IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

HIGH IMPACT PRACTICES (HIPS)
CORNERSTONE PROJECTS: ROBOTICS
INFORMATION PACKET
2025 - 2026

INSTITUTE OF AERONAUTICAL ENGINEERING

25 YEARS 2000 2025

I appreciate IARE students who are showing interest **Robotics with AI Cognition-(Robognition)** Project Program at the Institute of Aeronautical Engineering.

Cornerstone Projects are comprehensive, application-based projects typically undertaken by students in the second or third year of their academic programs. These projects serve experience that integrates the knowledge, skills, and competencies acquired throughout the curriculum. They are designed for a single semester with a small team upto two students to solve real-world problems through innovative, interdisciplinary.

This visionary initiative brings together undergraduate students, postgraduates, faculty members, and research scholars to form collaborative innovation groups focused on solving real-world problems through robotics, embedded systems, and artificial intelligence. This Robognition initiative empowers next-generation engineers to design, simulate, build, and control intelligent robotic systems that can perceive, decide, and act autonomously in dynamic environments. This program aims to integrate AI, machine learning, computer vision, sensor fusion, and control algorithms into the full lifecycle of robotic systems, spanning design, development, testing, deployment, and maintenance. Robotics plays a crucial role in designing, developing, and maintaining intelligent machines that perform physical tasks with precision, speed, and adaptability. Mechanical engineers provide the backbone of robotic systems through their expertise in mechanisms, dynamics, control systems, materials, manufacturing, and thermal management.

Core Goals of Intelligent Robotics Engineering (Robognition) Project Program:

- **Autonomous Navigation and Control**: Develop AI-driven control systems for mobile robots and drones to operate in structured and unstructured environments using SLAM, path planning, and reinforcement learning.
- **Perception and Sensing**: Integrate computer vision and sensor fusion for real-time object detection, localization, obstacle avoidance, and semantic understanding of environments.
- **Human-Robot Collaboration (HRC)**: Design collaborative robots (cobots) that can safely work alongside humans in manufacturing, healthcare, and domestic settings using natural language processing, gesture recognition, and force-feedback control.
- **Robotic Manipulation**: Enable robotic arms and grippers with intelligent force, torque, and vision feedback to handle objects of varying size, shape, and fragility with precision and adaptability.
- **Swarm Robotics and Multi-Agent Systems**: Implement decentralized control algorithms for fleets of robots to achieve tasks like area coverage, exploration, delivery, and surveillance in coordination.
- **Robotics for Social Good**: Design assistive and service robots for applications in elder care, education, agriculture, disaster relief, and smart infrastructure inspection.
- **Simulation and Digital Twin for Robotics**: Create physics-based simulation models and AI-enabled digital twins for robot testing, predictive maintenance, and mission rehearsal before real-world deployment.
- **Energy-Efficient and Sustainable Robotics**: Optimize power consumption in robotic systems using intelligent scheduling, efficient locomotion, and adaptive behavior planning for sustainable operation.

The Robognition project program addresses key SDGs by promoting clean energy, sustainable innovation, climate action, and community well-being.

Sustainability Development Goals (SDGs) for the Dept. of ME, IARE	
SDG #3	Ensure healthy lives and promote well-being for all at all ages
SDG #4	Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all
SDG #5	Achieve gender equality and empower all women and girls
SDG #6	Ensure availability and sustainable management of water and sanitation for all
SDG #7	Ensure access to affordable, reliable, sustainable and modern energy for all
SDG #8	Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all
SDG #9	Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation
SDG #10	Reduce inequality within and among countries
SDG #11	Make cities and human settlements inclusive, safe, resilient and sustainable
SDG #12	Ensure sustainable consumption and production patterns
SDG #13	Take urgent action to combat climate change and its impacts

The following research domains are recommended for **HIPs- Robognition** Projects, and selected students should find the research gap and frame the problem statements from any one of the themes below**.**

1. Autonomous Mobile Robot for Warehouse Automation Using AI *(SDG #8, SDG #9, SDG #12)*
2. Computer Vision-Based Object Detection and Grasping with Robotic Arm *(SDG #4, SDG #9)*
3. Reinforcement Learning for Robotic Path Planning *(SDG #7, SDG #9, SDG #11)*
4. Human-Robot Interaction System Using NLP and Gesture Recognition *(SDG #4, SDG #9, SDG #10)*
5. Energy-Efficient Robotic Arm Design for Industrial Use *(SDG #8, SDG #9, SDG #12)*
6. Digital Twin for Predictive Maintenance of Industrial Robots *(SDG #8, SDG #9, SDG #12)*
7. Swarm Robotics for Agricultural Monitoring and Pest Detection *(SDG #2, SDG #6, SDG #12, SDG #15)*
8. AI-Enabled Social Robots for Education and Elder Care *(SDG #3, SDG #4, SDG #5, SDG #10)*
9. Self-Balancing Two-Wheel Robot with Dynamic Stabilization *(SDG #7, SDG #9, SDG #13)*
10. Gesture-Controlled Robotic Arm for Disabled Individuals *(SDG #3, SDG #9, SDG #10)*

In order to participate in **Robognition** Projects, you must formally apply and be accepted by the project coordinator. To proceed, please mail to the project coordinator, Dr. B Vijaya Krishna, Assistant Professor, Mechanical Engineering, Email Id: b.vijaya@iare.ac.in. This will bring up all available open positions tagged as **Robognition** projects. When submitting a project document and an updated résumé, include a statement regarding why you are interested in working with the team to which you are applying. Please note that participation by the **Robognition** project team requires registration for the accompanying research statement from any of the specified domains. More information will be provided to all selected **Robognition** project applicants who have been offered a position. If you have any questions about a particular team, please contact the team's faculty mentor(s). We encourage you to contemplate this fascinating new opportunity. We look forward to receiving your application submission.

