NATIONAL INSTITUTE OF ELECTRONICS AND INFORMATION TECHNOLOGY

DEEMED TO BE UNIVERSITY UNDER DISTINCT CATEGORY

(AN AUTONOMOUS INSTITUTION UNDER MINISTRY OF ELECTRONICS & IT, GOVT. OF INDIA)

Curriculum and Syllabi for

M.Tech - Automotive Electronics

(2025-26 Scheme)

रा.इ.सू.प्रौ.सं

NIELIT

Deemed to be University

Main Campus at Ropar and Constituent Units at Aizawl, Agartala, Aurangabad, Calicut, Gorakhpur, Imphal, Itanagar, Kekri, Kohima, Patna and Srinagar

Vision

To empower and transform fresh graduates and upskill working professionals into competent engineers through advanced education and research in automotive embedded systems, while fostering innovation, sustainability, and safety in next-generation intelligent transportation technologies.

Mission

To train and transform fresh graduates and working professionals into industry-ready engineers by imparting advanced knowledge and practical skills in automotive embedded systems, fostering innovation, sustainability, and safety in intelligent transportation through research-driven, industry-aligned learning.

Program Education Objectives

PEO1: Graduates will possess strong technical expertise in automotive embedded systems and electronics product design, with a solid foundation in both hardware and software aspects.
PEO2: Graduates will be capable of applying advanced technologies such as AI, machine learning, data science etc. to solve complex problems in intelligent transportation and automotive electronics.
PEO3: Graduates will demonstrate proficiency in adhering to industry standards and practices, including vehicle communication protocols, functional safety and automotive software engineering.
PEO4: Graduates will engage in research and innovation, contributing to sustainable and safe advancements in automotive electronics and embedded systems.
PEO5: Graduates will exhibit professional ethics, effective communication, and lifelong learning abilities, preparing them for leadership roles in industry, academia, and entrepreneurship.

Program Outcomes

Upon successful completion of the program, the learners will be able to:
PO1: Apply advanced knowledge of mathematics, science, and engineering fundamentals to analyze and design automotive embedded systems and electronics products.
PO2: Design and develop hardware and software components of automotive embedded systems, ensuring functionality, reliability, and efficiency.
PO3: Use modern tools and technologies, including AI, machine learning and real-time operating systems, for solving complex engineering problems in intelligent transportation systems.
PO4: Demonstrate understanding and implementation of industry standards and protocols such as ISO 26262, vehicle communication protocols, and cyber security measures for automotive applications.
PO5: Conduct research and development activities to innovate and improve automotive electronics, embedded systems, and intelligent vehicle technologies.
PO6: Exhibit professional and ethical responsibility, effective communication, teamwork, and leadership skills in multidisciplinary environments.

Program Specific Outcomes

PSO1: Apply specialized knowledge in embedded systems, automotive communication protocols, and real-time operating systems to design and implement efficient and reliable automotive electronics solutions.
PSO2: Develop and integrate advanced technologies such as AI, machine learning, and autonomous systems within automotive embedded platforms to enable intelligent, safe, and sustainable mobility solutions.
PSO3: Implement functional safety, security standards, and quality management practices in the design and development of automotive software and hardware to meet industry compliance requirements.

CATEGORY WISE CREDIT DISTRIBUTION

Category	Category Code	Credits
Open Elective Courses	OE	8
Audit Courses (without grade or credit)	AU	0
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE	36
Professional Elective Courses	PE	16
Program Core Courses	PC	20
Total Credits (Common track)		**80**

EVALUATION AND ASSESSMENT

1. Performance of a student in a semester shall be evaluated through continuous Class assessment, Tutorial/Lab assessment, Mid-Semester Examination (MSE) and End-Semester Examination (ESE). Both the MSE and ESE shall be the University examination and will be conducted as notified by the Controller of Examinations (CoE) of the University.

2. The continuous assessment shall be based on assignments, tutorials, paper presentation / quizzes/ viva-voce / flipped classes, lab work / projects / fieldwork and attendance, etc.

3. The MSE/ESE shall be comprising of written papers.

4. The overall assessment of the students will be done as per the following scheme:

S. No.	Assessment Type	Marks Weightage
1.	Mid Semester Examination	20
2.	End Semester Examination	40
3.	Continuous Assessment - Practical /Lab/ Tutorial	25
4.	Continuous Assessment - Theory	15
Total		**100**

For more details, please refer the applicable ordinance for this programme and Assessment SOPs.

CURRICULUM

Semester 1

Semester Code	Course Code	Course Title	L	T	P	Cr
PC-MTEAUT-101	DASH250002	Advanced Engineering Mathematics	3	1	0	4
PC-MTEAUT-102	DOEE250185	System Design Using Embedded Processors	3	0	2	4
PC-MTEAUT-103	DOEE250046	Automotive Control Systems	3	0	2	4
PE-MTEAUT-104	Elective	Program Elective				4
EE-MTEAUT-105	DASH250095	Research Methodology and IPR	3	1	0	4
AU-MTEAUT-106	Elective	Audit Elective				0

Semester 2

Semester Code	Course Code	Course Title	L	T	P	Cr
PC-MTEAUT-201	DOAI250002	Advanced Machine Learning	3	0	2	4
PC-MTEAUT-202	DOEE250058	In Vehicle Communication Protocols	3	0	2	4
PE-MTEAUT-203	Elective	Program Elective				4
PE-MTEAUT-204	Elective	Program Elective				4
OE-MTEAUT-205	Elective	Open Elective				4
AU-MTEAUT-206	Elective	Audit Elective				0

Semester 3

Semester Code	Course Code	Course Title	L	T	P	Cr
PE-MTEAUT-301	Elective	Program Elective				4
OE-MTEAUT-302	Elective	Open Elective				4
EE-MTEAUT-303	EE	Dissertation -Phase I				14

Semester 4

Semester Code	Course Code	Course Title	L	T	P	Cr
EE-MTEAUT-401	EE	Dissertation -Phase II				18

Elective Courses

Elective Courses for PE Category						
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**	**For Sem./Code**
DCSE250020	Cloud Computing	3	0	2	4	104
DCSE250056	Data Structures and Algorithms	3	0	2	4	104
DOEE250090	Embedded Programming	3	0	2	4	104
DOEE250081	Autonomous Vehicle Systems and AI Integration	3	0	2	4	203
DOEE250088	Embedded OS and RTOS	3	0	2	4	203
DOEE250194	Vehicle Security and Comfort Systems	3	1	0	4	203
DOEE250089	Automotive Power Electronics and Motor Drives	3	1	0	4	204
DOEE250091	Software and ISO Standards in Automotive Systems	3	1	0	4	204
DOEE250093	Automotive Engine Management and Sensors Systems	3	1	0	4	204

Elective Courses for OE Category
Students may opt courses under MOOC, NPTEL, SWAYAM, NSQF/NCVET aligned courses or other relevant technical subjects not included in their core curriculum. The exercise of option shall be subject to the Course Schedule of the respective Constituent Unit, the credit requirements and availability of seats. Guidelines for choosing MOOC/Online courses are given separately and shall have to be adhered to.

