autonomous control systems, AI-based decision-making algorithms, Precision operation in agricultural robots.
Human-Robot Interaction: Collaborative robots in farming, Safety in agricultural automation, Human assistance systems.

Real World Assignment:

1. Design a robotic gripper for fruit harvesting applications.
2. Develop a simulation model for automated seed planting operations.
3. Study the use of collaborative robots in greenhouse farming applications.

Exemplary:

1. Robotic manipulators enable automated harvesting of fruits and vegetables with minimal crop damage.
2. Collaborative robots assist farmers in repetitive agricultural tasks, improving productivity and operational safety.

Unit V	Precision Farming and Emerging Trends in Agricultural Robotics	(07 Hours)

Precision Farming Technologies: Smart agriculture concepts, Data-driven farming methods, IoT-based agricultural monitoring systems.
Data Analytics and Machine Learning: Crop yield prediction, Pest and disease detection, Resource optimization techniques.
Emerging Technologies: Swarm robotics, Soft robotics, Drone-based farming systems, AI integration in agriculture.
Future Trends and Challenges: Regulatory and policy aspects, Sustainable agricultural automation, Future scope of intelligent farming systems.

Real World Assignment:

1. Build an IoT-based smart irrigation monitoring system.
2. Use machine learning techniques to predict crop yield from environmental data.
3. Study the role of drone-based monitoring systems in precision agriculture.

Exemplary:

1. IoT-enabled agricultural robots support real-time monitoring and intelligent decision-making in precision farming.
2. Swarm robotics and AI-based systems are transforming future agricultural automation and smart farming applications.

Learning Resources

Text Books:

1. Karkee, M., & Zhang, Q. (2021). *Fundamentals of agricultural and field robotics*. Springer.
2. Hong, T. S., Virk, G. S., & Yuta, S. (2017). *Agricultural robots: Mechanisms, controls, and applications*. CRC Press.
3. Saini, A. (2025). *Robotics in agriculture: Automating farming processes*. Zenodo.
4. Shamshiri, R. R., & Hameed, I. A. (2025). *Mobile robots for digital farming*. CRC Press.

Reference Books:

1. Adamides, G., & Edan, Y. (2024.). *Advances in human–robot collaboration in agricultural robotics*. Springer
2. Bechar, A. (2021). *Innovation in agricultural robotics for precision agriculture: A roadmap for integrating robots in precision agriculture*. Springer.
3. Srivastava, S. K., Srivastava, D., Cengiz, K., & Gaur, P. (2024). *Smart agritech: Robotics, AI, and internet of things (IoT) in agriculture.* Scrivener Publishing.
4. Yun, Y., Sheng, W., & Zhang, Z. (2024). *Advanced sensing and robotics technologies in smart agriculture*. Springer.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Precision Agriculture
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_ag02

2. NPTEL Course: Artificial Intelligence
   https://nptel.ac.in/courses/106105077

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PEC-362A-RAI: Advanced Artificial Intelligence				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**3 Hours/Week**	**3**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:

Basics of Robotics & AI (PCC-201-RAI),
Advances in Robotics and Artificial Intelligence (PCC-304-RAI)

Course Objectives:

The students will be able to

1. To introduce the concepts of probability theory, graphical models, and inference techniques used in advanced AI systems.
2. To understand exact and approximate inference methods in probabilistic graphical models.
3. To study optimization techniques, learning methods, and structure learning in graphical models.
4. To explore causality, utility theory, and decision-making concepts in intelligent systems.
5. To analyze sequential decision problems, game theory concepts, and applications of advanced AI techniques.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: UNDERSTAND probability theory, graphical models, and inference mechanisms used in advanced artificial intelligence systems.

CO2: APPLY exact and approximate inference techniques for solving probabilistic reasoning problems.

CO3: ANALYZE optimization, parameter estimation, and structure learning methods in graphical models.

CO4: EVALUATE causality, utility functions, and decision-making strategies in intelligent systems.

CO5: DEVELOP AI-based solutions using sequential decision-making and game theory concepts for real-world applications.

Course Contents

Unit I	Probability Theory and Graphical Models	(07 Hours)

Introduction to Probability Theory: Random variables, Conditional probability, Bayes theorem, Joint and marginal probability distributions.
Bayesian Networks: Representation of uncertainty, Directed graphical models, Conditional independence concepts, Independence and I-Maps.
Undirected Graphical Models: Markov networks, Graph structures, Factorization and probabilistic reasoning.
Applications of Graphical Models: Intelligent systems, Medical diagnosis, Robotics and autonomous systems.

Real World Assignment:

1. Study the application of Bayesian networks in intelligent healthcare systems.
2. Compare directed and undirected graphical models used in AI applications.
3. Analyze probabilistic reasoning techniques used in autonomous robotic systems.

Exemplary:

1. Bayesian networks are widely used for reasoning under uncertainty in AI systems.
2. Graphical models improve representation and inference in complex intelligent applications.

Unit II	Exact and Approximate Inference Techniques	(07 Hours)

Exact Inference Methods: Variable elimination, Clique tree algorithms, Belief propagation tree construction.
Template-Based Representations: Local probabilistic models, Shared parameter models and structured representations.
Approximate Inference Methods: Sampling techniques, Monte Carlo methods, Markov chains and probabilistic sampling.
Inference in Temporal Models: MAP inference, Dynamic probabilistic models and sequential inference systems.

Real World Assignment:

1. Implement variable elimination for a simple Bayesian network.
2. Study sampling techniques used in probabilistic inference systems.
3. Analyze temporal inference methods used in autonomous navigation systems.

Exemplary:

1. Approximate inference techniques help solve large-scale probabilistic problems efficiently.
2. Belief propagation improves reasoning in graphical model-based AI systems.

Unit III	Optimization and Learning in Graphical Models	(07 Hours)

Optimization Techniques: Introduction to optimization, Objective functions, Constraint handling methods, Convex and non-convex optimization approaches used in AI systems.
Learning Graphical Models: Parameter estimation methods, Maximum likelihood estimation, Bayesian learning approaches and shared parameter networks.

Structure Learning: Structure search methods, Score-based and constraint-based learning, Learning from partially observed data and hidden variables.
Learning Algorithms: Gradient descent, Expectation-Maximization (EM) algorithm and undirected structure learning.

Real World Assignment:

1. Implement gradient descent for parameter optimization in AI models.
2. Study EM algorithm applications in machine learning systems.
3. Analyze structure learning methods used in probabilistic graphical models.

Exemplary:

1. Optimization techniques improve the learning performance and accuracy of AI systems.
2. Hidden variable models are useful for representing incomplete and uncertain data.
3. Structure learning enables intelligent systems to automatically discover relationships from data.