Autonomous Mobile Robot for Warehouse Automation Using AI

Dr. Ch Sandeep, Associate Professor & Head, Dept. of ME - Faculty Mentor

GOALS

This project aims to design and develop an Autonomous Mobile Robot (AMR) system powered by AI for automating warehouse operations such as inventory transport, object detection, obstacle avoidance, and route planning. The use of AMRs addresses the growing demand for flexible, scalable, and intelligent logistics solutions in e-commerce, manufacturing, and retail warehouses.

Traditional warehouse automation relies on fixed-path AGVs (Automated Guided Vehicles) or conveyor systems. In contrast, AMRs offer real-time decision-making capabilities using AI, allowing them to dynamically respond to changing environments, avoid collisions, and efficiently navigate to target locations.

Key Goals:

- Design a mobile robot platform capable of navigating warehouse layouts autonomously.
- Implement computer vision for shelf/item detection, barcode/QR scanning, and obstacle identification.
- Develop AI-based path planning and dynamic route optimization algorithms.
- Integrate sensor fusion (LIDAR, IMU, camera) for accurate localization and mapping (SLAM).
- Enable real-time decision-making and fleet coordination for task allocation.
- Improve logistics efficiency, reduce human workload, and enhance safety in warehouse environments.

METHODS & TECHNOLOGIES

Methods (Process & Approach):

- **Robot Hardware & Design**
  Select or design a mobile robot base with wheels, chassis, and power system. Integrate sensors (LIDAR, ultrasonic, cameras, IMU, encoders).
- **Perception & Environment Mapping**
  Implement Simultaneous Localization and Mapping (SLAM) using LIDAR/vision-based techniques. Use CNN models for object and obstacle detection.
- **Path Planning and Navigation**
  Apply A*, D*, or RRT algorithms for static and dynamic path planning. Implement PID/Model Predictive Control for motion control.
- **AI-Driven Task Management**
  Use reinforcement learning or optimization heuristics to allocate tasks and optimize delivery routes. Apply ML for dynamic re-routing and collision avoidance in real-time.
- **Warehouse Interaction**
  Detect and scan items using computer vision. Navigate to shelf positions, pick/place payloads using simple robotic arms or mechanical lift mechanisms.
- **Testing & Simulation**
  Use Gazebo/ROS or Webots for simulation and testing of navigation and control algorithms.

- **Deployment & Visualization**
  Visualize routes, metrics, and robot state using RViz, ROS dashboards, or IoT interfaces.

Technologies Used:

- **Hardware Platforms:** Raspberry Pi, NVIDIA Jetson Nano, Arduino, TurtleBot, LIDAR (RPLidar), Depth Camera (Intel RealSense), Servo motors
- **Robotics Frameworks:** ROS (Robot Operating System), OpenCV, MoveIt
- **AI/ML Frameworks:** TensorFlow, PyTorch, Scikit-learn, Keras
- **Algorithms:** DNN, YOLOv5, SLAM (GMapping, Hector SLAM), RRT*, A*, PPO
- **Programming Languages:** Python, C++, ROS scripting

MAJORS & AREAS OF INTEREST

This project is ideal for students from Mechanical Engineering, Mechatronics, Electronics and Communication Engineering (ECE), Electrical Engineering, and Computer Science who are interested in autonomous robotics, AI-based navigation, and smart logistics systems. Key focus areas include:

- **Mobile Robotics Design & Prototyping** – Building robust, scalable mobile robot platforms for warehouse operations.
- **Computer Vision & Object Detection** – Implementing AI-driven vision systems for shelf/item detection, barcode/QR code scanning, and environment recognition.
- **SLAM (Simultaneous Localization and Mapping)** – Integrating LIDAR, IMU, and camera data for real-time localization, mapping, and path planning.
- **AI & Machine Learning for Navigation** – Developing algorithms for dynamic route optimization, obstacle avoidance, and intelligent decision-making.
- **Sensor Fusion & Control Systems** – Combining multiple sensors for precise navigation and fleet coordination.
- **Warehouse Automation & Logistics Optimization** – Streamlining inventory management, reducing manual labor, and improving operational safety.
- **Edge Computing & IoT Integration** – Enabling low-latency processing and cloud connectivity for fleet coordination and analytics.

MENTOR CONTACT INFORMATION

Dr. Ch Sandeep
Email: ch.sandeep@iare.ac.in

Computer Vision-Based Object Detection and Grasping with Robotic Arm

Dr. B Vijaya Krishna, Assistant Professor, Dept. of ME - Faculty Mentor

GOAL

This project aims to develop a computer vision-guided robotic arm system capable of accurately detecting, classifying, and grasping various objects in dynamic and unstructured environments. The integration of deep learning-based vision algorithms with robotic manipulation will enable intelligent pick-and-place tasks useful in applications such as warehouse automation, smart manufacturing, waste sorting, and service robotics.

Traditional robotic arms rely on pre-programmed coordinates or fixed environments. This limits their adaptability in real-world scenarios. By combining AI-based object detection (e.g., YOLO, SSD, or Faster R-CNN) and pose estimation with motion planning algorithms, the robotic arm can autonomously detect and manipulate target objects in cluttered or changing environments.

Key Goals:

- Develop deep learning models for object detection and classification using real-time camera feed.
- Implement 2D/3D object localization and pose estimation for robotic manipulation.
- Integrate vision system with robotic control to perform real-time grasp planning.
- Enable autonomous pick-and-place of objects with minimal human intervention.
- Optimize grasp quality and reliability using reinforcement learning or heuristic algorithms.