Elective Courses for AU (Audit) Category					
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**
DASH240027	Disaster Management	2	0	0	0
DASH240047	English for Research Paper Writing	2	0	0	0
DASH240052	Environmental Science	3	0	0	0
DASH240085	Pedagogy Studies	2	0	0	0
DASH240086	Personality Development through Life Enlightenment Skills	2	0	0	0
DASH240097	Sanskrit for Technical Knowledge	2	0	0	0
DASH240103	Stress Management by Yoga	2	0	0	0
DASH240104	Technical Presentation and Report Writing	0	0	2	0
DASH240106	Value Education	2	0	0	0
DASH240124	Constitution of India - AU	2	0	0	0
DASH250023	Constitution of India	2	1	0	0
DASH250027	Disaster Management	3	0	0	0
DASH250034	Energy and Environmental Engineering	3	0	0	0
DASH250047	English for Research Paper Writing	2	0	0	0
DASH250054	Essence of Indian Knowledge and Tradition	2	0	0	0
DASH250085	Pedagogy Studies	2	0	0	0
DASH250086	Personality Development	2	0	0	0
DASH250097	Sanskrit for Technical Knowledge	3	0	0	0
DASH250103	Stress Management by Yoga	3	0	0	0

DASH250106	Value Education	2	0	0	0
DASH250117	Innovative Thinking and Entrepreneurship Skills	1	0	0	0
DASH250118	Intellectual Property Rights	2	0	0	0
DCSA240080	Training on Computer Networking	0	0	2	0
DCSA240304	Lab Based on Statistical Package	0	0	2	0
Any other Scheduled Course as per the Course Schedule of the respective Constituent Unit can also be chosen, subject to availability of seats. However, there shall be no assessment and grading for the course chosen as Audit Course.					

The choice of all type of Electives available shall be as per the Course Schedule of the respective Constituent Unit.

Guidelines for Massive Open Online Courses (MOOC) / approved Online Courses

1. Relevant Swayam, MOOC, NPTEL, NIELIT NSQF and any other online courses approved by UGC/AICTE shall also be allowed as Open Electives(OE).

2. One credit of MOOC course should be minimum of 30 hours.

3. A student can choose any approved online course as Open Elective, subject to minimum credit requirements.

4. The students willing to opt OE under MOOC in their next semester, will submit their request to the MOOC Coordinator at their respective campus.

5. Once approved, the campus shall display the list of accepted courses.

6. The student has to submit certificate issued by the institution offering the MOOCs course, along with the number of credits and grades, to get credits transferred into his/her marks certificate issued by NDU.

7. The transfer of credits shall be as adopted by NDU.

Detailed Syllabus

Course Code	Course Name	L	T	P	Cr
DASH250002	Advanced Engineering Mathematics	3	1	0	4

Course Outcomes:

CO1 : Model real-world uncertainty using probability theory and manipulate random variables in applied contexts.
CO2 : Analyze convergence behavior of sequences of random variables and apply statistical limit theorems in AI/ML.
CO3 : Solve abstract algebraic and vector space problems relevant to machine learning algorithmic design.
CO4 : Perform linear transformations, change of basis, and analyze solvability of linear systems in data-centric applications.
CO5 : Apply numerical methods such as interpolation, numerical differentiation, and integration in data approximation and machine learning tasks.

Module 1:

Probability Theory and Random Variables (9 Hours)

Axiomatic definition of probability, Bayes' theorem with applications, Random variables: discrete and continuous, Cumulative distribution function (CDF), Probability Mass Function (PMF), Probability Density Function (PDF), Conditional and joint distributions, independence of random variables.

Module 2:

Expectation, Moment Generating Functions, and Limit Theorems (9 Hours)

Expectation and fundamental properties, Moment generating functions (MGF) and characteristic functions (CF), Conditional expectation and covariance matrix, Uncorrelated random variables and joint Gaussian PDFs, Markov inequality, Chebyshev inequalities, Central Limit Theorem (CLT), weak and strong laws of large numbers, Modes of convergence of random variables. Random Process, Markov Process and Markov Chain.

Module 3:

Algebraic Structures and Vector Spaces (9 Hours)

Groups, rings, and fields (basic definitions and examples), Vector spaces and subspaces over a field, Linear independence, span, basis, and dimension, Direct sum of subspaces. System of Equations, Solving system of linear equations using Gaussian elimination.

Module 4:

Linear Transformations (9 Hours)

Linear transformations and their matrix representation, Four fundamental subspaces: null space, column space, row space, and left null space, Rank, Rank-Nullity Theorem, Solving systems of linear equations: existence and uniqueness, Transpose of a linear transformation. Orthogonality & Orthonormality, Orthogonalization, Eigen Value, Eigen Vector, Diagonalization of vectors.

Module 5:

Interpolation, Numerical Differentiation & Integration (9 Hours)

Lagrange’s interpolation, Newton’s divided difference interpolation, Newton’s forward and backward difference interpolation, Approximation of derivatives using interpolation polynomials, Numerical integration using: Trapezoidal rule, Simpson’s 1/3 rule.

Labs / Practicals:

N/A

References:

1. Gupta, S. C., and Kapoor, V. K., Fundamentals of Mathematical Statistics, Sultan Chand & Sons, New Delhi.
2. Goon, A. M., Gupta, M. K., and Dasgupta, B., An Outline of Statistical Theory, Vol. 1 & 2, World Press, Kolkata.
3. Hogg, R. V., McKean, J., and Craig, A. T., Introduction to Mathematical Statistics, Pearson Education.
4. Papoulis, A., and Pillai, S. U., Probability, Random Variables and Stochastic Processes, McGraw Hill Education.
5. Rao, C. R., Linear Statistical Inference and Its Applications, Wiley Eastern Limited.
6. Ramana, B. V., Higher Engineering Mathematics, Tata McGraw Hill Education.
7. Hoffman, K., and Kunze, R., Linear Algebra, Prentice Hall of India.
8. Jain, M. K., Iyengar, S. R. K., and Jain, R. K., Numerical Methods for Scientific and Engineering Computation, New Age International Publishers, New Delhi.
9. Kreyszig, E., Advanced Engineering Mathematics, Wiley India Edition.
10. Lipschutz, S., and Lipson, M., Linear Algebra, Schaum’s Outline Series, McGraw Hill Education.