Unit IV	Causality and Decision Theory	(07 Hours)

Causality in Artificial Intelligence: Cause-effect relationships, Causal reasoning, Probabilistic causation, Causal graphs and inference in intelligent systems.
Utility Functions: Concepts of utility, Preference modeling, Risk analysis and rational decision-making under uncertainty.
Decision Problems: Expected utility, Value of information, Decision trees and uncertainty handling in AI-based systems.
Decision Theory Applications: Decision analysis in robotics, Healthcare and intelligent systems.

Real World Assignment:

1. Study utility-based decision-making in autonomous robots.
2. Analyze causality models used in intelligent recommendation systems.
3. Prepare a report on decision theory applications in healthcare AI.

Exemplary:

1. Utility theory helps intelligent systems make rational decisions under uncertainty.
2. Causal reasoning improves explainability and reliability in AI systems.

Unit V	Sequential Decision Making and Game Theory	(07 Hours)

Sequential Decision Problems: Sequential planning, Decision processes, Policy selection and optimization of long-term actions in intelligent systems.
Decision-Making Basics: Utility theory, Sequential decision models, Intelligent agent behavior and decision-making under uncertain environments.
Elementary Game Theory: Strategic interactions, Cooperative and non-cooperative games, Nash equilibrium concepts.
Applications of Advanced AI: Reinforcement learning concepts, Multi-agent systems and intelligent decision-support applications.

Real World Assignment:

1. Study sequential decision-making in autonomous vehicle systems.
2. Analyze game theory applications in multi-agent robotic systems.
3. Prepare a case study on AI-based decision-support systems used in industries.

Exemplary:

1. Sequential decision-making enables intelligent agents to optimize long-term actions.
2. Game theory concepts improve coordination and strategy development in multi-agent AI systems.

Learning Resources

Text Books:

1. Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques*. MIT Press.
2. Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th Ed.). Pearson Publication.
3. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
4. Shi, Z. (2011). *Advanced artificial intelligence*. World Scientific Publishing Company.

Reference Books:

1. Trivedi, M. C. (2025). *A classical approach to artificial intelligence*. Khanna Book Publishing.
2. Luger, G. F. (2008). *Artificial intelligence* (5th ed.). Pearson Education.
3. Trivedi, M. C. (2025). *Data science and data analytics using Python*. Khanna Book Publishing.
4. Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Machine Learning
   https://onlinecourses.nptel.ac.in/noc25_cs74/preview

2. NPTEL Course: Deep Learning
   https://onlinecourses.nptel.ac.in/noc25_cs124/preview

3. Coursera Course: Advanced Topics in Artificial Intelligence
   https://www.coursera.org/learn/advanced-topics-in-artificial-intelligence

4.

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
PEC-362B-RAI: Data Analytics				
Teaching Scheme		**Credit**	**Examination Scheme**	
Theory	**3 Hours/Week**	**3**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite:
Mathematics for Intelligent Systems (MDM-221-RAI)

Course Objectives:

The students will be able

1. To understand descriptive, predictive, and prescriptive analytics and the data analytics lifecycle.
2. To apply data cleaning, transformation, and normalization techniques.
3. To utilize Python libraries (NumPy, Pandas, SciPy, Matplotlib) for data manipulation, statistical computations, and database operations.
4. To create effective single-variable and multi-variable visualizations to discover patterns and communicate insights.
5. To apply descriptive and inferential statistics including hypothesis testing, regression, and basic machine learning algorithms (classification, clustering, association rule mining).

Course Outcomes:

On completion of the course, learner will be able to:

CO1: DIFFERENTIATE between descriptive, predictive, and prescriptive analytics and classify data types (nominal, ordinal, interval, and ratio) for real-world datasets.

CO2: PERFORM data pre-processing steps such as handling missing values, outliers, normalization, binning, and encoding categorical variables using Python.

CO3: GENERATE appropriate visualizations (histograms, box plots, scatter plots, parallel coordinate plots) to explore and present univariate and multivariate data.

CO4: COMPUTE descriptive statistics, apply probability distributions, and conduct hypothesis tests (t-test, chi-square, ANOVA) and linear regression to draw inferences from data.

CO5: IMPLEMENT basic machine learning models including Naïve Bayes, Decision Tree, Logistic Regression, K-Means clustering, and Apriori algorithm for pattern discovery and prediction.

Course Contents

Unit I	Foundations of Data Analytics	(07 Hours)

Descriptive, Predictive, and Prescriptive Analytics: Definition and comparison of the three analytics types, Understanding data types (qualitative vs. quantitative, nominal, ordinal, interval, ratio), Overview of the data analytics lifecycle: Problem definition, data collection, analysis, interpretation, and decision-making.

Data Pre-Processing Steps: Data cleaning (handling inconsistencies, outliers, duplicates), Data transformation (scaling, aggregation), Introduction to data visualization as the final step for pattern discovery and result communication.

Real World Assignment:

1. Generate a conveyor sensor dataset. Classify each attribute by data type. Write one descriptive, predictive, and prescriptive analytics question.
2. Implement the 5-step data analytics lifecycle for adaptive speed control. Submit: problem statement, analysis method, decision flowchart.
3. Build a cleaning pipeline for raw PLC data (missing values, outlier, duplicates, inconsistent labels). Submit script, cleaned dataset and justification.
4. Create a two-page dashboard: one for pattern discovery, one for management communication. Submit dashboard and short report on chart choices.

Exemplary: Mobile robot conveyor integration, Collaborative robot (cobot) sorting line, Autonomous mobile robot (AMR) parcel handoff, Robotic warehouse picking system, Drone package delivery drop zone, Robotic arm sorting with vision system.

Unit II	Data Analytics Tools & Data Handling	(07 Hours)

Python Libraries for Analytics: Introduction to NumPy (arrays, vectorized operations), Pandas (Series, DataFrame, data manipulation), SciPy (statistical functions), Matplotlib (basic plotting), Performing statistical procedures like mean, median, variance, and correlation using these libraries.

Data Operations in Python: Importing/exporting data from CSV, Excel, JSON, Handling missing values through deletion and imputation, Data formatting, normalization (min-max, z-score), Binning, Converting categorical variables using label encoding and one-hot encoding, Accessing databases (SQLite/MySQL) using Python.

Real World Assignment:

1. Record 50 readings of any physical process (room temperature, fan speed, motor noise). Import into Pandas. Compute mean, median, variance, correlation. Plot using Matplotlib.
2. Download any real-world CSV (weather, traffic, sales) with missing values. Clean it using Pandas (dropna, fillna). Normalize one column (min-max). Bin another column into 3 groups. Submit before/after.
3. Create an SQLite table of employee/customer data with categorical columns (city, gender, department). Use Python to apply label encoding and one-hot encoding. Export final DataFrame to JSON.

