IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

HIGH IMPACT PRACTICES [HIPS]

CORNERSTONE PROJECTS: CONTROL SYSTEMS

INFORMATION PACKET

2025-2026

INSTITUTE OF AERONAUTICAL ENGINEERING
25 YEARS 2000 2025

I appreciate your interest in the Cornerstone Project (CoP), Department of EEE at the Institute of Aeronautical Engineering!

A cornerstone project (CoP) is typically introduced during the early or middle stages of an academic program at the Institute of Aeronautical Engineering. It focuses on helping students build foundational skills and understand how to apply basic concepts to real-world scenarios. These projects are usually smaller in scope, moderately complex, and designed to strengthen practical understanding of core subjects.

The Cornerstone Projects provide a platform for students to bridge the gap between classroom concepts and industry-relevant skills. Control Systems education benefits greatly from cornerstone projects that bridge theory with real-world application. One foundational project is the **self-balancing robot**, which mimics an inverted pendulum system. This project teaches students about unstable system dynamics and how feedback control, particularly PID and state-space methods, can stabilize such systems. Students gain experience with sensors like gyroscopes and accelerometers, and use microcontrollers (such as Arduino or Raspberry Pi) to implement real-time control algorithms..

Cornerstone Project (CoP) teams are:

- Collaborative Project – This is an excellent opportunity for students who are committed to working towards social developments and emerging needs.
- Project Activity – The project coordinator listed current working areas for offering cornerstone projects with a team size of at least two students. The coordinator allotted mentors based on the work area and facilitated exclusive project laboratories for selected cornerstone project (CoP)students. This cornerstone project (CoP)bridges the gap between academic learning and real-world social applications. It helps enhance the professional development
- Short-term - Each undergraduate student may participate in a project for an assigned period.

The primary goal of Cornerstone Projects in the Department of Electrical and Electronics Engineering (EEE) is to integrate foundational engineering knowledge with practical, real-world problem-solving to foster innovation, critical thinking, and hands-on technical skills.

- Apply theoretical concepts from circuits, electronics, Control Systems, embedded systems, and signal processing.
- Encourage team-based design thinking and interdisciplinary collaboration.
- Promote awareness of sustainable and socially relevant solutions aligned with global challenges (such as the UN Sustainable Development Goals).
- Prepare students for industry, entrepreneurship, and advanced research through experiential learning.
- Strengthen skills in circuit design, embedded systems, signal processing, IoT, power systems, and control engineering.
- Encourage students to design original solutions to engineering problems using creative approaches and emerging technologies.
- Cultivate the ability to work effectively in interdisciplinary teams, reflecting real-world engineering environments.
- Provide opportunities to work with tools and platforms like MATLAB, Arduino, Raspberry Pi, LabVIEW, and simulation software.
- Align project outcomes with social, environmental, or community needs—often by mapping them to the **UN Sustainable Development Goals (SDGs)**.
- Inspire students to explore deeper concepts, conduct experiments, and pursue independent or faculty-guided research.
- Develop adaptability and curiosity that prepare students for emerging technologies and continuous learning.

The research theme of this control systems-based projects also focus on the challenges presented by the Sustainable Development Goals (SDGs).

IARE Sustainability Development Goals (SDGs) highlighted with Blue Colour Font	
SDG #1	End poverty in all its forms everywhere
SDG #2	End hunger, achieve food security and improved nutrition and promote sustainable agriculture
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
SDG #14	Conserve and sustainably use the oceans, seas and marine resources for sustainable development
SDG #15	Protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss
SDG #16	Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels
SDG #17	Strengthen the means of implementation and revitalize the Global Partnership for Sustainable Development

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

1. Safe Control via AI Learning Control Barrier Functions (CBFs) with Neural Nets. **(SDG#3, SDG #4, SDG#9, SDG #11)**
2. AI-Based Fault-Tolerant Control Systems for Safety-Critical Applications. **(SDG#3, SDG #4, SDG#9, SDG #11)**
3. AI-Augmented Nonlinear and Chaotic Systems Control. **(SDG#4, SDG #9, SDG#11)**
4. Real-Time ML-Enhanced PID Tuning and Controller Auto-Calibration. **(SDG#4, SDG #9, SDG#11, SDG #12)**
5. Model Predictive Control (MPC) Enhanced with AI **(SDG#4, SDG #9, SDG#11, SDG #12,)**
6. Fault-Tolerant AI-Control Systems **(SDG#3, SDG #4, SDG#9, SDG #11)**
7. Energy-Efficient Control Using AI Optimization **(SDG#7, SDG #9, SDG#11, SDG #12, SDG#13)**
8. Human-in-the-Loop AI Control Systems, **(SDG#3, SDG #4, SDG#9, SDG #11)**

In order to participate in control systems Projects, you must formally apply and be accepted by the project coordinator. To proceed, please mail to the project coordinator, Dr. Damodhar Reddy, Professor and Head, Dept. of EEE, Email Id: dr.damodharreddy@iare.ac.in. This will bring up all available open positions tagged as control systems projects. When submitting a project document and an updated résumé, include a state- ment regarding why you are interested in working with the team to which you are applying. Please note that participation by the control systems project team requires registration for the accompanying research statement from any of the specified domains. More information will be provided to all selected control systems 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.

Safe Control via AI: Learning Control Barrier Functions (CBFs) with Neural Nets.

GOALS

The primary goal of this High Impact Practice (HIP) is to design AI-augmented controllers that ensure safety guarantees in dynamic systems by learning Control Barrier Functions (CBFs) using Neural Networks (NNs). CBFs enforce safety-critical constraints (like collision avoidance, speed limits, or safety envelopes), and combining them with AI enables adaptive, real-time, and robust safety-aware control.