Course Code	Course Name	L	T	P	Cr
DOEE250185	System Design Using Embedded Processors	3	0	2	4

Course Outcomes:

CO 1 Comprehend Embedded Processor and its software
CO 2 Design an Embedded system with ARM microcontrollers
CO 3 Design an Embedded system using processors, memory I/O devices and communication network within realistic constraints
CO 4 Comprehend ARM Cortex M4 microcontrollers
CO 5 Comprehend the peripheral programming of ARM cortex M4 Microcontrollers
CO 6 Comprehend advanced Embedded Controllers, Features and case studies
CO 7 Perform ARM Cortex M4 Microcontroller configuration for various applications using peripheral devices
CO 8 Demonstrate an embedded system on ARM Cortex M4 Microcontroller development board

Module 1:

Introduction to Embedded Systems
Overview of embedded system architecture, Development and debugging Tools for Embedded Systems, Overview ARM Architecture - Architecture Versions, Instruction Set Development,Thumb-2 and Instruction Set Architecture, AMBA, AXI bus overview. Overview of the ARM Cortex-Mx Processor Architectures.

Module 2:

ARM Cortex M4 Microcontroller System
ARM Cortex M4 Core, Interconnect Matrix in ARM Cortex M4 Microcontroller, System configuration Controller, NVIC, External Interrupt Controllers, DMA, Reset and Clock Control, Clock Recovery System, Power Control.

Module 3:

ARM Cortex M4 Microcontroller Peripheral Overview
Introduction to ARM Cortex-M4 Programming, overview of CMSIS, GPIO, ADC, DAC, Communication & Peripherals - USART, UART, I2C, SPI, USB, CAN
Watchdogs and Timers, PWM.

Module 4:

Memory, Safety and Security in ARM Cortex Microcontroller
Flash, Quad SPI Interface, Flexible Memory controller, CRC, Random Number Generator, memory protections, Advanced Encryption Standard HW Accelerator (AES), Safety support.

Module 5:

Advanced Embedded Controllers, Features and case studies
Programming for Power-Efficient Computing - High Level and low level Techniques, Cortex M7, M23 and M33 Controllers and Features, Overview of mbed platform, Embedded Systems case studies - Consumer, Medical, Automotive.

Labs / Practicals:

Embedded C Programming on ARM Cortex M4 Microcontroller with CMSIS/HAL libraries, Embedded C Programming - Peripheral Programming - GPIO, ADC, DAC, USART, Timers and PWM, Design of a real-time data acquisition & control system using the ARM Cortex M4 Microcontroller

References:

1. Yiu J. The Definitive Guide to ARM Cortex M3 and Cortex M4 Processors, 3rd Edition, Elsevier
2. Andrew N Sloss, Dominic Symes, Chris Wright, “ARM System Developer's Guide - Designing and Optimizing System Software”, 2006, Elsevier

3. Steve Furber, "ARM System-on-Chip Architecture", 2nd Edition, Pearson Education
4. Raj Kamal, "Microcontroller - Architecture Programming Interfacing and System Design" 1st Edition, Pearson Education
5. P.S Manoharan, P.S. Kannan, "Microcontroller based System Design", 1st Edition, Scitech Publications
6. Cortex-M4 Technical Reference Manual (TRM)
7. STMicroelectronics Nucleo-G4xx development board user manual

Course Code	Course Name	L	T	P	Cr
DOEE250046	Automotive Control Systems	3	0	2	4

Course Outcomes:

CO 1 Understand fundamentals of control systems and feedback theory.
CO 2 Model and analyze automotive dynamic systems (mechanical and electrical).
CO 3 Design basic control strategies (PID, state feedback) for automotive systems.
CO 4 Apply control techniques to real-world automotive applications such as cruise control, ABS, and ESC.
CO 5 Analyze performance and stability of automotive control systems using simulation tools.
CO 6 Implement control logic using software tools and embedded controllers.

Module 1:

Introduction to Control Systems
- Basic concepts of open and closed-loop systems
- Transfer function, block diagram simplification
- Feedback and stability concepts
- Time and frequency domain analysis

Module 2:

Modeling of Automotive Systems
- Modeling of mechanical systems (sprung/unsprung mass, suspensions)
- Modeling of electrical systems (motors, actuators)
- State-space representation
- Linearization and system identification

Module 3:

Classical Control Techniques
- PID control design and tuning
- Root locus and frequency response
- Lead-lag compensators
- Design examples: Cruise control, throttle control

Module 4:

Automotive Control Applications
- Anti-lock Braking System (ABS) control logic
- Electronic Stability Control (ESC)
- Adaptive Cruise Control (ACC)
- Drive-by-wire systems

Module 5:

Control Implementation and Tools
- Embedded implementation of control logic
- MATLAB/Simulink for control design
- Rapid control prototyping
- Hardware-in-the-loop (HIL) simulation

Labs / Practicals:

Practical Sessions on MATLAB/Simulink/LabVIEW and other Tools.

References:

1. Rajamani, Rajesh. Vehicle Dynamics and Control. Springer.
2. Ioan Doré Landau. Digital Control Systems. Springer.
3. I. J. Nagrath, M. Gopal. Control Systems Engineering. New Age Publishers.
4. Karl Johan Åström, Richard M. Murray. Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press.
5. William Ribbens. Understanding Automotive Electronics. Elsevier.

Course Code	Course Name	L	T	P	Cr
DASH250095	Research Methodology and IPR	3	1	0	4

Course Outcomes:

CO1: Understand the principles and process of research methodology.
CO2: Apply appropriate research design and data collection methods in computing projects.
CO3: Analyze and interpret qualitative and quantitative data using statistical tools.
CO4: Develop ethically sound, well-documented research or project reports conforming to academic and industrial standards.
CO5: Understand various forms of intellectual property and apply processes for protection and patent filing.

Module 1:

Introduction to Research and Research Ethics
Definition and objectives of research, types of research – fundamental, applied, exploratory, empirical, significance of research in computer applications, research ethics and integrity, plagiarism, citation styles (APA, IEEE), tools for plagiarism check.

Module 2:

Research Design and Data Collection
Research problem identification and formulation, hypothesis generation, literature review techniques, types of research design: experimental, descriptive, analytical, sampling techniques, primary and secondary data sources, survey design, case study design.

Module 3:

Data Analysis and Interpretation
Quantitative and qualitative data analysis, measures of central tendency and dispersion, t-test, ANOVA, correlation and regression, chi-square test, software tools: Excel, R, SPSS (demo-level), data visualization and result interpretation.