Exemplary: Robot arm torque analysis, Vision-guided robot sorting, Swarm robot distance tracking, Humanoid robot gait analysis, Robot vacuum floor mapping, Collaborative robot force data, Drone flight log database.

Unit III	Data Visualization	(07 Hours)
Principles & Single-Variable Charts: Graphic representation of data and characteristics of effective graphical displays (clarity, accuracy, simplicity), Single-variable charts: Dot plot, jitter plot, error bar plot, box-and-whisker plot, histogram. **Multi-Variable Charts:** Two-variable charts: Bar chart, scatter plot, line plot, log-log plot, Charts for more than two variables: Stacked plots, parallel coordinate plot.		
Real World Assignment: 1. Plot a histogram, box plot, and jitter plot for 200 robot motor current readings. Identify outliers and skewness. 2. Using AMR speed and battery voltage data, create a scatter plot, line plot, and log-log plot. State the correlation. 3. For three robot types (articulated, SCARA, cartesian), create a stacked bar chart and a parallel coordinate plot using four performance metrics. Write one insight per chart.		
Exemplary: Robot joint temperature monitoring, Robot arm vibration study, Swarm robot spacing comparison, Humanoid robot gait stability, Collaborative robot force monitoring, Drone altitude control testing, Robot vacuum coverage efficiency.		
Unit IV	**Descriptive and Inferential Statistics**	**(07 Hours)**
Descriptive Statistics & Probability Distributions: Measures of central tendency (mean, median, mode), Measures of dispersion (range, variance, standard deviation), Measures of shape (skewness, kurtosis), Probability distributions: Normal, binomial, poisson distributions. **Inferential Statistics**: Hypothesis testing: Null/alternative hypotheses, p-value, significance level, Type I/II errors, t-test, chi-square test, ANOVA (one-way and two-way), Regression analysis: Simple and multiple linear regression, evaluation metrics (R-squared, RMSE).		
Real World Assignment: 1. Compute mean, median, variance, skewness, and kurtosis for 500 robot motor current readings. State if data is normally distributed. 2. Using binomial distribution (5% defect rate), find P (exactly 3 defects in 20 items). Using Poisson (avg 2.5 defects/hour), find P (2 defects in 1 hour). 3. Perform t-test to compare speeds of two robot brands (α=0.05). State null hypothesis, p-value, conclusion, and Type I/II error possibility.		
Exemplary: Conveyor sorting accuracy, Robot vacuum coverage time, Drone hover stability, Vision system misclassification, Swarm robot communication delay, Collaborative robot force monitoring, AMR battery life testing, Robot arm cycle time analysis.		
Unit V	**Machine Learning Concepts**	**(07 Hours)**
Classification Algorithms: Overview of supervised learning, Bayes' classifier (Naïve Bayes), Decision tree (entropy, information gain, pruning, etc.), Logistic regression (sigmoid function), Support vector machines (SVM): Hyperplane and kernel trick basics.		

Clustering & Advanced Topics: Overview of unsupervised learning, K-Means algorithm (centroid, elbow method), Apriori algorithm (support, confidence, and lift), Introduction to recommendation systems: Collaborative filtering and content-based filtering basics.

Real World Assignment:

1. Using conveyor sorting data (weight, speed, vibration, etc.), build a Naïve Bayes and a Decision Tree classifier to predict defective items. Compare accuracy.
2. Apply K-Means to 200 robot joint angle readings. Use elbow method to find optimal K. Submit elbow plot and cluster interpretation.

Exemplary: Robot arm fault classification, Drone flight pattern analysis, Robot maintenance recommendation, Warehouse robot basket analysis, Swarm robot behavior grouping, Vision-based object sorting, Collaborative robot collision detection.

Learning Resources

Text Books:

1. Jain, V. K. (2023a). *Data science and analytics (with Python, R, SPSS programming)*. Khanna Book Publishing.
2. Chopra, R. (2023). *Data science with AI, ML, DL*. Khanna Book Publishing.
3. Sivanandam, S. N., & Deepa, S. N. (2006). *Introduction to neural networks using Matlab 6.0*. Tata McGraw-Hill Education.
4. Acharya, S., & Chellapan, S. (2015). *Big data and analytics*. Wiley.

Reference Books:

1. Baesens, B. (2014). *Analytics in a big data world: The essential guide to data science and its applications*. Wiley Publishers.
2. Franks, B. (2012). *Taming the big data tidal wave: Finding opportunities in huge data streams with advanced analytics*. John Wiley & Sons.
3. Provost, F., & Fawcett, T. (2013). *Data science for business*. O'Reilly Publishers.
4. Schutt, R., & O'Neil, C. (2013). *Doing data science*. O'Reilly Publishers.
5. Rajaraman, A., & Ullman, J. D. (2012). *Mining of massive datasets*. Cambridge University Press.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Introduction to Data Analytics
   https://nptel.ac.in/courses/110106072

2. NPTEL Course: Data Analytics with python
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs86

3. Coursera Course: Introduction to Data Analytics
   https://www.coursera.org/learn/introduction-to-data-analytics

<table>
<tr><th colspan="5">Savitribai Phule Pune University
Third Year of Robotics and Artificial Intelligence (2024 Pattern)</th></tr>
<tr><th colspan="5">PEC-362C-RAI: Mechatronics System Design</th></tr>
<tr><th colspan="2">Teaching Scheme</th><th>Credit</th><th colspan="2">Examination Scheme</th></tr>
<tr><td>Theory</td><td>3 Hours/Week</td><td rowspan="2">3</td><td>CCE</td><td>30 Marks</td></tr>
<tr><td>Practical</td><td>NA</td><td>End-Semester</td><td>70 Marks</td></tr>
</table>

Prerequisite:

Analog and Digital Electronics (PCC-203-RAI),
Sensors for Industrial Robotics (PCC-251-RAI)

Course Objectives:

The students will be able

1. To introduce the principles of mechatronic design, including adaptive control, distributed systems, and integrated product development.
2. To provide knowledge of real-time data acquisition, I/O processes, and open-source microcontroller platforms (Arduino, Raspberry Pi) with wireless communication modules.
3. To explain calibration techniques for various sensors and transducers through practical case studies.
4. To analyze control system applications such as temperature control, inverted pendulum, and pick-and-place robots.
5. To design mechatronic products including motor control, ECU, barcode reader, AI-based detection, and FMEA.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: APPLY mechatronics design with mechanisms, load conditions, isolation, ergonomics, and safety.
CO2: IMPLEMENT real-time data acquisition using microcontrollers and wireless modules with mobile apps.
CO3: CALIBRATE sensors and transducers for automotive and industrial applications.
CO4: ANALYZE and implement control strategies for systems such as inverted pendulum, temperature control, pick-and-place robots, and carpark barriers.
CO5: DESIGN mechatronic products involving DC/AC motor control, ECU, barcode reader, advanced actuators, AI-based detection, and FMEA reliability.