Safety-Critical Control Design: Formulate controllers that guarantee system safety by maintaining system states within predefined safe sets using learned CBFs.

.**Learning CBFs from Data:** Use supervised learning, reinforcement learning, or imitation learning to model CBFs from simulation or real-world trajectory data, enabling safe behavior learning without full analytical models.

Integration with Control Lyapunov Functions (CLFs): Jointly optimize stability (via CLFs) and safety (via CBFs) in control policies using neural architectures.

End-to-End Trainable Safe Controllers: Design neural network-based control systems that learn to map system states directly to safe control actions, respecting control limits and safety barriers.

Real-Time Policy Adaptation: Enable online adaptation of learned CBFs in response to dynamic environments or system uncertainties, ensuring real-time constraint satisfaction.

Verification and Interpretability: Develop methods to verify and interpret learned CBFs, ensuring they remain valid and robust under disturbances or unmolded dynamics.

Safe Learning in the Loop: Integrate safety filters (CBFs) into reinforcement learning pipelines to enable exploration without violating safety constraints.

Hardware and Embedded Implementation: Deploy neural CBF-based controllers on real-time platforms like drones, autonomous ground vehicles, and robotic arms for safety-critical applications.

Application to Real-World Systems: Apply the approach to autonomous navigation, robot-human collaboration, medical robotics, and energy systems, where maintaining safety boundaries are non-negotiable.

Human-Aware and Environment-Aware Safety: Train CBFs to respect social and environmental constraints, such as maintaining safe distances from humans, obeying traffic rules, or protecting fragile payloads.

METHODS & TECHNOLOGIES

This HIP combines the theoretical foundations of control barrier functions (CBFs) with modern machine learning approaches to achieve safe, adaptive, and real-time control. The methods and tools below are essential for modeling, learning, verifying, and deploying AI-based safety controllers.

METHODS

- **Control Barrier Function (CBF) Formulation:** Define mathematical conditions to ensure system state remains in a safe set. Construct inequality constraints that the controller must satisfy at every time step.
- **Neural Network-Based CBF Learning:** Use supervised learning to approximate CBFs from expert demonstrations or simulation trajectories. Use deep reinforcement learning (DRL) to learn safe behaviors constrained by CBF conditions.
- **Safe Reinforcement Learning:** Apply techniques such as reward shaping, safe policy optimization, and constrained RL where CBFs act as safety filters.
- **Joint CBF and Control Lyapunov Function (CLF) Optimization:** Solve constrained optimization problems combining safety (CBFs) and stability (CLFs) using quadratic programming or neural policies.
- **End-to-End Differentiable Control:** Use differentiable programming to integrate CBF constraints directly into neural control policies and enable gradient-based learning.
- **Formal Verification & Robustness Testing:** Verify that learned CBFs generalize well across

disturbances, uncertainties, and model mismatches using robustness and reachability analysis tools.

- **Simulation-Driven Data Generation:** Use simulators to create diverse and rich datasets covering unsafe and safe trajectories for training.
- **Closed-Loop Simulation and Real-Time Testing:** Deploy learned controllers in simulation and test beds to validate safe closed-loop performance under real-time constraints.

Technologies & Tools

- **AI & Learning Libraries:**
  1. TensorFlow, PyTorch – For training neural networks to learn CBFs.
  2. Stable-Baselines3, RLlib – For reinforcement learning with safety constraints.
- **Control & Optimization Tools:**
  1. CVXPY, OSQP, Gurobi, MATLAB Optimization Toolbox – For solving constrained control problems involving CBF and CLF formulations.
  2. Control Systems Toolbox (MATLAB/Simulink) – For simulating nonlinear systems and applying control laws.
- **Simulation & Analysis Environments:**
  1. MATLAB/Simulink, MuJoCo, Gazebo, CARLA, ROS – For simulation of autonomous vehicles, robotic platforms, and safe learning.
  2. Julia (DifferentialEquations.jl, ControlSystems.jl) – For flexible scientific computing in safe control applications.
- **Verification & Safety Analysis:**
  1. Reachability Analysis Tools: CORA, SpaceEx, Verisig
  2. Safety-Critical Analysis Frameworks: Breach (MATLAB), S-TaLiRo
- **Hardware Platforms:**
  1. Raspberry Pi, Arduino, NVIDIA Jetson, STM32, dSPACE, Real-Time Linux – For deploying and testing real-time AI-CBF controllers.
  2. Sensors/Actuators – Integrated via I2C, UART, CAN for real-world interaction.

MAJORS & AREAS OF INTEREST

This HIP integrates AI, control theory, and real-time systems to develop safety-critical intelligent controllers. Students and researchers from the following majors will benefit the most from participating in this initiative:

Artificial Intelligence & Machine Learning: Safe reinforcement learning, neural network-based function approximation, Policy learning under safety constraints, deep learning for real-time decision-making.

Artificial Intelligence & Machine Learning: Deep learning, reinforcement learning, neuro-fuzzy systems for adaptive behavior modeling.

Data Science: Data-driven modelling and predictive analytics, Time-series analysis of system trajectories, Supervised learning from expert demonstrations, Outlier detection and safety violation prediction.

Robotics & Mechatronics: Human-robot interaction safety, Safe motion planning and collision avoidance, Sensor fusion and state estimation, Embedded AI controllers for mobile or industrial robots.

Cyber-Physical Systems & Embedded Systems: Real-time operating systems and interrupt handling, Safety-critical system designs for aerospace and automotive applications, Integration of learned controllers on physical platforms, Robustness to communication delays and hardware constraints.