Module 4:

Technical Writing and Project Report Preparation
Structure of research papers and reports, abstract, introduction, literature review, methodology, results, and conclusions, referencing tools (Mendeley/Zotero), IEEE/ACM/SCI journal guidelines, thesis and dissertation formatting, proposal writing basics.

Module 5:

Intellectual Property Rights and Patent Process
Introduction to IPR: patents, copyrights, trademarks, industrial design, Indian and international patent systems (WIPO, TRIPS), criteria for patentability, patent filing process in India (IPINDIA portal), open-source licenses, patent drafting basics, IPR in software and AI innovations.

Labs / Practicals:

N/A

References:

1. C.R. Kothari & Gaurav Garg – Research Methodology: Methods and Techniques, New Age International.
2. Ranjit Kumar – Research Methodology – A Step-by-Step Guide, Sage Publications India.
3. P. Narayan – Intellectual Property Law, Eastern Book Company.
4. Neeraj Pandey & Khushdeep Dharni – Intellectual Property Rights, PHI Learning.
5. Yogesh Kumar Singh – Fundamentals of Research Methodology and Statistics, New Age International.
6. Bharat Bhushan – Research Methodology in Computer Science, Himalaya Publishing.
7. WIPO and IPIndia Manuals – Guidelines for Filing Patents in India.

Course Code	Course Name	L	T	P	Cr
DOAI250002	Advanced Machine Learning	3	0	2	4

Course Outcomes:

CO1: Understand the foundational concepts of machine learning, explore various learning paradigms and tasks, and apply essential data preprocessing and evaluation techniques.
CO2: Apply linear and logistic regression, regularization methods, and classification algorithms including SVM, Decision Trees, and Naïve Bayes for predictive modeling.
CO3: Analyze data using clustering algorithms and perform effective feature selection and dimensionality reduction to enhance machine learning model performance.
CO4: Gain knowledge of probabilistic graphical models and implement ensemble techniques to improve model robustness and accuracy.
CO5: Develop and apply deep learning models including CNNs, RNNs, LSTMs, and Transformers for advanced applications in vision, sequence processing, and natural language understanding.

Module 1:

Overview of machine learning: supervised, semi-supervised, unsupervised learning, reinforcement learning.
Types of Machine Learning Tasks: Classification, Regression, and Clustering.
Cross Validation.
Classifier and Regressor Performance Evaluation Measures, ROC Curves.
Cost functions: Definition and Types.
Bias-Variance trade off.
Data Preprocessing: Standardization, Normalization, Label Encoding, One-hot Encoding.

Module 2:

Linear Regression, Least Square Gradients.
Regularization: Ridge and Lasso Regression.
Classification Methods: K Nearest Neighbor, Logistic Regression.
Sigmoid function and differentiation.
Support Vector Machine (SVM): Optimal Separating Hyperplane, Kernel Trick, Kernel Functions.
Bayes Rule, Naïve Bayes Model.
Decision Tree – ID3.

Module 3:

Clustering Methods: K-means Clustering, Hierarchical Clustering Techniques, Density-Based Clustering
Feature Selection Techniques: Entropy, Correlation Coefficient, Chi-square Test, Forward and Backward Selection.
Dimensionality Reduction: PCA, LDA, t-SNE.
Maximum Likelihood Estimation Technique.

Module 4:

Basics of Graphical Models: Bayesian Networks, Hidden Markov Model.
Ensemble Methods: Boosting, Bagging, Random Forest, XGBoost

Module 5:

Neural Networks – Concept of Perceptron and Artificial Neuron.
Feed Forward Neural Network, Back Propagation Algorithm.
Convolutional Neural Networks (CNNs) for image classification and object detection.
Recurrent Neural Networks (RNNs),Long Short-Term Memory (LSTM) networks for sequential data processing, Autoencoders.
Transformer Models (BERT, GPT) for NLP applications.

Labs / Practicals:

1. Implement supervised, unsupervised, and semi-supervised learning on sample datasets and identify the type of machine learning task.
2. Apply cross-validation and evaluate classifier performance. Calculate accuracy, precision, recall, F1-score, and ROC-AUC metrics of models.
3. Perform data preprocessing using standardization, normalization, label encoding, and one-hot encoding and observe its effect on model accuracy.
4. Implement linear, Ridge, and Lasso regression models on a dataset and compare their performance using MSE.
5. Apply K-Nearest Neighbor, Logistic Regression, and SVM classifiers on a dataset and visualize decision boundaries.
6. Train and evaluate a Decision Tree using the ID3 algorithm and a Naïve Bayes classifier on a classification dataset.
7. Implement and compare K-means, hierarchical clustering, and DBSCAN on a dataset using visualization and silhouette scores.
8. Perform feature selection using entropy, correlation coefficient, and chi-square test, and evaluate model accuracy with selected features.
9. Apply PCA, LDA, and t-SNE for dimensionality reduction and visualize data in 2D space.
10. Build a Hidden Markov Model to predict sequences and evaluate performance using real or synthetic sequential data.
11. Train and compare Random Forest, Bagging, AdaBoost, and XGBoost models on a classification problem.
12. Implement a feedforward neural network with backpropagation to classify handwritten digits using the MNIST dataset.
13. Build a Convolutional Neural Network (CNN) for image classification using CIFAR-10 or MNIST and evaluate performance.
14. Train an LSTM model for time-series prediction and use a pre-trained BERT/GPT model for sentiment analysis in NLP.

References:

1. Ethem Alpaydın, Introduction to Machine Learning , MIT Press, 2014.
2. Kevin Murphy, Machine Learning: A Probabilistic Perspective (MLAPP), MIT Press,2012.
3. Han, Jiawei, and Micheline Kamber. Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann Publishers.
4. Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006.

Course Code	Course Name	L	T	P	Cr
DOEE250058	In Vehicle Communication Protocols	3	0	2	4

Course Outcomes:

CO 1 Understand the architecture and need for in-vehicle communication protocols.
CO 2 Understand and analyze CAN and LIN protocols and their practical applications.
CO 3 Understand and apply diagnostics using UDS and OBD-II over communication buses.
CO 4 Analyze FlexRay and Automotive Ethernet communication stacks and real-time data exchange.
CO 5 Design in-vehicle network architectures for ECUs using multiple protocols and evaluate performance.
CO 6 Apply protocol testing and debugging using Vector tools (CANoe /CANalyzer) or equivalent.

Module 1:

Introduction to In-Vehicle Networking
- Evolution of in-vehicle networks: from point-to-point to multiplexing
- Layered architecture: OSI and AUTOSAR perspectives
- Signal encoding methods and bit timing basics
- ECU architectures and integration into IVNs

Module 2:

CAN (Controller Area Network)
- CAN architecture: message framing, arbitration, error detection
- CAN physical and data link layer
- CAN FD enhancements
- Introduction to UDS (ISO 14229) over CAN
- Real-world examples: Body control module, Instrument cluster

Module 3:

LIN (Local Interconnect Network)
- LIN protocol overview and its role as sub-network to CAN
- LIN cluster topology, master/slave roles
- LIN frame structure and schedule tables
- Use cases in power windows, seat control, etc.