Course Contents		
Unit I	**Mechanical Systems Design & Human Factors**	**(07 Hours)**

Mechatronics Design Approach: Control program control, Adaptive control and distributed systems, Design process, Types of design (original, adaptive, variant), Integrated product design, Mechanisms (linkages, cams, gears), Load conditions (static, dynamic, cyclic), Design of flexible structures, Environmental isolation (vibration, thermal).
Man-Machine Interface & Safety: Industrial design and ergonomics, Information transfer (machine to man and man to machine), Visual and tactile displays, Control devices, Safety (fail-safe design, emergency stops, guarding).

Real World Assignment:

1. Design an adaptive speed control for a conveyor sorting system that responds to static, dynamic, and cyclic loads using distributed PLCs.
2. Specify fail-safe braking, dual emergency stops, and interlocked guarding for a hydraulic press with visual and tactile operator displays.
3. Select a cam, linkage, and gear train for a pick-and-place robot arm, justifying original vs. adaptive design choices.
4. Design a vibration-isolated mounting platform for a sensitive optical instrument, including ergonomic placement of controls and thermal isolation.

Exemplary: Adaptive cruise control in automobiles, Hydraulic press with emergency stops and guarding, Cam-driven pick-and-place robot arm, Vibration-isolated optical table for laboratories, Gear train in an electric vehicle transmission, Fail-safe braking system in elevators.

Unit II	**Real-Time Interfacing & Microcontrollers**	**(07 Hours)**

Real-Time Data Acquisition & I/O: Elements of data acquisition and control (sensors, signal conditioning, ADC/DAC, actuators), Overview of I/O process (digital, analog, serial), Installation of I/O card & software, Installation of application software, Over framing (sampling rate, buffering).
Microcontrollers & Wireless Integration: Introduction to open-source hardware (Arduino, Raspberry Pi), shields/modules for GPS, GPRS/GSM, Bluetooth, RFID, Xbee, Integration with wireless networks (Wi-Fi), Databases and web pages, Web and mobile phone apps.

Real World Assignment:

1. Design a real-time temperature monitoring system using a sensor, signal conditioning, ADC, and Arduino with proper sampling rate and buffering.
2. Develop a GPS and GSM-based vehicle tracking system using Raspberry Pi, Xbee, and a web database with live location display.
3. Create a Bluetooth-controlled home automation system (RFID door access + Wi-Fi database logging) with an Android app interface.

Exemplary: Arduino-based weather station with SD card logging, Raspberry Pi surveillance camera with Wi-Fi streaming, RFID attendance system with database storage, Bluetooth-controlled robotic car, Xbee-based wireless sensor network for agriculture, GPRS-enabled industrial machine monitor with SMS alert.

Unit III	Sensors, Calibration & Control Applications	(07 Hours)

Sensor & Transducer Calibration Systems: Transducer calibration system for automotive applications, Strain gauge weighing system, Solenoid force-displacement calibration system, Rotary optical encoder, Thermocouple calibration.
Control System Case Studies: Inverted pendulum control, Controlling temperature of a hot/cold reservoir, Pick and place robot, Carpark barriers, PLC vs microcontroller comparison.

Real World Assignment:

1. Calibrate a thermocouple and a rotary optical encoder for an automotive engine test stand, comparing readings to a reference standard.
2. Design a PID controller for an inverted pendulum system and contrast it with a PLC-based car park barrier control logic.
3. Develop a calibration procedure for a solenoid force-displacement system and a strain gauge weighing system used in a pick-and-place robot.

Exemplary: Strain gauge-based electronic weighing scale calibration, Solenoid force-displacement tester for automotive fuel injectors, Rotary optical encoder calibration for CNC machine feedback, Inverted pendulum balance control system (Segway-type vehicle), PID-controlled temperature bath for hot/cold reservoir calibration.

Unit IV	Process Control & Mechatronic Product Design	(07 Hours)

Process & Thermal Control Case Studies: Thermal cycle fatigue of a ceramic plate, pH control system, De-icing temperature control system, Skip control of a CD player, Autofocus camera and exposure control.
Mechatronics Product Design Case Studies: Motion control using D.C. motor (PWM, H-bridge, encoder), A.C. motor (VFD control), Solenoids, Car engine management system (ECU, sensors, fuel injection), Barcode reader.

Real World Assignment:

1. Design a temperature control system that cycles a ceramic plate between 0°C and 200°C without causing thermal fatigue. Also, add a de-icing control logic to keep the plate above freezing during cold starts. Use thermocouple feedback and a heating/cooling actuator.
2. Design a DC motor speed/position control using PWM, H-bridge, and encoder feedback. Separately, implement VFD control for an AC motor. Compare both for a conveyor belt application in terms of torque ripple and efficiency.
3. Analyze the engine control unit (ECU) of a car – including sensors (O2, knock, coolant) and fuel injector actuation. Contrast it with a barcode reader's autofocus and exposure control loop. Show how both follow the same sensor, decision, actuator architecture.

Exemplary: Ceramic plate thermal cycle fatigue tester, Industrial pH neutralization control system, Aircraft wing de-icing temperature controller, CD player skip control mechanism, Autofocus camera with exposure control, PWM & H-bridge based DC motor speed controller, VFD controlled AC motor for industrial conveyor.

Unit V	Advanced Actuators, AI Integration & System Reliability	(07 Hours)

Advanced Actuators & Smart Materials: Shape memory alloys (SMA), Piezoelectric actuators, Magnetorheological (MR) fluids, Ultrasonic Motors: Principles, Driving circuits. **AI & Machine Learning in Mechatronics:** Supervised learning for sensor fusion, Anomaly detection in real-time control, Neural network-based PID tuning, Rule-based expert systems for fault diagnosis. **System Integration & Reliability:** Electromagnetic compatibility (EMC), Grounding and shielding, Power budgeting for embedded systems, Watchdog timers, Redundancy concepts (hardware vs. software), And Failure Mode Effects Analysis (FMEA) for mechatronic products.

Real World Assignment:

1. Design a piezoelectric actuator-based micro-positioning stage for a CNC machine. Propose a driving circuit, compare its step resolution to a conventional stepper motor.
2. Develop a neural network-based anomaly detection system for a DC motor used in a pick-and-place robot. Train it using encoder current and vibration data (simulated or real) to predict bearing failure.

Exemplary: SMA wire-actuated micro-gripper for soft robotics, MR fluid-based adaptive shock absorber for automotive suspension, Ultrasonic motor in a camera autofocus lens (compared to voice coil), AI-predicted maintenance on an industrial conveyor bearing using vibration spectra.