METHODS & TECHNOLOGIES

Methods (Process & Approach):

- **Data Collection and Annotation**
  Gather datasets of objects under different lighting, orientation, and occlusion. Annotate images for training object detection and classification models.
- **Model Development**
  Train object detection models (e.g., YOLOv8, SSD, Faster R-CNN) for real-time inference. Use OpenCV and MediaPipe for contour detection and landmark extraction. Integrate depth sensing (e.g., Intel RealSense, LiDAR) for 3D object localization.
- **Grasp Planning and Robotic Control**
  Implement grasp planning algorithms like GQCNN or heuristic-based grasp pose generation. Use motion planning libraries (e.g., MoveIt, RRT*) for path planning and collision avoidance. Interface the model with a robotic arm (e.g., Dobot, UR5, Niryo One) using ROS or custom Python APIs.
- **Testing and Evaluation**
  Assess detection accuracy (mAP), grasp success rate, and response latency. Conduct trials on various object shapes, sizes, and positions in semi-structured environments.
- **Deployment**
  Build a GUI or dashboard for system control and monitoring. Deploy models on GPU-equipped systems or edge devices like NVIDIA Jetson Nano or TX2.

Technologies Used:

- **Deep Learning Frameworks**: PyTorch, TensorFlow, Keras, Detectron2, YOLOv8
- **Computer Vision Tools**: OpenCV, MediaPipe, RealSense SDK, Open3D
- **Robotic Platforms**: UR5, Dobot Magician, Niryo One, Arduino-based arm
- **Middleware & Control**: ROS, MoveIt, Gazebo, RViz
- **Programming Languages**: Python, C++

MAJORS & AREAS OF INTEREST

This project is ideal for I-driven robotics, computer vision, and intelligent manipulation systems. Key focus areas include:

- **Deep Learning for Object Detection** – Using YOLO, SSD, or Faster R-CNN for real-time object recognition and classification.
- **2D/3D Pose Estimation** – Implementing camera-based localization techniques to determine object position and orientation for grasping.
- **Robotic Manipulation & Motion Planning** – Integrating kinematic modeling and motion control algorithms for pick-and-place tasks.
- **Vision-Based Control Systems** – Linking AI vision modules with robotic arm controllers for real-time decision-making.
- **Reinforcement Learning & Grasp Optimization** – Improving grasp success rate using adaptive learning techniques or heuristic-based strategies.
- **Hardware-Software Integration** – Interfacing cameras, sensors, and actuators with robotic platforms using ROS (Robot Operating System).
- **Applications in Smart Manufacturing & Logistics** – Automating warehouse operations, assembly lines, or waste sorting through intelligent object handling.

MENTOR CONTACT INFORMATION

Dr. B Vijaya Krishna
Email: b.vijayakrishna@iare.ac.in

Reinforcement Learning for Robotic Path Planning

Dr. G Hima Bindu, Assistant Professor - Dept. of ME - Faculty Mentor

GOAL

This project aims to develop an autonomous robotic navigation system that uses Reinforcement Learning (RL) to learn optimal path planning strategies in complex, dynamic environments. Unlike traditional path planning algorithms, reinforcement learning allows robots to learn from experience and adapt to uncertain, partially known, or changing scenarios without explicit programming.

By integrating RL algorithms with robotic control frameworks and simulation environments, the robot can learn policies for obstacle avoidance, goal-oriented navigation, and energy-efficient movement. This approach is highly applicable in indoor navigation, warehouse automation, planetary exploration, and mobile delivery systems.

Key Goals:

- Develop RL-based algorithms (e.g., Q-learning, DQN, PPO) for real-time path planning.
- Train agents in simulation to navigate toward goals while avoiding dynamic and static obstacles.
- Enable transfer of learned policies from simulation to real-world robotic systems.
- Evaluate navigation performance based on path optimality, success rate, and computational efficiency.
- Explore hybrid methods combining traditional planners with RL for robust decision-making.

METHODS & TECHNOLOGIES

Methods (Process & Approach):

- **Environment Setup and Simulation**
  Design custom 2D/3D environments using Gazebo, PyBullet, or Unity ML-Agents. Model robots with differential drive, LiDAR, and vision sensors.
- **RL Algorithm Development**
  Use model-free RL algorithms like Deep Q-Network (DQN), Proximal Policy Optimization (PPO), or Soft Actor-Critic (SAC). Define reward functions based on distance to goal, obstacle collisions, and path smoothness.
- **Training and Optimization**
  Train RL agents using policy gradient or value-based methods in simulation. Perform hyperparameter tuning and implement experience replay and exploration strategies (e.g., ε-greedy, entropy regularization).
- **Sim-to-Real Transfer**
  Apply domain randomization or fine-tuning to ensure learned policy generalizes to real-world settings. Deploy policy on physical robots (e.g., TurtleBot, JetBot) using ROS interface.
- **Evaluation and Visualization**
  Use metrics like path length, time-to-goal, collision count, and reward per episode. Visualize training progress, heatmaps, and robot trajectories using RViz or Matplotlib.

Technologies Used

- **Reinforcement Learning Frameworks**: Stable-Baselines3, RLlib, TensorFlow Agents, OpenAI Gym
- **Simulation Environments**: Gazebo, Webots, Unity ML-Agents, PyBullet
- **Robotic Platforms**: TurtleBot3, JetBot, ROSbots, custom wheeled robots
- **Middleware & Tools**: ROS (Robot Operating System), RViz, MoveBase
- **Programming Languages**: Python, C++, ROS launch files

MAJORS & AREAS OF INTEREST

This project is ideal for AI-driven navigation, autonomous robotics, and machine learning. Key focus areas include:

- **Reinforcement Learning Algorithms** – Implementing Q-Learning, Deep Q-Networks (DQN), PPO, or A3C for path optimization.
- **Robotic Navigation & Control** – Designing control strategies for autonomous navigation in dynamic and uncertain environments.
- **Simulation Environments for Training** – Using tools like Gazebo, ROS, or PyBullet to simulate and train robotic agents.
- **Policy Transfer from Simulation to Real Robots** – Bridging the sim-to-real gap for deploying trained RL policies on physical robots.
- **Obstacle Avoidance & Goal-Oriented Planning** – Developing intelligent strategies to avoid collisions while ensuring optimal paths.
- **Hybrid Path Planning Approaches** – Combining traditional algorithms (A*, Dijkstra) with RL for robust and adaptive decision-making.
- **Applications in Autonomous Systems** – Implementing solutions for warehouse robots, planetary rovers, mobile delivery bots, and indoor navigation.