Module 4:

FlexRay and Automotive Ethernet
- FlexRay protocol: dual channels, time-triggered vs event-triggered
- FlexRay topologies, frame format, synchronization
- Automotive Ethernet: Time Sensitive Networking (TSN), 100BASE-T1
- Comparison with traditional automotive protocols

Module 5:

Diagnostics, Tools and Protocol Design
- OBD-II and diagnostic protocol basics
- Diagnostic communication over CAN, K-Line
- Hands-on toolchain: Introduction to CANoe, CANalyzer or open-source tools
- Design of multi-protocol in-vehicle communication architecture
- Security, redundancy, and fail-safe considerations

Labs / Practicals:

Practical Sessions on protocol testing and debugging using Vector tools (CANoe/CANalyzer) or equivalent.

References:

1. Paret, Dominique. Multiplexed Networks for Embedded Systems: CAN, LIN, FlexRay, Safe-by-Wire. Wiley, 2007.
2. Agarwal, Rajesh. In-Vehicle Networking. SAE International, 2015.
3. Navet, Nicolas and Simonot-Lion, Françoise. Automotive Embedded Systems Handbook. CRC Press, 2009.
4. Pfeiffer, Ayre, and Keydel. Embedded Networking with CAN and FlexRay. Springer.
5. Official Vector Documentation (for CANoe/CANalyzer tools)

Course Code	Course Name	L	T	P	Cr
DCSE250020	Cloud Computing	3	0	2	4

Course Outcomes:

CO1: Explain the evolution, architecture, and key characteristics of cloud computing, including service models (IaaS, PaaS, SaaS) and deployment types (public, private, hybrid).

CO2: Describe various types and implementation levels of virtualization, including CPU, memory, storage, and network virtualization in cloud environments.

CO3: Analyze layered cloud architecture and address design challenges related to inter-cloud resource management and platform deployment.

CO4: Apply distributed and parallel programming paradigms such as MapReduce and Hadoop for cloud-based application development and understand various cloud platforms like AWS, OpenStack, and Google App Engine.

CO5: Evaluate cloud security challenges and solutions, including data and application security, VM security, and risk management strategies in cloud environments.

Module 1:

Introduction to Cloud Computing

Evolution of Cloud Computing, System Models for Distributed and Cloud Computing, NIST Cloud Computing Reference Architecture, Features of Cloud Computing, Cloud Services – IaaS, PaaS, SaaS, Cloud service Providers – Public , Private and Hybrid Clouds.

Module 2:

Introduction to component virtualization

Basics of Virtualization, Types of Virtualization, Implementation Levels of Virtualization, Virtualization of CPU, Memory, I/O Devices, Desktop Virtualization, Server Virtualization, Storage Virtualization, Network Virtualization.

Module 3:

Architectural Design of Compute and Storage Clouds

Layered Cloud Architecture Development, Design Challenges, Inter Cloud Resource Management, Resource Provisioning and Platform Deployment, Global Exchange of Cloud Resources.

Module 4:

Parallel and Distributed Programming Paradigms

Map Reduce, Twister and Iterative MapReduce, Hadoop Library from Apache, Mapping Applications, Programming Support, Google App Engine, Amazon AWS, Cloud Software Environments - Eucalyptus, Open Nebula, OpenStack.

Module 5:

Security Overview

Cloud Security Challenges, Software-as-a-Service Security, Security Governance, Risk Management, Security Monitoring, Security Architecture Design, Data Security, Application Security, Virtual Machine Security.

Labs / Practicals:

1. Exploring Cloud Deployment and Service Models
2. Simulating Public, Private, and Hybrid Clouds
3. Virtualization of CPU and Memory
4. Network and Storage Virtualization
5. Deployment of a Layered Cloud Architecture
6. Resource Provisioning and Scaling
7. Deploying a Web Application
8. Simulating Security Attacks and Defenses in a Cloud VM
9. Implementing Access Control and Encryption on Cloud Storage

References:

[1] R. Buyya, C. Vecchiola, and T. Selvi, Mastering Cloud Computing: Foundations and Applications Programming. New Delhi, India: McGraw-Hill Education, 2013.
[2] A. T. Velte, T. J. Velte, and R. Elsenpeter, Cloud Computing: A Practical Approach. New Delhi, India: McGraw-Hill Education, 2010.
[3] K. Hwang, G. C. Fox, and J. J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Burlington, MA, USA: Morgan Kaufmann, 2012.
[4] T. Erl, R. Puttini, and Z. Mahmood, Cloud Computing: Concepts, Technology & Architecture. Boston, MA, USA: Pearson Education, 2013.
[5] G. Reese, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. Sebastopol, CA, USA: O'Reilly Media, 2009.

Course Code	Course Name	L	T	P	Cr
DCSE250056	Data Structures and Algorithms	3	0	2	4

Course Outcomes:

1)Classify Data structures
2)Describe Linear Data Structures
3)Explain Non-Linear Data Structures
4)Explain basic algorithmic concepts and recursion
5)Apply different Sorting and Searching Algorithms

Module 1:

Introduction to Data Structures:Basic Terminology, Classification of Data Structures, Operations on Data Structures

Module 2:

Linear Data Structures: Arrays: Introduction to Arrays, Representation in Memory, Operations on an Array, Two Dimensional Arrays
Linked Lists: Singly Linked List, Representation in Memory, Operations on a Single Linked List, Circular Linked Lists, Doubly Linked Lists.
Stacks: Introduction to Stacks, Array Representation of Stacks, Operations on a Stack, Linked List Representation and Operations of Stack, Applications of Stacks-Infix-to-Postfix Transformation, evaluating Postfix Expressions.
Queues: Introduction to Queues, Array Representation of Queues, Operations on a Queue,
Linked List Representation and Operations of Queue, Types of Queues-DeQueue, Circular Queue, Applications of Queues-Round Robin Algorithm

Module 3:

Non Linear Data Structures:
Trees: Basic Terminologies, Definition and Concepts of Binary Trees, Representations of a Binary Tree using Arrays and Linked Lists, Operations on a Binary Tree-Insertion, Deletion, Traversals, Types of Binary Trees.
Graphs: Graph Terminologies, Representation of Graphs- Set, List, Matrix, Graph Traversals

Module 4:

Introduction to Algorithms:
Algorithms and flow charts, Time & Space complexity (definition only)
Recursion: Basic concepts and examples of recursion e.g. factorial problem, Fibonacci sequence.