Learning Resources

Text Books:

1. Bolton, W. (2021). *Mechatronics: Electronic control systems in mechanical and electrical engineering (7th Ed.).* Pearson Education.
2. Shetty, D., & Kolk, R. A. (2010). *Mechatronics system design (2nd Ed.).* Cengage Learning.
3. Alciatore, D. G. (2018). *Introduction to mechatronics and measurement systems (5th Ed.).* McGraw-Hill Education.
4. Carryer, J. E., Ohline, R. M., & Kenny, T. W. (2011). *Introduction to mechatronic design.* Pearson.

Reference Books:

1. Janschek, K. (2012). *Mechatronic systems design: Methods, models, concepts.* Springer Berlin Heidelberg.
2. De Silva, C. W. (2019). *Mechatronic systems: Devices, design, control, operation and monitoring.* CRC Press.
3. Bishop, R. H. (2017). *The mechatronics handbook (2nd Ed.).* CRC Press.
4. Munnig Schmidt, R., Schitter, G., & Rankers, A. (2020). *The design of high-performance mechatronics: High-tech functionality by multidisciplinary system integration (3rd rev. Ed.).* IOS Press.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Design of Mechatronics System
   https://onlinecourses.nptel.ac.in/noc21_me129/preview
2. NPTEL Course: Mechatronics
   https://onlinecourses.nptel.ac.in/noc21_me27/preview
3. Coursera Course: CPS Design for Mechatronics, Healthcare, EV & Robotics
   https://www.coursera.org/learn/cps-design-for-mechatronics-healthcare-ev--robotics

Savitribai Phule Pune University
Third Year of Robotics and Artificial Intelligence (2024 Pattern)

PEC-362D-RAI: AI for Medical Applications

Teaching Scheme		Credit	Examination Scheme	
Theory	**3 Hours/Week**	**3**	**CCE**	**30 Marks**
Practical	**NA**		**End-Semester**	**70 Marks**

Prerequisite :

Biomechanics and Biomaterials (MDM-271-RAI),
Machine Learning for Robotics (PCC-302-RAI)

Course Objectives:

The student will be able

1. To introduce the fundamental concepts of Artificial Intelligence in healthcare and medical applications.
2. To understand machine learning techniques used in clinical diagnosis and medical data analysis.
3. To explore knowledge representation and reasoning methods in intelligent healthcare systems.
4. To study implementation tools, evaluation techniques, and ethical considerations in medical AI systems.
5. To examine advanced AI applications in healthcare, personalized medicine, and future trends in medical robotics.

Course Outcomes:

On completion of the course, learner will be able to:

CO1: UNDERSTAND the concepts and applications of Artificial Intelligence in healthcare systems.

CO2: APPLY machine learning and deep learning techniques for medical data analysis and disease prediction.

CO3: ANALYZE knowledge representation and reasoning methods used in intelligent healthcare applications.

CO4: EVALUATE AI models in healthcare using performance metrics, ethical principles, and regulatory considerations.

CO5: DEVELOP AI-based healthcare and medical robotic solutions for diagnosis, monitoring, and personalized treatment.

Course Contents

Unit I	Foundations of Artificial Intelligence in Healthcare	(07 Hours)

Introduction to Human and Artificial Intelligence: Terminologies, Computational models of intelligence, Conceptual frameworks from cognitive psychology, Neuroscience, Information theory and linguistics.

Philosophical foundations of AI: Review of mathematical and statistical concepts, Logarithmic loss, Cross entropy, Optimizing cost functions, Linear regression and logistic regression.

Applications of AI in Healthcare: Role of AI in diagnosis, Treatment planning, Medical imaging and patient monitoring.

Real World Assignment:

1. Implement logistic regression for predicting the presence of diabetes using healthcare datasets.
2. Compare cross-entropy loss and logarithmic loss using a medical classification dataset.
3. Study and present AI-based healthcare systems used in hospitals.

Exemplary:

1. AI systems assist doctors in disease diagnosis and patient risk prediction using large-scale healthcare data.
2. Regression and optimization techniques improve the accuracy of intelligent medical decision systems.

Unit II	Machine Learning Techniques for Medical Applications	(07 Hours)

Forms of Learning: Supervised, Semi-supervised, Unsupervised, Active and transfer learning.

Supervised Learning: Decision trees, Support vector machines, Non-parametric learning methods.

Bio-inspired Learning: Perceptron models, Artificial neural networks, Deep neural networks, Recurrent neural networks, Convolutional neural networks.

Unsupervised Learning: Clustering methods, Dimensionality reduction, Feature selection, and Feature extraction.

Real World Assignment:

1. Train a CNN model for classification of medical images such as X-rays or MRI scans.
2. Apply clustering techniques to group patients based on symptoms and diagnostic reports.
3. Implement SVM for detecting heart disease using healthcare datasets.

Exemplary:

1. Deep learning techniques improve medical image analysis and automated diagnosis systems.
2. Machine learning algorithms support early disease detection and healthcare analytics.

Unit III	Knowledge Representation and Reasoning in Healthcare	(07 Hours)

Knowledge Representation: Propositional logic, First-order logic, Semantic networks, and Ontological engineering for healthcare applications.

Reasoning Techniques: Rule-based reasoning, Probabilistic reasoning, Bayesian networks, and Uncertainty handling in medical diagnosis.
Time-Series Analysis: Temporal reasoning and probabilistic models for patient monitoring, and Disease progression prediction.
Medical Expert Systems: Clinical decision support systems (CDSS), and Intelligent treatment recommendation systems.
Emerging Paradigms: Artificial social intelligence, Emotional intelligence in healthcare robots, and Conversational AI for healthcare assistance.

Real World Assignment:

1. Develop a rule-based expert system for symptom-based disease prediction.
2. Use probabilistic reasoning for patient risk assessment in ICU monitoring.
3. Create ontology-based healthcare knowledge representation using Protégé tools.

Exemplary:

1. Knowledge representation enables expert systems for automated clinical decision support.
2. Probabilistic reasoning improves intelligent patient monitoring and predictive healthcare systems.
3. Conversational AI assists patients in healthcare support and appointment management.

Unit IV	Implementation, Evaluation and Ethics of Medical AI	(07 Hours)

Implementation of Medical AI: Tools and technologies for AI implementation in healthcare systems.
Model Evaluation and Performance Metrics: Accuracy, Precision, Recall, F1-score, ROC curve, Cross-validation, Model interpretability.
Ethics of AI: Bias, Fairness, Accountability, Transparency, Privacy, and Security issues in healthcare applications.
Legal and Social Issues: Ethical frameworks and regulatory challenges in AI-powered medicine.

Real World Assignment:

1. Evaluate the performance of different ML models on healthcare datasets using performance metrics.
2. Analyze bias and fairness issues in AI-based patient diagnosis systems.
3. Prepare a report on legal and ethical guidelines for AI in healthcare.