Autonomous Systems & UAVs: Safety-aware navigation and path planning, redundant safety enforcement using learned CBFs, Mission assurance in uncertain and dynamic environments, Deployment of AI-CBFs in drones, autonomous vehicles, and medical robots

MENTOR CONTACT INFORMATION

Dr. Damodhar Reddy
Email: dr.damodharreddy@iare.ac.in

PARTNERS & SPONSORS

None

AI-Based Fault-Tolerant Control Systems for Safety-Critical Applications.

GOAL

Design AI-Enhanced Fault Detection, Isolation, and Recovery (FDIR) Systems:
Develop intelligent algorithms capable of identifying and isolating faults in sensors, actuators, or control logic in real time. Use machine learning, deep learning, or hybrid models (e.g., neuro-fuzzy systems) for pattern recognition and anomaly detection. Integrate statistical and data-driven approaches with model-based techniques to enhance detection accuracy.

Develop Robust and Resilient Control Architectures:
Implement AI-based fault-tolerant control strategies such as gain scheduling, adaptive control, reinforcement learning, or model predictive control. Ensure system stability, robustness, and minimum performance degradation under various fault conditions. Combine passive and active fault-tolerance techniques for layered resilience.

Integrate Dynamic Reconfiguration and Decision-Making:
Design AI systems that dynamically reconfigure control laws or switch between subsystems based on fault scenarios. Incorporate decision-making under uncertainty using techniques like Bayesian reasoning, Markov Decision Processes, or fuzzy logic. Utilize real-time optimization to select optimal recovery strategies.

Improve Predictive Maintenance and Fault Prognosis:
Employ AI to predict failures before they occur using historical data and condition monitoring.
Design early-warning systems that enable pre-emptive action to prevent system-level failures.
Reduce life-cycle maintenance costs by shifting from reactive to predictive approaches.

METHODS & TECHNOLOGIES

Model-Based Approaches: Observer-based methods (Luenberger observer, Kalman filter, Sliding Mode Observer), Parity space methods, analytical redundancy (comparing system model vs. measured outputs)

Data-Driven & AI-Based Approaches: Supervised Learning (e.g., SVMs, Random Forests, CNNs), Unsupervised Learning (e.g., Auto encoders, K-means, PCA), anomaly detection using neural networks (RNNs, LSTMs for time-series), signal processing techniques (Wavelet Transform, FFT)

Passive Fault-Tolerant Control: Robust control ($H\infty$, μ-synthesis), linear Parameter-Varying (LPV) systems, gain scheduling.

Active Fault-Tolerant Control: Adaptive control (MRAC, L1 adaptive), model Predictive Control (MPC), reinforcement Learning-based control (DDPG, PPO, etc.), switching and reconfiguration strategies.

System Integration and Real-Time Deployment: Sensor fusion (Kalman Filter, Particle Filter),hardware-in-the-loop (HIL) simulation, real-time AI inference on edge devices,explainability and Human-in-the-Loop AI,explainable AI (XAI) tools: SHAP, LIME, human trust calibration using interpretable models, visual dashboards for anomaly/fault display.

MAJORS & AREAS OF INTEREST

The development and deployment of AI-based fault-tolerant control systems in safety-critical applications is inherently interdisciplinary, drawing upon a diverse range of academic majors and research areas. Central to this field is Control Systems Engineering, which provides the theoretical foundation for designing robust and adaptive controllers capable of maintaining system stability in the presence of faults.

Closely related is Electrical and Electronics Engineering, which focuses on the integration of sensors, actuators, and embedded systems that facilitate real-time fault detection and mitigation. For applications involving autonomous machines or complex electromechanical systems, Mechatronics Engineering and Robotics Engineering offer crucial expertise in system integration, hardware interfacing, and intelligent actuation.

From a domain-specific perspective, Aerospace Engineering plays a critical role in the design of fault-tolerant flight control systems, where failures can have catastrophic consequences. Similarly, Automotive Engineering increasingly focuses on fault-tolerant architectures for autonomous vehicles and driver assistance systems, while Biomedical Engineering address's fault resilience in medical devices such as ventilators, infusion pumps, and robotic surgical platforms. In high-risk environments such as nuclear facilities, Nuclear Engineering contributes to the design of control systems that ensure continuous operation under strict safety constraints.

On the computational side, Artificial Intelligence, Machine Learning, and Computer Engineering are key enablers of intelligent fault detection, prediction, and decision-making. These disciplines contribute advanced algorithms such as deep learning, reinforcement learning, and hybrid AI, which enable systems to adapt to faults dynamically and with minimal human intervention. Cyber-Physical Systems (CPS) and Embedded Systems bring together the software and hardware aspects necessary for implementing real-time AI-based fault responses in physical environments.

Additionally, emerging areas such as Explainable AI, Formal Verification, and Human-in-the-Loop Systems are becoming increasingly important. These areas ensure that AI decisions are interpretable, verifiable, and aligned with human safety expectations. From a systems engineering standpoint, Data Science, Digital Twin Technology, and System Safety Engineering provide tools for modeling, simulating, and validating complex fault scenarios, enhancing predictive maintenance, and complying with international safety standards such as ISO 26262 (automotive), DO-178C (aerospace), and IEC 61508 (industrial).

MENTOR CONTACT INFORMATION

Dr. B Navothna

Email: b.navothna@iare.ac.in

PARTNERS & SPONSORS

None

AI-Augmented Nonlinear and Chaotic Systems Control.

GOALS

The primary goal of this HIP is to design and implement AI-augmented controllers capable of stabilizing and regulating nonlinear and chaotic systems. These systems are highly sensitive to initial conditions, nonlinear in dynamics, and often unpredictable using conventional control methods. By incorporating AI, the system gains adaptability, robustness, and improved predictive capabilities.