Module 5:

Sorting & Searching Algorithms: Sorting Algorithms: Algorithms and their analysis (time and space) — Bubble sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Heap sort and Radix Sort
Searching Algorithms: Linear search — Binary search –Concept of Hashing.

Labs / Practicals:

1.Write a program to implement array
2.Write a program to add two matrices using two dimensional arrays
3.Write a program using recursive and non-recursive functions to perform search operation in a given list of integers using linear search technique
4.Write a program to implement search operation in a given list of integers using binary search technique

5.Write a program to implement insertion sorting for a given random data
6.Write a program to implement bubble sorting for a given random data
7.Write a program to implement quick sorting for a given random data
8.Write a program to implement selection sorting for a given random data.

References:

1)Data Structures, R.S. Salaria, Khanna Book Publishing, New Delhi
2)Data Structures Using C, Reema Thareja, Oxford University Press India.
3)Classic Data Structures, SamantaDebasis, Prentice Hall of India.
4)Fundamentals of Data Structure in C, Horowitz, Ellis, Sahni, Sartaj, AndersonFreed, Susan, University Press, India.
5)Data Structures: A Pseudo code approach with C, Richard F. Gilberg, Behrouz A. Forouzan, CENGAGE Learning, India.
6)Data Structures and Algorithms: Concepts, Techniques and Applications, G. A. V. Pai, McGraw- Hill Education, India.
7)Introduction to Algorithms, T.H. Cormen, C.E. Leiserson, R. L. Rivest, C. Stein, MIT Press

Course Code	Course Name	L	T	P	Cr
DOEE250090	Embedded Programming	3	0	2	4

Course Outcomes:

CO1: Understand fundamental concepts of programming, algorithms, flowcharts, and the basics of computation for effective problem-solving and logical thinking.

CO2: Apply C programming constructs including data types, operators, control statements, functions, arrays, and pointers to develop modular programs.

CO3: Develop and manipulate advanced C structures such as unions, linked lists, and other data structures, with applications in embedded system contexts.

CO4: Demonstrate object-oriented programming concepts using C++, including class hierarchies, templates, and exception handling for robust code development.

CO5: Analyze and apply embedded programming standards such as MISRA C/C++, use compiler directives, and perform code optimization and profiling for reliable and efficient embedded software development.

CO6: To understand Programming & algorithms for problem solving

CO7: To understand the Programming Concepts

CO8:Build Application using High level languages C/C++ Programming

Module 1:

Introduction to Programming & algorithms for problem solving

The Basic Model of Computation, Algorithms, Flow-charts, Programming Languages, Compilation, Linking and Loading, Testing and Debugging, Algorithms for Problem Solving: Decimal Base to Binary Base conversion, Reversing digits of an integer, GCD (Greatest Common Division), LCM etc. , Use case diagrams and UML.

Module 2:

C Programming Basics

GNU Tools: gcc, gdb, gprof, Makefiles, Basic data types, operations, and flow control (decision-making statements), Flow control (loops), typecasting, and computer logic, Switch-case, arrays, and the basics of strings, pointers, functions, storage class.

Module 3:

Advanced C programming for Embedded Systems

Structure and union, Linear and nonlinear data structures, Linked List

Module 4:

Object Oriented Programming concepts with C++
Overview of C++, , Fundamentals of the object-oriented approach, Class hierarchy, Advanced class concepts, Templates, Accessing data and dealing with exceptions.

Module 5:

Embedded Programming Coding standards & Concepts
Coding Standards For Compliance , ANSI C, C89, C95, C99, C11, MISRA C & C++, Profiling & Optimisation

techniques, Pragma , floating-point exceptions rounding multi-precision libraries (GMP, MPFR, MPIR), IEEE754, Static and dynamic Code analysis.

Labs / Practicals:

Develop algorithms/ Flow diagrams for any one from the following
1.Decimal Base to Binary Base conversion
2.Reversing digits of an integer
3. GCD Greatest Common Division)
4. LCM

Familiarisation of UML diagrams With the help of open source UML tools
1. Draw UML diagrams
2. Export diagrams
3. Share diagrams using Eclipse
4. Create new, custom UML elements
5. Build sequence and activity diagrams from plain text

C Programming using Linux platforms
1. Familiarisation of Linux Commands
2. Example usage of gcc compiler with sample C programs
3. C programming covering data types, operators, loops, strings

C Programming using Linux platforms
1. Examples using Pointers using C
2. Examples using functions, pass by value, pass by reference methods.
3. Examples covering the structure and union

Advanced C Programming for Embedded Systems (Linear & Nonlinear data structure implementation using C Language)
1. Write a c program to implement queue and its operations
2. Write a C Program to implement stack and its operation
3. Write a C program to implement singly linked list and its operations
4. Write a C program to implement circular linked lists and its operations.

Class hierarchy C++ programming
1. Write program to understand the class concepts in C++
2. Write C++ program to cover inheritance concepts in C++
3. Write C++ program to cover overloading concepts in C++.

Advanced class concepts using C++ programming
1. Write C++ program to cover virtual functions in C++.
2. Write C++ program to cover Templates, Accessing data and dealing with exceptions.

References:

Text Books
1. C Programming language, Kernighan, Brian W, Ritchie, Dennis M, Prentice Hall PTR
2. “Embedded C”,Michael J. Pont, Addison Wesley Reference

Reference Books
1. The Complete Reference C++, Herbert Schildt, TMH
2. GNU C++ For Linux, Tom Swan , PrenticeHall India
3. Daniel W. Lewis, “Fundamentals of embedded software where C and assembly meet”, Pearson Education, 2002.

Course Code	Course Name	L	T	P	Cr
DOEE250081	Autonomous Vehicle Systems and AI Integration	3	0	2	4

Course Outcomes:

CO1: Understand autonomous vehicle architectures, SAE levels, and key subsystems including drive-by-wire and electric powertrains.
CO2: Apply sensor technologies and AI-based perception for environment sensing and object tracking.
CO3: Implement localization and mapping using GPS, IMU, sensor fusion, SLAM, and HD maps.
CO4: Develop path planning and AI decision-making using algorithms, motion planning, and reinforcement learning.
CO5: Integrate AI in control systems, deploy on real-time platforms, and ensure safety and validation compliance.

Module 1:

Autonomous Vehicles

Overview and Levels of Autonomy (SAE Levels), Functional Architecture of Autonomous Vehicles, Key Subsystems: Perception, Localization, Planning, and Control. Drive-by-Wire Systems, Introduction to Electric Powertrains for Autonomous Vehicles.