MENTOR CONTACT INFORMATION

Dr. G Hima Bindu
Email: dr.ghimabindu @iare.ac.in

Human-Robot Interaction System Using NLP and Gesture Recognition

Dr. G Sarat Raju, Assistant Professor - Dept. of ME - Faculty Mentor

GOAL

This project aims to develop an intelligent Human-Robot Interaction (HRI) system that combines Natural Language Processing (NLP) with gesture recognition to enable seamless, intuitive, and multimodal communication between humans and robots. As robotics increasingly integrates into homes, healthcare, manufacturing, and service sectors, natural and adaptive interaction becomes essential for usability and user acceptance.

The proposed system allows users to issue commands or interact with robots using voice-based language as well as hand gestures, enabling accessibility, flexibility, and robustness even in noisy or vision-constrained environments.

Key Goals:

- Design an HRI system that can understand human commands via both speech and hand gestures.
- Integrate NLP techniques for intent recognition, command parsing, and conversational responses.
- Employ computer vision and deep learning for robust real-time gesture recognition.
- Fuse multimodal inputs to enhance interaction accuracy and context understanding.
- Deploy the system on a robotic platform (e.g., robotic arm, mobile robot) for real-world task execution.

METHODS & TECHNOLOGIES

Methods (Process & Approach):

- **Speech-Based NLP Interaction**
  Capture voice input using microphones or headsets. Apply speech-to-text conversion using models like Google Speech API or Mozilla DeepSpeech. Use NLP (e.g., BERT, spaCy) for intent detection and command extraction.
- **Gesture Recognition**
  Capture hand/body gestures via RGB camera or depth sensors (e.g., Kinect, RealSense). Use pose estimation libraries (e.g., MediaPipe, OpenPose) for hand/keypoint tracking. Train deep learning models (e.g., CNN, LSTM) to classify static and dynamic gestures.
- **Multimodal Fusion**
  Implement rule-based or learning-based decision models to combine NLP and gesture inputs. Resolve conflicting or ambiguous commands using context-aware logic.
- **Robot Integration**
  Connect processed commands to robot control systems using ROS or microcontroller interfaces. Enable robot responses (movement, speech, feedback) based on user interaction.
- **User Interface & Feedback**
  Build a dashboard for visual feedback, control logs, and system diagnostics. Incorporate speech synthesis (e.g., pyttsx3, gTTS) for conversational robot responses.

Technologies Used:

- **NLP & Speech Processing:** NLTK, spaCy, BERT, Rasa, Google Speech API, Mozilla DeepSpeech
- **Computer Vision & Gesture Recognition:** OpenCV, MediaPipe, OpenPose, Tensor-Flow, PyTorch
- **Deep Learning Models:** CNNs for image-based gesture classification, LSTM/RNN for gesture sequences
- **Hardware:** RGB/depth camera, microphones, microcontrollers, servo motors, robotic arm or mobile base
- **Programming Languages:** Python, C++, JavaScript (for interface components)

MAJORS & AREAS OF INTEREST

This project is ideal for human-robot interaction (HRI), natural language processing, and gesture-based interfaces. Key focus areas include:

- **Natural Language Processing (NLP)** – Developing speech-to-text, intent recognition, and conversational AI for robot commands.
- **Gesture Recognition Systems** – Using vision-based (camera/depth sensors) or wearable sensor-based approaches for hand or body gesture detection.
- **Multimodal HRI Interfaces** – Combining voice, gesture, and contextual cues to create seamless human-robot communication.
- **Computer Vision & Deep Learning** – Applying CNNs, LSTMs, or transformer models for gesture classification and tracking.
- **Embedded Systems & Robotics Integration** – Connecting HRI modules with robotic arms, mobile robots, or assistive devices.
- **User Experience & Ergonomic Design** – Ensuring the system is intuitive, responsive, and adaptive for diverse users.
- **Applications in Service Robotics** – Leveraging HRI for collaborative robots (cobots), healthcare, assistive living, and smart home automation.

MENTOR CONTACT INFORMATION

Dr. G Sarat Raju
Email: g.saratraju@iare.ac.in

Energy-Efficient Robotic Arm Design for Industrial Use

Dr. Ch Sandeep, Associate Professor & Head - Dept. of ME - Faculty Mentor

GOAL

This project focuses on the design and development of an energy-efficient robotic arm tailored for industrial applications such as pick-and-place, assembly, welding, and material handling. In modern manufacturing environments, robotic arms contribute significantly to productivity but can also consume large amounts of energy, especially during repetitive and continuous operations.

The aim is to reduce the energy footprint of robotic arms by optimizing mechanical structure, control algorithms, and actuation strategies—without compromising performance, precision, or payload capacity.

Key Goals:

- Design a lightweight, structurally optimized robotic arm to reduce energy consumption.
- Develop control strategies that minimize actuator usage through intelligent motion planning.
- Integrate low-power components and energy-saving mechanisms such as regenerative braking and adaptive torque control.
- Evaluate the trade-offs between performance, energy usage, and task accuracy.
- Support sustainable manufacturing practices by reducing operational energy demand.