Stabilization of Chaotic Dynamics: Use AI algorithms to stabilize systems exhibiting chaotic behavior (e.g., Chua's circuit, Lorenz system, chemical reactors).

Real-Time Adaptive Control: Leverage AI to adapt controller parameters in real time based on system dynamics and disturbances.

Robust Control of Uncertain Nonlinear Systems: Handle modeling inaccuracies and parametric variations using intelligent approximates like neural networks or fuzzy systems.

Prediction and Forecasting: Integrate ML models to predict future behavior of chaotic systems for preemptive control. Hybrid AI-PID/Sliding Mode Controllers: Enhance conventional control laws with AI-tuned strategies to extend performance in nonlinear regimes.

Minimize Energy and Actuator Usage: Use data-driven optimization to achieve efficient control action with minimal energy or actuator wear.

Resilient System Design: Improve the resilience of critical applications (e.g., plasma control, cardiac systems, unmanned aerial vehicles) using AI-informed strategies

METHODS & TECHNOLOGIES

Methods

- **System Modelling:** Use differential equations or hybrid models to represent nonlinear/chaotic dynamics.
- **Chaos Analysis:** Perform Lyapunov analysis, bifurcation diagrams, Poincaré maps.
- **Data Generation & Simulation:** Generate synthetic data from simulations for AI training.
- **AI-Based Controller Design:** Reinforcement Learning, Deep Learning, and Fuzzy-Neural hybrid methods.
- **Feedback Learning:** Adapt control policies based on real-time performance feedback.
- **Closed-Loop Testing:** Evaluate AI controllers in real-time simulations and on embedded platforms.

Technologies & Tools

- **AI Libraries:** PyTorch, TensorFlow, Keras
- **Simulation Environments:** MATLAB/Simulink, SciPy, Julia, Control Systems Toolbox
- **Nonlinear System Tools:** XPPAUT, ChaosToolbox, Bifurcation software
- Hardware Platforms: Arduino, Raspberry Pi, NVIDIA Jetson, STM32, DSP boards
- **Real-Time OS:** FreeRTOS, RT-Linux (for real-time implementation)
- **Communication Protocols:** CAN, I2C, UART (for sensor/actuator integration)

MAJORS & AREAS OF INTEREST

The relevant academic programs and technical domains that provide the foundational knowledge and skill sets needed to understand, build, and improve real-time AI-Augmented Nonlinear and Chaotic Systems Control

Control Systems Engineering: Foundation in nonlinear and adaptive control theory, Lyapunov stability, state-space modeling.

Artificial Intelligence & Machine Learning: Deep learning, reinforcement learning, neuro-fuzzy systems for adaptive behavior modeling.

Electrical and Electronics Engineering: Circuit modeling, real-time hardware implementation, feedback electronics.
Applied Mathematics: Chaos theory, differential equations, dynamical systems, stability analysis.
Instrumentation & Embedded Systems: Real-time measurement, controller design on embedded hardware.
Data Science: Feature engineering, anomaly detection, time-series forecasting of chaotic variables.
Mechatronics & Robotics: Integration of AI controllers in robotic systems for real-time motion control in nonlinear environments.
The convergence of these majors reflects as a learner explores mathematical control foundations and AI integration, build and test AI controllers on real hardware for nonlinear systems, Prepares students for careers in autonomous systems, smart manufacturing, aerospace, and biomedical engineering.

MENTOR CONTACT INFORMATION

Dr. Damodhar Reddy
Email: dr.damodharreddy@iare.ac.in

PARTNERS & SPONSORS

None

Real-Time ML-Enhanced PID Tuning and Controller Auto-Calibration

GOAL

The primary goal of this HIP is to design a real-time intelligent control system that uses Machine Learning (ML) to enhance and auto-calibrate PID controllers during system operation. Conventional PID tuning often requires manual effort or offline optimization, which is inefficient and not adaptive to dynamic system changes. ML-based auto-tuning empowers control systems to adapt in real time to disturbances, model changes, or parameter drift, thus improving performance, stability, and energy efficiency.

Real-Time PID Tuning: Use ML algorithms to automatically adjust PID gains in real time for improved transient and steady-state performance.

Self-Calibration: Build intelligent systems capable of recognizing system behavior changes and self-adjusting control parameters without human intervention.

Model-Free Optimization: Apply reinforcement learning or adaptive algorithms where mathematical modeling is difficult or impractical.

Reduced Overshoot and Settling Time: Optimize PID behavior to minimize performance indices like IAE, ISE, or ITAE in real time.

Robustness to Disturbances and Noise: Enable dynamic retuning under external disturbances or noisy measurements.

Scalable across Systems: Generalize ML-based tuning strategies across HVAC, process control, robotics, and automotive domains

METHODS & TECHNOLOGIES

Methods:

Data Acquisition & pre-processing: Collect real-time signals such as process variable, control error, and system response from sensors; apply filtering, normalization, and feature extraction.

Performance Evaluation: Monitor real-time performance indices (Integral Absolute Error – IAE, Integral Square Error – ISE, Integral Time-weighted Absolute Error – ITAE) to guide learning and tuning.

Reinforcement Learning (RL): Apply Q-learning, Deep Q-Networks (DQN), or Actor-Critic models to dynamically tune PID parameters based on feedback and reward signals.

Supervised Learning for Gain Prediction: Train regression models or neural networks using labeled datasets of optimal PID gains under different operating conditions.

Adaptive Gain Scheduling: Use clustering, classification, or fuzzy logic to switch PID gains based on system behavior or region of operation.

Model-Free Policy Optimization: Deploy algorithms that learn control policies without requiring an explicit plant model, making them suitable for black-box or nonlinear systems.