Exemplary:

1. Model evaluation ensures reliable and interpretable AI systems in clinical applications.
2. Ethical AI frameworks help maintain fairness, transparency, and patient safety in healthcare technologies.

Unit V	Advanced Applications and Future Trends in Medical AI	(07 Hours)

Applications in Healthcare: Risk stratification, Disease progression modeling, Intelligent clinical decision systems, Phenotype discovery, and Medical imaging analysis.
Personalized Medicine: AI-driven treatment recommendations and predictive healthcare systems.
AI in Medical Robotics: Applications in robotic surgery, Rehabilitation robotics, and Assistive healthcare robots.
Future Trends and Challenges: AI for precision medicine, Explainable AI, AI-integrated wearable devices, Emerging healthcare technologies.

Real World Assignment:

1. Develop an AI-based clinical decision support system prototype.
2. Analyze tissue morphology using image processing and deep learning techniques.
   Design a concept for AI-enabled robotic assistance in rehabilitation therapy.

Exemplary:

1. AI-powered clinical systems support evidence-based diagnosis and personalized treatment planning.
2. Intelligent healthcare robots improve rehabilitation, surgical precision, and patient care.

Learning Resources

Text Books:

1. Topol, E. (2019). *Deep medicine: How artificial intelligence can make healthcare human again*. Basic Books.
2. Cleophas, T. J., & Zwinderman, A. H. (2018). *Machine learning in medicine: A complete overview.* Springer.
3. Reddy, C. K., & Aggarwal, C. C. (2015). *Healthcare data analytics*. CRC Press.
4. Deserno, T. M. (2020). *Medical image analysis*. Springer.

Reference Books:

1. Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th Ed.). Pearson.
2. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
4. Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification* (2nd Ed.). Wiley.
5. Nilsson, N. J. (2009). *The quest for artificial intelligence*. Cambridge University Press.
6. Shortliffe, E. H., & Cimino, J. J. (Eds.). (2014). *Biomedical informatics: Computer applications in health care and biomedicine* (4th Ed.). Springer.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Medical Image Analysis
   https://onlinecourses.nptel.ac.in/noc25_ee91/preview
2. NPTEL Course: AI in Healthcare
   https://onlinecourses.nptel.ac.in/noc25_cs95/preview
3. NPTEL Course: Artificial Intelligence in Drug Discovery and Development
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc25_ch96
4. Coursera Course: AI in Healthcare Specialization
   https://www.coursera.org/specializations/ai-healthcare

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
MDM-371-RAI: Image Processing and Computer Vision				
Teaching Scheme		**Credits**	**Examination Scheme**	
Theory	**NA**	**1**	**CCE**	**NA**
Practical	**2 Hours/Week**		**Practical**	**50 Marks**

Prerequisite:

Mathematics for Intelligent Systems (MDM-221-RAI)

Course Objectives:

The student will be able

1. To analyze robotic manipulator fundamentals including degrees of freedom, workspace, repeatability, and sensor integration (encoders, vision).
2. To apply image processing techniques for assembly line object analysis.
3. To implement geometric transformations and understand interpolation methods.
4. To develop camera calibration and 3D vision skills including monocular calibration, stereo disparity mapping, point cloud generation, and point cloud alignment (ICP).
5. To design and evaluate vision-based robotic systems using feature, machine learning, and Kalman filtering for grasping and real-time detection.

Course Outcomes:

After successful completion of the course, learner will be able to:

CO1: COMPUTE the degrees of freedom, workspace, and repeatability of a robotic arm using joint encoders and manual measurement techniques.

CO2: APPLY digitization, noise filtering, histogram analysis, contrast stretching, and histogram equalization to enhance images from a robotic assembly line.

CO3: COMPARE the performance of Sobel, Prewitt, and Canny edge detectors, and perform binary morphological operations on segmented object parts.

CO4: **IMPLEMENT** geometric transforms (translation, rotation, scaling, affine, perspective) and calibrate a monocular camera using Zhang's method.

CO5: DESIGN a vision-based grasping pipeline using Kalman filter, feature extraction (SIFT/SURF/ORB), SVM/YOLO models, and stereo depth estimation.

Laboratory Experiments/Assignments	
The student must complete the following activity for their Term Work Journal (Any 8)	
Practical 1	Compute the degrees of freedom, workspace, and repeatability of a robotic arm using joint encoders and manual measurement.
Practical 2	Set up a machine vision system and perform digitization, noise filtering, and histogram analysis on an object from a robotic assembly line.
Practical 3	Implement contrast stretching and histogram equalization, and compare mean, Gaussian, median, and bilateral filters on noisy images.
Practical 4	Apply Sobel, Prewitt, and Canny edge detectors, and perform binary morphological operations on segmented parts.
Practical 5	Write code for translation, rotation, scaling, affine, and perspective transforms with interpolation on a checkerboard image.
Practical 6	Calibrate a monocular camera using Zhang's method to extract intrinsic and extrinsic parameters and undistort images.
Practical 7	Implement a Kalman filter that fuses noisy encoder and camera measurements for a mobile robot's position estimation.
Practical 8	Compute a disparity map and depth estimation from a stereo image pair using epipolar geometry and block matching.
Practical 9	Generate a point cloud from a stereo camera, filter outliers, and align two-point clouds using the Iterative Closest Point algorithm.
Practical 10	Extract HOG features from a dataset of objects and train an SVM classifier for object detection in a robotic workspace.
Practical 11	Train or fine-tune a YOLO model for real-time detection of a robotic gripper, assembly parts, and obstacles in a video feed.
Practical 12	Extract SIFT, SURF, ORB, and Harris corner features from an object and implement a vision-based grasping pipeline using a robotic arm.

Learning Resources

Text Books:

1. Corke, P. (2023). *Robotics, vision and control: Fundamental algorithms in MATLAB (3rd Ed.).* Springer.
2. Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*, MIT Press
3. Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). *Introduction to autonomous mobile robots (2nd Ed.)*. MIT Press.
4. Lynch, K. M., & Park, F. C. (2017). *Modern robotics: Mechanics, planning, and control*. Cambridge University Press.

Reference Books:

1. Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing* (4th Ed.). Pearson.
2. Hartley, R., & Zisserman, A. (2004). *Multiple view geometry in computer vision (2nd Ed.)*. Cambridge University Press.
3. Siciliano, B., & Khatib, O. (2016). *Springer handbook of robotics (2nd Ed.)*. Springer International Publishing.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Computer Vision and Image Processing - Fundamentals and Applications
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_ee31

2. NPTEL Course: Computer Vision
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc19_cs58

3. NPTEL Course: Modern Computer Vision
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_ee74

Real World Assignment:

1. Compute the DOF, workspace volume, and positional repeatability of a robotic arm using joint encoder feedback and manual external measurements.
2. Set up a vision system to digitize an assembly line object, apply noise filtering, and analyze its intensity histogram.
3. Implement contrast stretching and histogram equalization, then quantitatively compare mean, Gaussian, median, and bilateral filters on noisy images.
4. Apply Sobel, Prewitt, and Canny edge detectors, and perform binary morphological operations to clean segmented part regions.
5. Implement translation, rotation, scaling, affine, and perspective transforms with different interpolation methods on a checkerboard image.
6. Calibrate a monocular camera using Zhang's method to extract intrinsic/extrinsic parameters and undistort lens distortions.
7. Implement a Kalman filter to fuse noisy encoder and camera measurements for improved mobile robot position estimation.
8. Compute a disparity map and depth estimates from a stereo image pair using epipolar geometry and block matching.