METHODS & TECHNOLOGIES

Methods (Process & Approach):

- **Mechanical Design & Structural Optimization**
  Use CAD tools (e.g., SolidWorks, Fusion 360) for modeling robotic arm components. Perform finite element analysis (FEA) to reduce weight while maintaining rigidity. Optimize joint placement, link dimensions, and material selection for minimal energy usage.
- **Kinematic and Dynamic Modeling**
  Use Denavit–Hartenberg (DH) parameters and inverse kinematics for movement planning. Develop dynamic models to analyze torque and force requirements at each joint.
- **Energy-Aware Control Algorithms**
  Implement path-planning strategies that minimize energy, such as jerk-limited or smooth trajectories. Use energy-optimal joint trajectories computed via optimization algorithms (e.g., gradient descent, genetic algorithms). Introduce sleep-mode or standby states during idle operation.
- **Component Integration & Actuation**
  Select low-power servo or stepper motors with high efficiency. Integrate sensors (e.g., encoders, current sensors) to monitor energy consumption in real time. Explore energy-recovery methods such as regenerative braking on deceleration.

- **Testing, Evaluation & Iteration**
  Measure energy consumption using power analyzers. Evaluate task performance metrics: energy per cycle, task completion time, and repeatability. Iteratively refine design and control software based on performance data.

Technologies Used:

- **Design & Simulation Tools:** SolidWorks, Fusion 360, ANSYS, MATLAB SimMechanics
- **Control & Programming:** Arduino, Raspberry Pi, STM32, ROS (Robot Operating System)
- **Motor Control:** PWM drivers, low-voltage servo/stepper motors, PID control
- **Optimization & AI:** MATLAB, SciPy, genetic algorithms, reinforcement learning (for adaptive control)
- **Sensors & Power Monitoring:** Load cells, encoders, current/voltage sensors, INA219, ACS712
- **Programming Languages:** Python, C/C++, MATLAB

MAJORS & AREAS OF INTEREST

This project is ideal for industrial robotics, energy optimization, and sustainable automation systems. Key focus areas include:

- **Robotic Arm Design & Structural Optimization** – Creating lightweight, high-strength robotic arm structures to minimize energy usage.
- **Energy-Efficient Actuation** – Developing control strategies that reduce actuator power consumption via optimal torque distribution and regenerative braking.
- **Motion Planning & Control Algorithms** – Implementing energy-optimized trajectories and intelligent motion planning.
- **Simulation & Analysis** – Using tools like MATLAB, Simulink, or ROS to simulate performance vs. energy trade-offs.
- **Embedded Control & Low-Power Electronics** – Integrating efficient drivers, motors, and controllers to reduce energy demand.
- **Sustainable Manufacturing** – Applying green engineering principles to lower the carbon footprint of robotic systems.
- **Applications in Industrial Automation** – Enhancing pick-and-place, welding, or assembly processes with energy-conscious robotic solutions.

MENTOR CONTACT INFORMATION

Dr. Ch Sandeep
Email: ch.sandeep@iare.ac.in

Digital Twin for Predictive Maintenance of Industrial Robots

Dr. G Hima Bindu, Assistant Professor - Dept. of ME - Faculty Mentor

GOAL

This project aims to develop a Digital Twin system for predictive maintenance of industrial robots, enabling real-time monitoring, virtual simulation, and early fault detection. Industrial robots are vital assets in smart factories, but unplanned downtime due to mechanical wear, sensor failures, or actuator issues can severely impact productivity and costs.

By integrating physical robot data with a virtual replica (digital twin), the system can anticipate potential failures, optimize maintenance schedules, and extend equipment lifespan. The digital twin acts as a dynamic, data-driven model that mirrors the real-time behavior and health of robotic systems.

Key Goals:

- Develop a real-time digital replica of industrial robotic systems using sensor and operational data.
- Monitor key performance indicators (KPIs) such as joint torque, vibration, temperature, and motor current.
- Predict component degradation and anomalies using machine learning techniques.
- Optimize maintenance schedules to reduce unplanned downtimes and enhance equipment availability.
- Provide actionable insights through a user-friendly dashboard for maintenance personnel and operators.

METHODS & TECHNOLOGIES

Methods (Process & Approach):

- **Data Acquisition and Integration**
  Collect real-time sensor data from the robot (e.g., joint angles, current, temperature, vibration). Use edge devices (e.g., Raspberry Pi, NVIDIA Jetson) for local data preprocessing. Create a digital model that syncs with the physical robot using kinematics and dynamics.
- **Digital Twin Development**
  Model the robot in simulation environments like Gazebo, Unity, or MATLAB Simulink. Continuously update the digital twin using live telemetry via ROS or MQTT protocols.
- **Predictive Analytics and Fault Detection**
  Train machine learning models (e.g., SVM, Random Forest, LSTM) to predict wear, failure, or drift in performance. Use anomaly detection algorithms to flag deviations from normal operating patterns.
- **Simulation and Scenario Testing**
  Simulate various fault conditions and operational scenarios in the digital twin. Validate predicted outcomes with historical maintenance logs or controlled experiments.
- **Visualization and Alert System**
  Develop an intuitive dashboard for real-time visualization of health indicators and predictive alerts. Send notifications or trigger maintenance tasks automatically via IoT platforms or mobile apps.