Module 2:

Sensors and Perception Systems

Sensor Technologies: LIDAR, Radar, Ultrasonic, Cameras. Sensor Models and Calibration, 3D Environment Perception: Point Cloud Processing, Object Detection and Tracking, AI for Image and Video Processing using CNNs.

Module 3:

Localization and Mapping

GPS, IMU and Sensor Fusion, SLAM (Simultaneous Localization and Mapping), Kalman and Particle Filters, Lane Detection and Road Mapping, HD Maps and Real-time Map Updates.

Module 4:

Path Planning and AI Decision Making

Route Planning Algorithms: A*, Dijkstra, Behavior Planning: Finite State Machines (FSM), Decision Trees, Motion Planning: RRT, MPC (Model Predictive Control), AI Techniques in Planning: Reinforcement Learning, Deep Q-Learning, Ethical and Social Aspects in Decision Making.

Module 5:

AI Integration and System Implementation

AI in Control Systems, Deep Learning Architectures for Autonomous Systems, Real-Time Computing Platforms: NVIDIA Jetson, ROS2, AUTOSAR, V2X Communication and Edge AI, Validation, Testing and Safety Standards (ISO 26262, SOTIF).

Labs / Practicals:

Study of Tesla Autopilot/ Waymo system architecture, Hands-on Sessions with OpenCV, ROS, and Python, MATLAB/Simulink. Design and simulation of an Autonomous Vehicles scenario.

References:

[1] S. Liu, L. Y. Wang, G. Yin, and C. Hu, Creating Autonomous Vehicle Systems. San Rafael, CA: Morgan & Claypool, 2017.
[2] R. Madhavan, Intelligent Vehicle Technologies: Theory and Applications. Boca Raton, FL: CRC Press, 2015.
[3] P. Koopman, How Safe is Safe Enough? Measuring and Predicting Autonomous Vehicle Safety. Cham, Switzerland: Springer, 2022.
[4] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, Introduction to Autonomous Mobile Robots, 2nd ed. Cambridge, MA: MIT Press, 2011.
[5] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics. Cambridge, MA: MIT Press, 2005.

Course Code	Course Name	L	T	P	Cr
DOEE250088	Embedded OS and RTOS	3	0	2	4

Course Outcomes:

CO 1 Attain comprehensive understanding of Embedded OS (Linux) fundamentals
CO 2 Attain comprehensive understanding of Embedded Processor and its software
CO 3 Attain comprehensive understanding of Embedded Linux Drivers
CO 4 Attain comprehensive understanding of basics of real-time concepts
CO 5 Attain comprehensive understanding of incorporating RTOS in an embedded system
CO 6 Attain comprehensive understanding of applying the knowledge for developing practical applications of modern real-time systems
CO 7 Get knowledge in Embedded OS (Linux) fundamentals
CO 8 Understand Embedded Linux Internal programming
CO 9 Comprehend Embedded Linux Drivers
CO 10 Learn about the basics of real-time OS Programming concepts

Module 1:

Introduction to Embedded Linux

Overview of Linux OS, Directory structures, basic Linux shell commands, Overview of Systems Calls, Classification of system Calls, Inter Process Communication, Multithreading and Thread Management

Module 2:

Embedded Linux Driver Development

Linux Kernel Module Programming, Character device driver development, USB device driver development, Block, Network and PCI device driver development

Module 3:

Building and Customisation of Linux for Embedded systems

Linux booting procedure, Bootloader, hypervisor, opensbi, qemu, Linux build tools - Buildroot and Yocto

Module 4:

Overview of Real Time OS

Basics of RTOS: Real-time concepts, Hard Real time and Soft Real-time, Differences between General Purpose OS & RTOS, Basic architecture of an RTOS, Scheduling Systems, RTOS Issues – Selecting a Real Time Operating System

Module 5:

RTOS for Embedded Applications

FreeRTOS, Thread creation & Management, Inter thread Communication, Mutual Exclusion, MbedOS, Other real time OS (VxWorks, Azure RTOS, SAFERTOS etc.)

Labs / Practicals:

Embedded Linux shell commands
Embedded Linux System Call programming
Embedded Linux Interprocess Communication
Development of Multithreaded application in Linux
Performing Thread Synchronization and resource protection (Mutex) based applications in Linux
Develop simple Kernel Module Programming
Develop basic character Device driver in Linux
Familiarization of FreeRTOS, Thread creation, synchronization - Port to FreeRTOS to ARM Cortex M4 development board

References:

1. GNU/LINUX Application Programming, Jones, M Tims
2. Sreekrishnan Venkateswaran Essential Linux Device Drivers, Prentice Hall 2008
3. Embedded/Real Time Systems Concepts, Design and Programming Black Book, Prasad, KVK
4. Software Design for Real-Time Systems: Cooling, J E Proceedings of 17the IEEE Real-Time Systems Symposium December 4-6, 1996 Washington, DC: IEEE Computer Society
5. FreeRTOS Reference Manual

Course Code	Course Name	L	T	P	Cr
DOEE250194	Vehicle Security and Comfort Systems	3	1	0	4

Course Outcomes:

CO1: Explain the function, design, and regulations of vehicle locking, anti-theft, and alarm systems.
CO2: Describe the construction and applications of acoustic signaling devices in vehicles.
CO3: Analyze occupant protection systems such as seat belts, pretensioners, and airbags.
CO4: Demonstrate the working of power and comfort drive systems including windows, sunroof, and seat adjustment.
CO5: Evaluate the role of ADAS, sensors, and data fusion in enhancing vehicle safety and convenience.

Module 1:

Vehicle Locking & Anti-Theft Systems

Function, structure, operating principle, Open by wire, Electrical locking system, Central locking system, Electronic vehicle immobilizer, functional description Comfort Entry/Go system, Regulations, Permissible alarm signals. System design, alarm detectors, Alarm system control unit, Alarm siren, Tilt sensor, Interior monitoring.

Module 2:

Signaling and Acoustic Systems

Acoustic signaling devices, applications, Horn, Fanfare horns, Integrated signaling functions.

Module 3:

Occupant Protection & Safety Systems

Occupant Protection Systems: Seat belts, seat-belt pretensioners, Airbags: front, side, components, Rollover protection system.

Module 4:

Power and Comfort Drives

Power Window Drives: Power window motors and controls, Power sunroof drives
Comfort Systems: Electrical seat adjustment, Steering-column adjustment, Multipurpose actuator.

Module 5:

Advanced Driver Assistance Systems (ADAS)

Critical driving situations and accident causes, Applications in convenience and safety, Sensors for all-round electronic visibility, Sensor data fusion, Integration with locking and protection systems.