Hardware-in-the-Loop (HIL) Testing: Use embedded or real-time simulation platforms to implement, evaluate, and validate ML-PID controllers in a closed-loop setup.

Technologies & Tools
ML & Optimization Frameworks:

1. TensorFlow, PyTorch, Scikit-learn (ML model development)
2. OpenAI Gym, Stable-Baselines3 (for Reinforcement Learning environments)
3. Optuna, Hyperopt (PID parameter optimization.

Simulation & Control Design Tools:

1. MATLAB/Simulink with PID Tuner Toolbox
2. Python (Control Systems Library, Gym Environments)

3. LabVIEW with real-time targets

Embedded & Real-Time Platforms:

1. Raspberry Pi, Arduino, STM32, BeagleBone Black.
2. PLCs (Siemens S7, Allen-Bradley) for industrial use.
3. NI CompactRIO or dSPACE for advanced HIL testing

MAJORS & AREAS OF INTEREST

The relevant academic programs and technical domains that provide the foundational knowledge and skill sets needed to understand, build, and improve real-time Real Time ML-Enhanced PID Tuning and Controller Auto-Calibration

Control Systems Engineering: Fundamental PID control theory, feedback loop design, frequency/time domain analysis, Ziegler–Nichols tuning, and stability analysis. Foundation in nonlinear and adaptive control theory, Lyapunov stability, state-space modeling.

Artificial Intelligence & Machine Learning: Deep learning, reinforcement learning, neuro-fuzzy systems for adaptive behavior modeling.

Electrical and Electronics Engineering: Circuit modeling, real-time hardware implementation, feedback electronics.

Applied Mathematics: Chaos theory, differential equations, dynamical systems, stability analysis.

Instrumentation & Embedded Systems: Real-time measurement, controller design on embedded hardware.
Data Science: Feature engineering, anomaly detection, time-series forecasting of chaotic variables.

Mechatronics & Robotics: Integration of AI controllers in robotic systems for real-time motion control in nonlinear environments.

The convergence of these majors reflects as a learner explores mathematical control foundations and AI integration, build and test AI controllers on real hardware for nonlinear systems, Prepares students for careers in autonomous systems, smart manufacturing, aerospace, and biomedical engineering.

MENTOR CONTACT INFORMATION

Dr. B Navothna
Email: b.navothna@iare.ac.in

PARTNERS & SPONSORS

None

Model Predictive Control (MPC) Enhanced with AI

GOAL

Design and implement an intelligent Model Predictive Control (MPC) framework enhanced by Artificial Intelligence techniques to optimize control performance in real-time under constraints, uncertainties, and nonlinearities.

Overcome MPC Limitations Using AI: Replace or augment traditional plant models in MPC using Neural Networks, LSTMs, or Gaussian Processes to handle complex nonlinear dynamics and time-varying systems.

Reduce Computational Load for Real-Time MPC: Use AI models to approximate optimization solutions or accelerate constraint handling using learning-based solvers.

Improve Robustness and Adaptability: Integrate Reinforcement Learning or Adaptive Learning techniques into MPC to adapt to unmodeled disturbances, parameter variations, and changing environments.

Enable Intelligent Multi-Objective Control: Implement AI-enhanced MPC to handle multiple objectives like energy efficiency, safety, and performance trade-offs through real-time prioritization.

Apply to Real-World Applications: Validate in complex domains such as: Autonomous Vehicles (e.g., path tracking, obstacle avoidance), Industrial Process Control (e.g., chemical reactors, furnaces), Smart Grid Systems (e.g., load dispatch, energy storage), Robotics and UAVs (e.g., trajectory planning)

METHODS & TECHNOLOGIES

Data-Driven System Identification: Use historical process data to model system dynamics via: Neural Networks (NNs), Long Short-Term Memory (LSTM) networks, Gaussian Process Regression (GPR).

AI-Augmented Prediction Models: Replace physics-based models in MPC with AI predictors: LSTM for time-series dynamics, Convolutional Neural Networks (CNNs) for spatial/vision inputs, Auto encoders for dimensionality reduction.

Learning-Based Optimization: Implement Reinforcement Learning (RL) agents to: Learn optimal control policies (e.g., with DDPG, PPO), Approximate MPC solutions when explicit optimization is too slow.

Adaptive and Online Learning: Use online training to adapt AI models to system drift or disturbances, incorporate real-time feedback into learning loops for continuous improvement.

Constraint Handling with AI: Use hybrid systems where AI assists in constraint tightening or active set estimation, Employ penalty-based methods or AI-informed feasible region estimators.

Hybrid MPC Frameworks: Combine traditional MPC with AI-enhanced modules: Predictive module → AI, Optimization module → classical solver, Adaptation & robustness → machine learning layer.

Simulation and Validation: Simulate and validate AI-MPC in platforms like:MATLAB/Simulink,Python (with cvxpy, casadi, do-mpc),ROS for robotic implementations.

Control Frameworks: do-mpc, cvxpy, mpctools, CasADi, FORCES Pro.

AI/ML Libraries: TensorFlow**,** PyTorch**,** Keras, scikit-learn**,** XGBoost for simpler models**,**GPy**,** GPyTorch for Gaussian Processes.

Reinforcement Learning Tools: OpenAI Gym, Stable-Baselines3, RLlib, Ray.

Model Deployment Tools: ONNX, TensorRT, Edge AI (NVIDIA Jetson, Intel Movidius).

Simulation Tools: MATLAB/Simulink, Simscape, LabVIEW,,Gazebo + ROS for robotics/MPC in physical environments.

Embedded and Real-Time Control: Arduino**,** Raspberry Pi**,** dSPACE**,** NI cRIO**,** Real-Time Linux.