Technologies Used:

- **Digital Twin Platforms:** MATLAB Simulink, Unity3D, Gazebo, ROS
- **Sensor Interfaces:** IMU, temperature sensors, current sensors, encoders
- **Edge & IoT Devices:** Raspberry Pi, NVIDIA Jetson Nano, ESP32
- **Data Analytics & AI:** Python, Scikit-learn, TensorFlow, Keras, PyCaret
- **Communication Protocols:** MQTT, OPC UA, ROS Topics, REST APIs
- **Visualization & UI:** Grafana, Streamlit, Node-RED, Power BI
- **Programming Languages:** Python, C++, MATLAB

MAJORS & AREAS OF INTEREST

This project is ideal for Industry 4.0, predictive maintenance, and robotics health monitoring. Key focus areas include:

- **Digital Twin Development** – Creating virtual replicas of industrial robots that mirror real-time physical operations.
- **Predictive Analytics & Machine Learning** – Applying ML algorithms to predict faults, wear, and anomalies in robotic components.
- **Sensor Integration & Data Acquisition** – Using vibration, temperature, torque, and current sensors for condition monitoring.
- **Real-Time Monitoring & Visualization** – Designing dashboards and visualization tools for health metrics and failure predictions.
- **IoT & Edge Computing for Robotics** – Connecting sensors and digital twins through IoT platforms for low-latency data processing.
- **Maintenance Optimization** – Reducing unplanned downtime through intelligent scheduling and data-driven insights.
- **Industrial Applications of Digital Twins** – Leveraging this technology for automation systems, smart factories, and robotics lifecycle management.

MENTOR CONTACT INFORMATION

Dr. G Hima Bindu
Email: dr.ghimabindu@iare.ac.in

Swarm Robotics for Agricultural Monitoring and Pest Detection

Dr. G Sarath Raju, Assistant Professor - Dept. of ME - Faculty Mentor

GOAL

This project aims to develop a swarm robotics system for real-time agricultural monitoring and pest detection, leveraging the collaborative behavior of multiple low-cost autonomous robots. Agriculture faces challenges such as crop health monitoring, early pest infestation detection, and the need for precision farming to improve yields and sustainability.

Swarm robotics—drawing inspiration from biological swarms (e.g., ants, bees)—provides a scalable and decentralized solution where multiple robots coordinate autonomously to cover large farmland areas. Equipped with sensors and cameras, these robots can detect anomalies such as pest-affected regions, nutrient deficiencies, or irrigation issues with high coverage and resilience.

Key Goals:

- Design and deploy a team of autonomous ground or aerial robots for large-scale farm surveillance.
- Implement decentralized swarm intelligence algorithms for navigation and area coverage.
- Enable real-time detection of pests and crop stress using onboard computer vision and environmental sensors.
- Ensure robustness, scalability, and cost-effectiveness of the robotic swarm for practical agricultural use.
- Support sustainable farming through targeted interventions and data-driven decision-making.

METHODS & TECHNOLOGIES

Methods (Process & Approach):

- **Swarm Behavior Design**
  Develop algorithms inspired by ant colony optimization (ACO), flocking, or particle swarm optimization (PSO). Enable decentralized communication and task allocation (e.g., leaderless coordination, area division).
- **Robotic Platform Development**
  Design ground (UGV) or aerial (UAV) robots equipped with cameras, GPS, and environmental sensors. Use Arduino/Raspberry Pi or ESP32 for onboard processing and control.
- **Agricultural Monitoring & Pest Detection**
  Capture high-resolution images and sensor data (temperature, humidity, soil moisture). Use deep learning models (e.g., CNN, YOLO) to detect pest patterns, leaf discoloration, or crop stress. Classify plant health status and generate pest infestation heatmaps
- **Communication and Data Fusion**
  Implement wireless communication between swarm units using Zigbee, LoRa, or Wi-Fi mesh. Aggregate and analyze data centrally or locally using edge AI on each robot.
- **Visualization & Farmer Interface**
  Build a mobile or web-based dashboard for real-time alerts, map-based visualization, and intervention planning. Allow users to query crop health history and pest spread over time.

Technologies Used:

- **Swarm & Control Algorithms:** Ant Colony Optimization, Boids model, PSO, FSMs
- **AI & Vision Models:** OpenCV, TensorFlow, YOLOv5, MobileNet, CNN-based classifiers
- **Robotics Hardware:** Arduino, Raspberry Pi, GPS modules, DHT11, soil sensors, ESP32, drone/UAV kits
- **Communication Protocols:** Zigbee, LoRa, Wi-Fi mesh, MQTT
- **Programming Languages:** Python, C/C++, Arduino IDE

MAJORS & AREAS OF INTEREST

This project is ideal for autonomous robotics, swarm intelligence, and precision agriculture. Key focus areas include:

- **Swarm Intelligence & Multi-Robot Coordination** – Designing decentralized algorithms inspired by biological swarms for large-area coverage and cooperative tasks.
- **Autonomous Ground and Aerial Robots** – Developing UAVs/drones or ground robots with autonomous navigation and obstacle avoidance.
- **Computer Vision & AI for Agriculture** – Detecting pests, crop stress, and anomalies using deep learning models and onboard cameras.
- **Sensor Integration for Smart Farming** – Using environmental sensors (humidity, temperature, soil health) for data-driven analysis.
- **IoT & Cloud Analytics** – Transmitting data to cloud platforms for real-time monitoring, decision-making, and visualization.
- **Energy-Efficient Robotic Design** – Optimizing power usage for long-duration field operations.
- **Sustainable Agriculture Applications** – Enabling targeted pest control, irrigation, and fertilization through intelligent robotic interventions.

MENTOR CONTACT INFORMATION

Dr. G Sarat Raju
Email: g.saratraju@iare.ac.in

AI-Enabled Social Robots for Education and Elder Care

Dr. V Mahider Reddy, Assistant Professor - Dept. of ME - Faculty Mentor

GOALS

This project aims to develop AI-powered social robots capable of engaging in empathetic, interactive, and context-aware communication with humans in educational and elder care environments. As aging populations and digital learning expand, social robots have emerged as promising tools to assist elderly individuals in daily living and support learners through personalized tutoring.

By integrating natural language processing (NLP), emotion recognition, and adaptive behavior models, the project focuses on building human-centric robots that foster companionship, mental stimulation, and tailored support, thus addressing both cognitive and emotional needs.