9. Generate a point cloud from a stereo camera, filter outliers, and align two-point clouds using the ICP algorithm.
10. Extract HOG features from a dataset of objects and train an SVM classifier for detection in a robotic workspace.
11. Train/fine-tune a YOLO model for real-time detection of a robotic gripper, assembly parts, and obstacles in a video feed.
12. Extract SIFT, SURF, ORB, and Harris corner features, then implement a vision-based grasping pipeline for a robotic arm.

Exemplary: Autonomous warehouse inventory scanning, Assembly line quality control, Robot arm pick-and-place, Obstacle detection & avoidance, 3D objects reconstruction & inspection, Stereo vision for autonomous driving, Mobile robot localization & navigation**,** Automated bin picking.

Savitribai Phule Pune University
Third Year of Robotics and Artificial Intelligence (2024 Pattern)

MDM-372-RAI: Industrial Automation

Teaching Scheme		Credits	Examination Scheme	
Theory	**NA**	**1**	**Term work**	**NA**
Practical	**2 Hours/Week**		**Practical**	**50 Marks**

Prerequisite:

Analog and Digital Electronics (PCC-203-RAI),
Sensors for Industrial Robotics (PCC-251-RAI)

Course Objectives:

The student will be able

1. To understand the fundamentals of industrial automation systems, PLC, SCADA, hydraulic and pneumatic systems.
2. To develop ladder logic programs using basic and advanced PLC instructions for industrial applications.
3. To interface PLC with sensors, actuators, hydraulic, pneumatic, VFD and motion control systems.
4. To demonstrate analog PLC programming and PID-based industrial process control applications.
5. To design and develop SCADA-based monitoring and control systems for industrial automation.

Course Outcomes:

After successful completion of the course, learner will be able to:

CO1: UNDERSTAND industrial automation architecture and PLC programming fundamentals.

CO2: DEVELOP ladder logic programs using timers, counters, comparison and mathematical instructions.

CO3: INTERFACE PLC with hydraulic, pneumatic, VFD and motion control systems.

CO4: APPLY analog PLC programming and PID control for industrial process applications.

CO5: DESIGN and **IMPLEMENT** SCADA systems for monitoring and automation applications.

Laboratory Experiments/Assignments	
The student shall complete the following activity as a Term Work Journal (Any 8).	
Practical 1	**Introduction to PLC and Ladder Programming** Study of PLC architecture, I/O modules and development of basic ladder logic programs using logic gates and Boolean operations.
Practical 2	**PLC Programming using Timers and Counters** Development of ladder logic programs using timers, Counters and cascading operations for industrial applications.
Practical 3	**PLC Programming for Industrial Automation Applications** Development of PLC programs for motor control, Interlocking, Alarm annunciator and process control applications.
Practical 4	**Advanced PLC Programming** Implementation of comparison, Mathematical, Sequencer, Shift register and data movement instructions using PLC.
Practical 5	**Analog PLC Programming and PID Control** Study and implementation of analog I/O modules and PID control using PLC for industrial process applications.
Practical 6	**Pneumatic System Components and Circuits** Study and simulation of pneumatic components and basic pneumatic circuits for industrial applications.
Practical 7	**Electro-Pneumatic Circuit Design** Development of electro-pneumatic circuits using relays, Solenoid valves and sensors for sequential operations.
Practical 8	**Hydraulic System Components and Circuits** Study and simulation of hydraulic components and hydraulic circuits for pressure and speed control applications.
Practical 9	**Electro-Hydraulic Systems and PLC Interfacing** Development of electro-hydraulic circuits and interfacing of hydraulic systems with PLC.
Practical 10	**SCADA System Development** Development of SCADA applications with graphical pages, Alarms, Trending and data logging features.

Learning Resources

Text Books:

1. Webb, J. W., & Reis, R. A. (2003). *Programmable logic controllers: Principles and applications* (5th Ed.). Prentice Hall of India.
2. Dunning, G. (2005). *Introduction to programmable logic controllers* (3rd Ed.). Delmar Thomson Learning.
3. Hackworth, J. R., & Hackworth, F. D. (2004). *Programmable logic controllers: Programming methods and applications*. Pearson Publication.
4. Esposito, A. (2009). *Fluid power with applications* (7th Ed.). Pearson Education.

Reference Books:

1. Petruzella, F. D. (2016). *Programmable logic controllers* (5th Ed.). McGraw-Hill Education.
2. Bolton, W. (2015). *Programmable logic controllers* (4th Ed.). Elsevier Newnes Publication.
3. Boyer, S. A. (2010). *SCADA: Supervisory control and data acquisition* (4th Ed.). ISA Publication.
4. Pati, S. (2018). *A text book on fluid mechanics and hydraulic machines*. Tata McGraw Hill.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Industrial Automation and Control
   https://onlinecourses.nptel.ac.in/noc25_ee84/preview
2. NPTEL Course: PLC Programming and Applications
   https://archive.nptel.ac.in/courses/108/105/108105064/
3. NPTEL Course: Hydraulic and Pneumatic Systems
   https://archive.nptel.ac.in/courses/112/107/112107144/
4. Coursera Course: Industrial Automation Engineering
   https://www.coursera.org/learn/industrial-automation-engineering

Real World Assignment:

1. Design and simulate a PLC-based automatic water level control system.
2. Develop a conveyor belt automation system using timers, counters and sensors.
3. Design an electro-pneumatic pick-and-place system for industrial applications.
4. Interface PLC with VFD for speed control of an induction motor.
5. Develop a SCADA-based monitoring system for temperature and level control.

6. Create a single-cycle electro-pneumatic circuit for a pick-and-place device that completes one full cycle per start pulse.
7. Design and simulate a hydraulic press control circuit.
8. Build a basic hydraulic circuit with a fixed-displacement pump and a 4/3 closed-center valve to raise and lower a press.
9. Develop a PLC program to sequence a hydraulic injection molding machine: clamp close, inject, hold, cool, clamp open using timers and limit switches.
10. Build a real-time trend graph in SCADA to display pump discharge pressure and flow rate over the last 10 minutes.
11. Create an industrial automation mini-project integrating PLC, SCADA and sensors.