MAJORS & AREAS OF INTEREST

MAJORS:

Control Systems Engineering: Focus: System modeling, optimization-based control, real-time control strategies.

Artificial Intelligence / Machine Learning: Focus: Deep learning, reinforcement learning, online learning for system identification and policy improvement.

Electrical and Electronics Engineering (EEE): Focus: Embedded systems, sensor integration, actuator control, real-time systems.

Mechanical Engineering: Focus: Mechatronics, robotics, automotive systems, HVAC control.

Aerospace Engineering: Focus: Flight control, UAV autopilot systems, trajectory planning with AI.

Computer Science and Engineering (CSE): Focus: AI algorithms, edge/cloud computing integration, high-performance simulation.

Robotics and Automation Engineering: Focus: Predictive path planning, adaptive control, AI-enhanced robot dynamics.

Mechatronics Engineering: Focus: Integration of AI in multi-domain control systems (e.g., electro-mechanical devices).

Data Science and Computational Engineering: Focus: Predictive modeling, system identification using AI, large-scale control systems.

Core Areas of Interest:

System Identification: Using machine learning (ML) techniques like Gaussian Processes, Neural Networks, or LSTMs to create accurate models for MPC.

Online Learning & Adaptation: Reinforcement learning (e.g., DDPG, PPO) to improve MPC decisions in dynamic environments.

Optimization Algorithms: AI-based solvers (e.g., genetic algorithms, metaheuristics) for faster and more adaptive MPC control action computation.

Robust & Adaptive Control: AI used to adjust the MPC cost function and constraints based on disturbances, uncertainty, and operational feedback.

Real-Time Implementation: Deploying AI-enhanced MPC on embedded platforms using edge AI accelerators (NVIDIA Jetson, Raspberry Pi + Coral, etc.).

Fault Tolerant & Predictive Maintenance: Using AI to forecast system degradation and adapt MPC accordingly.

Energy Optimization: Applying AI-MPC in smart grid, electric vehicles, and renewable energy systems for efficient control.

Human-in-the-Loop Control: Integrating user feedback with AI-based predictive models for collaborative or assistive systems (e.g., exoskeletons, autonomous vehicles).

MENTOR CONTACT INFORMATION

Dr. V Chandra Jagan Mohan
Email: vcjaganmohan@iare.ac.in

PARTNERS & SPONSORS

None

AI Fault-Tolerant AI-Control Systems

GOAL

AI Fault-Tolerant Control Systems aim to enhance the resilience and reliability of automated systems operating in uncertain or failure-prone environments. By integrating AI with traditional control strategies, these systems can detect, isolate, and compensate for faults in real time. This ensures continued safe operation without performance degradation. Such practices are critical in aerospace, robotics, healthcare, and other safety-critical applications.

Ensure System Reliability under Fault Conditions: Maintain control performance even in the presence of sensor, actuator, or component failures.

Enable Real-Time Fault Detection, Isolation, and Recovery (FDIR): Use AI to rapidly identify, localize, and respond to faults before they propagate or cause system degradation.

Minimize Downtime and Avoid Catastrophic Failures: Prevent system halts and ensure graceful degradation through predictive fault handling.

Automate Adaptive Control Reconfiguration: Implement self-healing controllers that automatically adjust parameters or switch control strategies in response to detected faults.

Enhance Safety and Resilience in Safety-Critical Systems: Apply AI to guarantee continued safe operation under degraded conditions, especially in aerospace, healthcare, and autonomous vehicles.

Enable Predictive Maintenance through Intelligent Monitoring: Leverage AI-driven diagnostics to predict faults before they occur, reducing maintenance costs and increasing uptime.

Improve Fault-Tolerance in Complex, Nonlinear, or High-Dimensional Systems: Use AI to manage the fault dynamics of systems that are too complex for traditional analytical approaches.

Support Decentralized or Distributed Fault-Tolerant Control: Facilitate resilient multi-agent or swarm control systems that can isolate local failures while preserving global goals.

Achieve Robust Performance with Incomplete or Corrupted Data: Use AI models to infer missing or corrupted measurements and maintain control functionality.

Accelerate Verification, Validation, and Deployment of Fault-Tolerant Strategies: Incorporate AI with digital twins or simulations to validate fault-tolerant control designs before real-world deployment.

METHODS & TECHNOLOGIES

Fault Detection and Diagnosis (FDD) Methods: Use supervised machine learning to classify normal vs faulty conditions, apply **unsupervised learning** (e.g., clustering, auto encoders) to detect unknown faults, Implement **residual generation techniques** (e.g., observer-based or parity space methods) for fault detection.

Fault Isolation Techniques: Analyze system response patterns to isolate the source of faults, Use Bayesian networks and decision trees for probabilistic fault localization.

Fault Estimation and Reconstruction: Employ Kalman filters extended Kalman filters (EKF), or neural networks to estimate fault magnitude and reconstruct missing/faulty data.

Control Reconfiguration and Adaptation: Use adaptive control to retune parameters when faults are detected, Apply model predictive control (MPC) that adapts to fault conditions in real time, Integrate switching control and gain scheduling for multi-mode fault adaptation**.**

Reinforcement Learning-Based Control: Train RL agents to optimize control policies under fault scenarios, Utilize safe reinforcement learning to preserve stability and safety during adaptation.

Robust and Redundant System Design: Design redundant control paths and backup controllers, Apply **robust control** techniques (e.g., H-infinity, sliding mode control) to tolerate model uncertainties and faults.

Digital Twin-Based Testing and Validation: Simulate faults and test controller responses using digital twins before deployment, Validate control logic with hardware-in-the-loop (HIL) simulation setups.