Key Goals:

- Design AI-driven social robots with natural language understanding and speech generation capabilities.
- Enable emotion recognition through voice tone, facial expressions, and behavioral cues.
- Create personalized interaction strategies for tutoring children and supporting elder care routines.
- Incorporate safety, accessibility, and user trust as core design principles.
- Evaluate robot effectiveness in improving user engagement, emotional well-being, and learning outcomes.

METHODS & TECHNOLOGIES

Methods (Process & Approach):

- **Human-Robot Interaction (HRI) Design**
  Develop interaction models that simulate social cues like eye contact, gestures, and speech modulation. Create dialogue management systems for task-based and casual conversation.
- **Natural Language Processing & Dialogue Systems**
  Use NLP models (e.g., GPT, BERT, Rasa) for intent detection and context tracking. Enable multi-language support and age-appropriate language simplification.
- **Emotion & Behavior Recognition**
  Use computer vision to detect facial expressions and gestures via camera input. Analyze speech prosody and body language for emotional inference using ML models (e.g., SVM, LSTM).
- **Task-Specific Functional Modules**
  Education: Interactive storytelling, quiz games, vocabulary teaching, and learning progress tracking. Elder Care: Medication reminders, wellness checks, guided exercises, and conversation-based companionship.
- **Robot Platform Development**
  Integrate sensors, microphones, speakers, and displays into humanoid or semi-humanoid robots. Program robot behavior through ROS or custom microcontroller-based firmware.
- **User Feedback & Adaptation**
  Continuously adapt robot behavior using reinforcement learning or feedback-based tuning. Collect user satisfaction, engagement metrics, and psychological well-being data for iterative improvement.

Technologies Used:

- **AI & NLP Models:** OpenAI GPT, BERT, spaCy, Rasa, Dialogflow
- **Computer Vision & Emotion Detection:** OpenCV, MediaPipe, Affectiva, FER+ dataset
- **Speech Processing:** Google Text-to-Speech, Mozilla DeepSpeech, pyttsx3, Whisper
- **Robotics Frameworks:** ROS, Arduino, Raspberry Pi, custom humanoid kits
- **Behavior & Control Systems:** Finite State Machines (FSM), RL-based behavior adaptation
- **Programming Languages:** Python, C++, JavaScript

MAJORS & AREAS OF INTEREST

This project is ideal for human-robot interaction, AI-driven social robotics, and assistive technologies. Key focus areas include:

- **Natural Language Processing (NLP) & Conversational AI** – Developing intelligent dialogue systems for personalized, human-like communication.
- **Emotion Recognition & Affective Computing** – Using voice, facial expressions, and behavioral cues to detect and respond to user emotions.
- **Human-Robot Interaction (HRI)** – Designing intuitive and empathetic interactions that enhance user engagement and trust.
- **AI Behavior Modeling** – Implementing adaptive behavior algorithms that tailor robot responses based on user needs and preferences.
- **Embedded Systems & Robotics Design** – Building safe, accessible, and ergonomically designed robotic platforms for social applications.
- **Applications in Education & Elder Care** – Enabling robots to act as tutors, companions, or assistive devices for cognitive and emotional support.
- **Ethical & Trustworthy AI** – Ensuring privacy, inclusivity, and user safety while interacting with social robots.

MENTOR CONTACT INFORMATION

Dr. V Mahider Reddy
Email: v.mahidharreddy@iare.ac.in

Self-Balancing Two-Wheel Robot with Dynamic Stabilization

Dr. B Vijaya Krishna, Assistant Professor - Faculty Mentor

GOALS

This project aims to design and build a **self-balancing two-wheel robot** that maintains its upright position using real-time sensor feedback and control algorithms. Inspired by the principles of inverted pendulum systems and similar to Segway-type mobility platforms, the robot employs dynamic stabilization to correct its orientation and respond to external disturbances.

The project serves as a hands-on platform to explore embedded systems, control theory, sensor fusion, and robotics — offering practical applications in autonomous navigation, personal transport systems, and mobile robotics research.

Key Goals:

- Develop a two-wheeled robot capable of balancing itself dynamically using sensor data.
- Implement closed-loop feedback control using PID or advanced control techniques.
- Integrate IMU sensors for tilt angle estimation and disturbance compensation.
- Enable remote or autonomous movement while maintaining stability.
- Explore real-time tuning, motion planning, and obstacle avoidance features.

METHODS & TECHNOLOGIES

Methods (Process & Approach):

- **Mechanical Design & Hardware Integration**
  Design a lightweight and symmetrical chassis using CAD tools (e.g., Fusion 360, Solid-Works).
  Mount motors, wheels, and the control board to minimize center of mass height and ensure structural stability.
- **Sensor Integration & Signal Processing**
  Use an Inertial Measurement Unit (IMU) such as MPU6050 to measure pitch angle and angular velocity.
  Apply sensor fusion algorithms (e.g., complementary filter or Kalman filter) to reduce noise and improve stability.
- **Control System Implementation**
  Implement a PID controller to maintain balance by adjusting motor speeds based on tilt error.
  Tune proportional, integral, and derivative gains to achieve smooth stabilization.
- **Motor Driver & Actuation**
  Use DC motors or stepper motors with encoders for precise speed control.
  Drive motors using H-Bridge motor drivers (e.g., L298N, BTS7960) based on control signals.
- **Remote Control & Navigation (Optional)**
  Integrate wireless control via Bluetooth, Wi-Fi, or smartphone app.
  Add obstacle detection using ultrasonic sensors or infrared proximity sensors.
- **Testing, Tuning & Real-Time Feedback**
  Continuously adjust PID gains and calibrate sensors during testing.
  Log data to visualize response curves, oscillation behavior, and convergence time.