Exemplary:

PLC programming and industrial automation systems, SCADA-based monitoring and supervisory control, Pneumatic and hydraulic automation systems, Motion control and drive applications, Smart factory and Industry 4.0 automation applications.

Savitribai Phule Pune University Third Year of Robotics and Artificial Intelligence (2024 Pattern)				
VSE-381-RAI: Autonomous Drones				
Teaching Scheme		**Credits**	**Examination Scheme**	
Theory	**NA**	**1**	**Term work**	**NA**
Practical	**2 Hours/Week**		**Practical**	**50 Marks**

Prerequisite:

Analog and Digital Electronics (PCC-203-RAI)

Course Objectives:

The student will be able

1. To introduce the fundamentals of unmanned aerial systems (UAS), drone platforms, and their applications.
2. To develop understanding of drone design, flight dynamics, propulsion systems, and control mechanisms.
3. To train students in autonomous flight control using sensors, PID controllers, GPS, and navigation algorithms.
4. To familiarize students with computer vision, object tracking, and rescue-based drone applications.
5. To demonstrate practical implementation of autonomous drone missions using simulation and real-world applications.

Course Outcomes:

After successful completion of the course, learner will be able to:

CO1: IDENTIFY the components, configurations, and working principles of autonomous drones and UAV systems.

CO2: DEVELOP drone models and **IMPLEMENT** stable autonomous flight using suitable control techniques.

CO3: APPLY PID-based position and navigation control algorithms for waypoint-based autonomous missions.

CO4: ANALYZE vision-based drone applications such as object tracking, surveillance, and rescue operations.

CO5: DESIGN and **EXECUTE** autonomous drone missions for real-world applications using simulation platforms and sensors.

Laboratory Experiments/Assignments	
The student shall complete the following activity as a Term Work Journal (Any 8).	
Practical 1	**Introduction to Autonomous Drones and UAV Platforms** Study the classification, components, configurations, and applications of autonomous drones and UAV platforms.
Practical 2	**3D Design and Modeling of Drone Frame** Design and develop a 3D model of a drone frame using CAD tools with suitable structural and aerodynamic features.
Practical 3	**Drone Assembly and Hardware Integration** Assemble drone hardware components and perform sensor calibration and system integration.
Practical 4	**Basic Flight Control and Stabilization** Perform basic flight operations and configure stabilization parameters for controlled drone movement.
Practical 5	**Autonomous Hovering using PID Controller** Develop and tune a PID-based control system to achieve stable autonomous hovering of the drone.
Practical 6	**Position Control and Local Navigation** Implement position control and local navigation techniques using onboard sensors and control algorithms.
Practical 7	**GPS-Based Autonomous Navigation** Develop waypoint-based autonomous navigation using GPS interfacing and route planning techniques.
Practical 8	**Object Detection and Tracking using Computer Vision** Implement computer vision techniques for object detection, tracking, and autonomous following operations.
Practical 9	**Rescue and Surveillance Mission Simulation** Simulate rescue and surveillance operations using autonomous navigation and vision techniques.
Practical 10	**Autonomous Drone Mission Planning and Demonstration** Design, integrate, and demonstrate a complete autonomous drone mission with navigation, sensing, and vision capabilities.

Learning Resources

Text Books:

1. Kilby, T., & Kilby, B. (2016). *Make: Getting started with drones* (1st Ed.). Maker Media Inc.
2. Sadraey, M. H. (2020). *Design of unmanned aerial systems* (1st Ed.). John Wiley & Sons.
3. Jha, A. R. (2020). *Theory, design, and applications of unmanned aerial vehicles* (1st Ed.). CRC Press.
4. Austin, R. (2010). *Unmanned aircraft systems: UAV design, development and deployment*. Wiley Publications.

Reference Books:

1. Beard, R., & McLain, T. (2012). *Small unmanned aircraft: Theory and practice*. Princeton University Press.
2. Bouabdallah, S. (2007). *Design and control of quadrotors with application to autonomous flying*. EPFL Press.
3. Raffo, G. V. (2015). *Modeling and control of mini-flying machines*. Springer Publications.
4. PX4 Development Team. (2023). *PX4 autopilot user guide and developer documentation*. PX4 Development Team.

MOOC / NPTEL/YouTube Links:

1. NPTEL Course: Drone Systems and Control
   https://nptel.ac.in/courses/101108661

2. NPTEL Course: Design of Fixed Wing Unmanned Aerial Vehicles
   https://nptel.ac.in/courses/101104073

3. Coursera Course: Autonomous Navigation for Flying Robots
   https://www.coursera.org/learn/robotics-flight

Real World Assignment:

1. Design and simulate a quadcopter frame for payload delivery applications.
2. Develop a drone stabilization system for autonomous hovering using PID control.
3. Model a foldable drone frame for backpack portability with hinge mechanisms and locking features for field deployment in remote areas.
4. Assemble a drone with an LED strip, buzzer, and GPS safety switch, programming failsafe triggers to activate them during low battery.

5. Execute a simple square pattern flight (10 m × 10 m) at constant 2 m altitude using a stabilized (Angle) mode.
6. Create a waypoint-based autonomous navigation system using GPS and IMU sensors.
7. Implement an object tracking drone using computer vision techniques.
8. Design a surveillance drone system for smart agriculture or border monitoring applications.
9. Develop a rescue drone capable of locating objects or humans within a defined area.
10. Perform a comparative analysis of commercial autopilot systems such as ArduPilot and PX4.

Exemplary:

Autonomous navigation and control, Drone-based surveillance systems, Precision agriculture using UAVs, Aerial mapping and inspection, Search and rescue drone applications, AI-based object tracking, Smart delivery drone systems.

Task Force for Curriculum Design and Development

Advisors & the Chairmen - Board of Studies

Dr. P. A. Patil	Dr. Kushare Prashant
Dr. Ravindra Navthar	Dr. Girish Modak
Dr. Amar Pandhare	Dr. Shiv Sai Dixit
Dr. Vishnu Wakchaure	Dr. Niranjan Kolhe
Dr. S. B. Zope	Dr. Ajit Bhosale
Dr. J. A. Hole	Dr. Narendra Deore
Dr. Mohan Khond	Dr. V. N. Chougule
Dr. A. B. Gawand	

Team Members for Course Design

Dr. Amol S. Bhanage

Prof. Nilesh C. Dhobale

Dr. Nilesh N. Satonkar

Prof. Shilpa N. Tambe

Compiled By

Dr. Amol S. Bhanage

Chairman

Dr. Pradeep A. Patil

Chairman– Board of Studies
Mechanical Engineering
Savitribai Phule Pune University, Pune

Dean

Dr. Raosaheb Latpate

Dean– Science and Technology

Savitribai Phule Pune University, Pune