Artificial Intelligence & Machine Learning Platforms: TensorFlow, PyTorch, Scikit-learn – for developing ML/DL models for fault detection and prediction.

Keras Autoencoders, LSTM Networks – for time-series fault modeling and reconstruction.

Real-Time Embedded AI Hardware: NVIDIA Jetson, Google Coral, Intel Movidius – for running AI inference on edge devices in real time, ARM Cortex-M/A series – for embedded control with AI capabilities.

Control Design and Simulation Tools: MATLAB/Simulink **–** for modeling, simulation, and control design under fault conditions, Lab VIEW – for real-time control and hardware integration, Modelica**,** Open Modelica – for physical modeling and fault simulation.

Digital Twin & Simulation Platforms: Ansys Twin Builder**,** Siemens NX**,** MATLAB Simscape – for virtual replica testing and predictive fault management, Hardware-in-the-Loop (HIL) **–** using systems like dSPACE**,** NI PXI **for** real-time validation of AI-based fault-tolerant control.

Sensor Fusion and Signal Processing Tools: ROS (Robot Operating System) **–** for integrating multiple **sensors** and AI modules in robotics, Kalman Filters**,** Extended Kalman Filters (EKF) **–** for estimating true states from noisy/faulty measurements.

Cloud and Edge Computing Platforms: AWS Greengrass**,** Azure IoT Edge**,** Edge Impulse **–** for deploying fault detection and control AI at the edge**,** Real-time operating systems (RTOS) **–** like FreeRTOS**,** VxWorks for reliable low-latency performance in embedded control**.**

MAJORS & AREAS OF INTEREST

Control Systems Engineering: Focuses on system stability, control design, and feedback mechanisms under faulty conditions.

Artificial Intelligence and Machine Learning: Develops intelligent models for fault detection, diagnosis, and adaptive control.

Electrical and Electronics Engineering (EEE): Designs fault-resilient circuits, embedded systems, and sensor/actuator interfaces.

Robotics and Automation Engineering: Applies AI fault-tolerant control to robotic systems in uncertain and dynamic environments.

Aerospace Engineering: Implements high-reliability control systems in aircraft, drones, and spacecraft.
Mechatronics Engineering: Integrates mechanical, electrical, and software systems for intelligent, resilient control.

Computer Science and Engineering (CSE): Contributes to AI algorithms, embedded software, and edge/cloud integration for fault management.

Embedded Systems Engineering: Develops real-time fault-tolerant controllers on hardware platforms with strict timing constraints.

Mechanical Engineering (Control Specialization): Applies fault-tolerant control to mechanical systems like HVAC, automotive, and robotics.

Systems Engineering: Focuses on designing, managing, and validating complex fault-tolerant control architectures.

MENTOR CONTACT INFORMATION

Dr. SK Abdul Pasha
Email: shaik.abdulpasha@iare.ac.in

PARTNERS & SPONSORS

None

Energy-Efficient Control Using AI Optimization

GOAL

The primary goal of Human-in-the-Loop AI Control Systems is to design and develop control frameworks that effectively combine human intelligence and decision-making capabilities with AI algorithms to improve system performance, safety, and reliability. This approach seeks to leverage the complementary strengths of humans—such as intuition, ethical judgment, and contextual understanding—with the speed, precision, and adaptability of AI, creating collaborative systems that are more robust and trustworthy than fully autonomous solutions.

Enhance Control System Performance with Human Oversight: Combine human intuition and experience with AI algorithms to improve accuracy, safety, and efficiency in control systems.

Enable Adaptive and Interactive AI Control: Design AI systems that can adapt based on real-time human feedback, corrections, or preferences.

Build Trustworthy AI-Controlled Systems: Improve human trust in AI by ensuring transparency, predictability, and controllability through HITL integration.

Improve Decision-Making in Complex Environments: Use human input to handle ambiguity or ethical dilemmas where fully autonomous systems may struggle.

Reduce Risk in Safety-Critical Systems: Allow human intervention or supervision in high-stakes applications (e.g., healthcare, aviation, robotics) to prevent catastrophic failures.

Design Human-AI Collaboration Interfaces: Develop intuitive interfaces (e.g., GUIs, AR/VR, voice commands) for effective communication between humans and AI systems.

Develop Feedback Mechanisms: Create robust methods for incorporating human feedback into the learning and control loop (e.g., real-time corrections, demonstrations, preference learning).

METHODS AND TECHNOLOGS

Reinforcement Learning with Human Feedback (RLHF): AI agents learn optimal behaviour from human guidance (e.g., preference comparisons, corrections).

Imitation Learning / Learning from Demonstration (LfD): AI learns control policies by mimicking human demonstrations.

Inverse Reinforcement Learning (IRL): AI infers the reward function underlying a human's actions. Robotics, autonomous driving, personalized AI behaviour.

Active Learning: AI queries a human for labels or decisions only when needed (uncertainty-driven). Reducing human effort while improving model accuracy.

Interactive Machine Learning: Continuous model improvement based on human interaction and iterative refinement.

Shared Control Systems: Humans and AI agents simultaneously control the system (e.g., haptic steering in autonomous vehicles). Combine strengths of both human intuition and AI precision.

Teleoperation with AI Assistance: Humans remotely control robots or systems with real-time AI support. AI auto-stabilizing a robotic arm while a human guides it.

MAJORS & AREAS OF INTEREST

Students and researchers interested in Human-in-the-Loop AI control systems typically come from interdisciplinary fields. Computer Science or Artificial Intelligence is a core major, as it provides the foundation in machine learning, reinforcement learning, and algorithm development necessary for HITL systems. Robotics and Mechatronics Engineering are highly relevant for implementing control systems that blend mechanical action with human oversight. Electrical and Mechanical Engineering offer strong backgrounds in automation, sensor integration, and real-time system control.