Technologies Used:

- **Microcontrollers:** Arduino UNO/Nano, ESP32, STM32
- **Sensors:** MPU6050, MPU9250 (IMU), encoders, ultrasonic sensors
- **Motor Drivers:** L298N, TB6612FNG, BTS7960
- **Communication:** Bluetooth (HC-05), Wi-Fi (ESP32), Android app
- **Design Tools:** SolidWorks, Fusion 360, Eagle CAD (for custom PCBs)
- **Programming Languages:** C/C++ (Arduino IDE), MicroPython, MATLAB (for simulation)

MAJORS & AREAS OF INTEREST

This project is ideal for control systems, embedded robotics, and dynamic stabilization technologies. Key focus areas include:

- **Control Theory & PID Tuning** – Implementing PID, LQR, or advanced control strategies for balancing and stability.
- **Sensor Fusion & IMU Integration** – Using accelerometers, gyroscopes, and sensor fusion algorithms (e.g., Kalman filters) for tilt and motion estimation.
- **Embedded Systems & Microcontroller Programming** – Developing real-time control firmware on platforms like Arduino, STM32, or Raspberry Pi.
- **Dynamic Stabilization & Mobility Design** – Exploring inverted pendulum principles for mobile robotics and personal transportation systems.
- **Autonomous Navigation & Motion Planning** – Enabling obstacle avoidance, path following, and remote/manual control.
- **Simulation & Testing** – Using MATLAB/Simulink or Gazebo to simulate robot dynamics and test control strategies.
- **Applications in Robotics R&D** – Deploying self-balancing platforms for logistics, service robots, and mobile assistive devices.

MENTOR CONTACT INFORMATION

Dr. B Vijaya Krishna

Email: b.vijayakrishna@iare.ac.in

AI-Enhanced Combustion System Design for Emissions Reduction

Dr. V Mahider Reddy, Assistant Professor - Dept. of ME - Faculty Mentor

GOALS

This project aims to design and develop a gesture-controlled robotic arm to assist individuals with physical disabilities in performing essential tasks such as eating, picking up objects, and interacting with their surroundings. By translating natural hand or body gestures into robotic movements, the system provides an intuitive, non-verbal control mechanism that empowers users with limited mobility or speech impairments.

The robotic arm acts as an assistive extension, enhancing independence, improving quality of life, and reducing reliance on caregivers. The project integrates wearable gesture-sensing technologies with intelligent robotic control for real-time, user-friendly operation.

Key Goals:

- Develop a wearable gesture-sensing system (e.g., using IMU, flex sensors, or EMG) for capturing hand or finger movements.
- Translate detected gestures into corresponding robotic arm actions through real-time signal processing.
- Design a lightweight, responsive, and safe robotic arm suitable for daily assistive functions.
- Ensure ease of use, adaptability to different user needs, and low-cost implementation.
- Support inclusive, accessible technology aligned with assistive healthcare innovation.

METHODS & TECHNOLOGIES

Methods (Process & Approach):

- **Gesture Detection and Interpretation**
  Use wearable sensors (e.g., accelerometer, gyroscope, flex sensors, or EMG) to capture gesture data.
  Process signals using filtering techniques and classify gestures using thresholding or ML classifiers (e.g., SVM, KNN, Decision Trees).
- **Robotic Arm Design & Control**
  Design a 4–6 DOF robotic arm using CAD tools and fabricate using lightweight materials (e.g., PLA, aluminum).
  Use servo or DC motors for smooth, precise actuation controlled via microcontroller (e.g., Arduino or ESP32).
  Implement inverse kinematics for mapping gestures to arm movements.
- **Signal Processing & Mapping**
  Calibrate sensor data and convert gesture patterns into control signals for motor actuation.
  Implement safety constraints (e.g., motion limits, emergency stop) to prevent injury.
- **User Interface & Feedback**
  Provide visual or audio feedback to confirm recognized gestures.
  Design a simple GUI or mobile interface for system configuration and usage tracking.
- **Testing & Personalization**
  Test system responsiveness, accuracy, and usability with different users.
  Allow customization of gestures or sensitivity based on user needs and comfort.

Technologies Used:

- **Sensors:** Flex sensors, MPU6050 (IMU), MyoWare EMG sensor, accelerometers
- **Microcontrollers:** Arduino UNO/Nano, ESP32, STM32
- **Actuators:** Servo motors (MG996R, SG90), stepper motors
- **Control Algorithms:** Thresholding, SVM/KNN for gesture classification, inverse kinematics
- **Design & Prototyping Tools:** Fusion 360, SolidWorks, 3D printer, laser cutter
- **Programming Languages:** C/C++ (Arduino IDE), Python (for signal processing and ML)
- **Communication Interfaces:** Bluetooth, Wi-Fi, USB (for wireless or wired control)

MAJORS & AREAS OF INTEREST

This project is ideal for assistive robotics, wearable sensing, and human-machine interaction. Key focus areas include:

- **Gesture Recognition Technologies** – Using IMU, EMG, flex sensors, or computer vision for real-time gesture detection.
- **Signal Processing & Machine Learning** – Interpreting sensor signals and mapping them to precise robotic movements.
- **Robotic Arm Design & Control –** Building lightweight, safe, and adaptive robotic arms for assistive functions.
- **Embedded Systems & Actuation** – Implementing real-time control firmware and low-latency communication with robotic hardware.
- **Human-Centered Design** – Ensuring usability, comfort, and accessibility for individuals with disabilities.
- **Assistive Healthcare Robotics** – Creating systems to enhance independence in daily activities like eating, reaching, or object manipulation.
- **Affordable & Scalable Solutions** – Designing cost-effective prototypes suitable for personal and healthcare environments.

MENTOR CONTACT INFORMATION

Dr. V Mahider Reddy
Email: v.mahidharreddy @iare.ac.in