Another critical field is Human-Computer Interaction (HCI), which focuses on designing interfaces that enable seamless human feedback and supervision. Meanwhile, majors in Cognitive Science or Psychology are important for understanding human decision-making, behavior modelling, and trust—essential components when developing AI systems that interact closely with humans. Biomedical Engineering also plays a role, especially in HITL systems for assistive technologies like prosthetics or brain-computer interfaces.

Key areas of interest in this domain include Human-Robot Interaction (HRI), where the goal is to create collaborative systems that safely and efficiently integrate human input with robotic autonomy. Reinforcement Learning from Human Feedback (RLHF) is a growing area where AI agents improve through human preferences, corrections, or demonstrations rather than purely algorithmic rewards. Interactive Machine Learning involves systems that evolve and adapt as users provide real-time feedback, improving over time in response to human input.

Other important fields include Teleportation and Assistive Technologies, where humans remotely guide or supervise AI-controlled systems, such as surgical robots or drones. Finally, Explainable AI (XAI) and Human Trust in AI are vital areas of interest, focusing on building systems that not only perform well but are transparent and interpretable, allowing human operators to understand, trust, and effectively intervene in the control process when needed.

MENTOR CONTACT INFORMATION

Dr. G Seshadri

Email: dr.gseshadri@iare.ac.in

PARTNERS & SPONSORS

None

Human-in-the-Loop AI Control Systems,

GOAL

The primary goal of Human-in-the-Loop AI Control Systems is to design and develop control frameworks that effectively combine human intelligence and decision-making capabilities with AI algorithms to improve system performance, safety, and reliability. This approach seeks to leverage the complementary strengths of humans such as intuition, ethical judgment, and contextual understanding with the speed, precision, and adaptability of AI, creating collaborative systems that are more robust and trustworthy than fully autonomous solutions.

Enhance Control System Performance with Human Oversight: Combine human intuition and experience with AI algorithms to improve accuracy, safety, and efficiency in control systems.

Enable Adaptive and Interactive AI Control: Design AI systems that can adapt based on real-time human feedback, corrections, or preferences.

Build Trustworthy AI-Controlled Systems: Improve human trust in AI by ensuring transparency, predictability, and controllability through HITL integration.

Improve Decision-Making in Complex Environments: Use human input to handle ambiguity or ethical dilemmas where fully autonomous systems may struggle.

Reduce Risk in Safety-Critical Systems: Allow human intervention or supervision in high-stakes applications (e.g., healthcare, aviation, robotics) to prevent catastrophic failures.

Design Human-AI Collaboration Interfaces: Develop intuitive interfaces (e.g., GUIs, AR/VR, voice commands) for effective communication between humans and AI systems.

Develop Feedback Mechanisms: Create robust methods for incorporating human feedback into the learning and control loop (e.g., real-time corrections, demonstrations, preference learning).

METHODS AND TECHNOLOGS

Reinforcement Learning with Human Feedback (RLHF): AI agents learn optimal behaviour from human guidance (e.g., preference comparisons, corrections).

Imitation Learning / Learning from Demonstration (LfD): AI learns control policies by mimicking human demonstrations.

Inverse Reinforcement Learning (IRL): AI infers the reward function underlying a human's actions. Robotics, autonomous driving, personalized AI behavior.

Active Learning: AI queries a human for labels or decisions only when needed (uncertainty-driven). Reducing human effort while improving model accuracy.

Interactive Machine Learning: Continuous model improvement based on human interaction and iterative refinement.

Shared Control Systems: Humans and AI agents simultaneously control the system (e.g., haptic steering in autonomous vehicles). Combine strengths of both human intuition and AI precision.

Teleoperation with AI Assistance: Humans remotely control robots or systems with real-time AI support. AI auto-stabilizing a robotic arm while a human guides it.

MAJORS & AREAS OF INTEREST

Students and researchers interested in Human-in-the-Loop AI control systems typically come from interdisciplinary fields. Computer Science or Artificial Intelligence is a core major, as it provides the foundation in machine learning, reinforcement learning, and algorithm development necessary for HITL systems. Robotics and Mechatronics Engineering are highly relevant for implementing control systems that blend mechanical action with human oversight. Electrical and Mechanical Engineering offer strong backgrounds in automation, sensor integration, and real-time system control.

Another critical field is Human-Computer Interaction (HCI), which focuses on designing interfaces that enable seamless human feedback and supervision. Meanwhile, majors in Cognitive Science or Psychology are important for understanding human decision-making, behaviors modelling, and trust—essential components when developing AI systems that interact closely with humans. Biomedical Engineering also plays a role, especially in HITL systems for assistive technologies like prosthetics or brain-computer interfaces.

Key areas of interest in this domain include Human-Robot Interaction (HRI), where the goal is to create collaborative systems that safely and efficiently integrate human input with robotic autonomy. Reinforcement Learning from Human Feedback (RLHF) is a growing area where AI agents improve through human preferences, corrections, or demonstrations rather than purely algorithmic rewards. Interactive Machine Learning involves systems that evolve and adapt as users provide real-time feedback, improving over time in response to human input.

Other important fields include Teleoperation and Assistive Technologies, where humans remotely guide or supervise AI-controlled systems, such as surgical robots or drones. Finally, Explainable AI (XAI) and Human Trust in AI are vital areas of interest, focusing on building systems that not only perform well but are transparent and interpretable, allowing human operators to understand, trust, and effectively intervene in the control process when needed.

MENTOR CONTACT INFORMATION

Dr. V Chandra Jagan Mohan
Email: vcjaganmohan@iare.ac.in

PARTNERS & SPONSORS

None