Labs / Practicals:

N/A

References:

Text Book

R. Bosch GmbH, Automotive Handbook, 9th ed. Hoboken, NJ, USA: Wiley, 2015.

Reference Books

R. Bosch GmbH, Safety, Comfort & Convenience Systems, 7th ed. Hoboken, NJ, USA: Wiley, 2007.

Course Code	Course Name	L	T	P	Cr
DOEE250089	Automotive Power Electronics and Motor Drives	3	1	0	4

Course Outcomes:

CO1: Explain the operation, characteristics, and protection of power semiconductor devices used in automotive and industrial applications.
CO2: Analyze the working of AC-DC converters, DC-AC inverters, and chopper circuits for power control in electric and hybrid vehicles.
CO3: Evaluate the performance and interface of ultracapacitors in combination with batteries for automotive energy storage systems.
CO4: Apply control techniques for various electric motors including induction, DC, BLDC, and SRM motors used in automotive drive systems.
CO5: Design and troubleshoot basic power electronic circuits and motor drive systems using appropriate methods and tools.

Module 1:

Power Electronics and Devices

Introduction to Power Electronics, Automotive semiconductor devices: SCR, Power Transistor, Power MOSFET, IGBT, Turn-on and turn-off circuits, Series and parallel operation of SCR, Protection circuits and Snubber Circuit Design.

Module 2:

Converters and Inverters

Half-wave and fully-controlled converters (R, RL, RLE load), Three-phase half-wave and fully-controlled converters, Voltage Source Inverters (120° & 180° conduction modes), Current Source Inverters, PWM techniques

Module 3:

Choppers and Ultracapacitors.

Step-up and step-down choppers, Types and applications of choppers.
Ultracapacitors: Theory of electronic double-layer capacitance, Sizing criteria and cell balancing, Converter interface, Integration with batteries.

Module 4:

Automotive Motor Control

Speed control methods, Induction motor control techniques, DC motor control techniques, Sensor-based control basics (current/speed feedback).

Module 5:

Automotive Drive Systems

BLDC Motor: Construction, characteristics, operation, Open-loop and closed-loop control, Switched Reluctance Motor (SRM): Construction, operation, and applications.

Labs / Practicals:

N/A

References:

Text Book
P. S. Bimbhra, Power Electronics, 14th ed. New Delhi, India: Khanna Publishers, 2015.

Reference Books
M. H. Rashid, Power Electronics: Circuits, Devices and Applications, 4th ed. Boston, MA, USA: Pearson, 2013.
L. Umanand, Power Electronics: Essentials and Applications. New Delhi, India: Wiley India Pvt. Ltd., 2009.
R. Krishnan, Electric Motor Drives: Modeling, Analysis, and Control. Upper Saddle River, NJ, USA: Prentice Hall, 2001.
A. M. Trzynadlowski, Fundamentals of Electric Drives, 2nd ed. Boca Raton, FL, USA: CRC Press, 2013.
T. Denton, Automotive Electrical and Electronic Systems, 5th ed. London, U.K.: Routledge, 2017.
C. C. Chan and K. T. Chau, Modern Electric Vehicle Technology. Oxford, U.K.: Oxford University Press, 2001.
M. Barak, Energy Storage Systems in Electronics. Cambridge, U.K.: Cambridge University Press, 2009.
A. Burke, “Ultracapacitors: Why, how, and where is the technology,” J. Power Sources, vol. 91, no. 1, pp. 37–50, Nov. 2000.

Course Code	Course Name	L	T	P	Cr
DOEE250091	Software and ISO Standards in Automotive Systems	3	1	0	4

Course Outcomes:

CO1:Describe the architecture, components, and requirements of AUTOSAR-based automotive software systems.
CO2: Analyze the functionality and configuration of the AUTOSAR communication stack, including network management and system services.
CO3:Apply AUTOSAR implementation concepts such as memory mapping, platform types, and software integration techniques.
CO4:Explain the structure and lifecycle of the ISO 26262 standard and apply functional safety analysis techniques such as ASIL classification and hazard analysis.
CO5:Evaluate the principles of ISO/TS 16949 and apply quality assurance practices in the design and development of automotive software systems.

Module 1:

AUTOSAR and Basic Software Standards

Overview of AUTOSAR architecture, General requirements for Basic Software Modules, Functional and fault operations, Error detection and handling mechanisms.

Module 2:

AUTOSAR Communication Stack and System Services

Communication Stack Overview, Network Management and TTCAN Interface Standards, TTCAN Drivers, System Services: Watchdog Manager, Synchronized Time Base Manager.

Module 3:

AutoSAR – Implementation and Integration

AUTOSAR Platform Types, Memory Mapping concepts, Software component integration and configuration, ECU abstraction and MCAL basics.

Module 4:

ISO 26262 Standard – Concepts and Implementation

Structure and Vocabulary of ISO 26262, Management of Functional Safety, Concept Phase, Product Development at System, Hardware, and Software Levels, Production, Operation, and Supporting Processes, ASIL-oriented and safety-oriented analysis.

Module 5:

ISO/TS 16949 and Functional Safety Case Studies

Overview of ISO/TS 16949 and its relevance to automotive software quality, Data transmission systems and modulation techniques, Noise, interference, and grounding (signal circuit and shield), Isolation techniques: Capacitive, Magnetic, Optical, ISO 26262: Case studies, Hazard Analysis and Risk Assessment, Safety goals, preliminary architecture, and functional safety concept.

Labs / Practicals:

N/A

References:

Text Book

D. Hoyle, Automotive Quality Systems, Oxford, U.K.: Butterworth-Heinemann Ltd., 2015.

Reference Books

H. Heineke, Functional Safety for Road Vehicles: New Challenges and Solutions for E-mobility and Automated Driving, Cham, Switzerland: Springer, 2021.
M. Lukasiewycz and H. Hamann, Automotive System Design: AUTOSAR Fundamentals and Applications, Cham, Switzerland: Springer, 2020.
ISO, ISO 26262 Road Vehicles – Functional Safety, 2nd ed., Geneva, Switzerland: International Organization for Standardization, 2018. [Standard Document]
R. Karam, AUTOSAR Compendium – Part 1: Application & RTE, 4th ed., Stuttgart, Germany: CreateSpace Independent Publishing Platform, 2018

Course Code	Course Name	L	T	P	Cr
DOEE250093	Automotive Engine Management and Sensors Systems	3	1	0	4

Course Outcomes:

CO1: Understand ECU architecture, design process, and engine control requirements for efficiency and emission compliance.
CO2: Analyze petrol and diesel engine management systems including ignition, fuel injection, and EGR control.
CO3: Evaluate emission control techniques, standards, and the impact of alternative fuels on engine management.
CO4: Explain working principles, characteristics, and signal conditioning of automotive transducers and sensors.
CO5: Apply and integrate automotive sensors with ECU for vehicle body, powertrain, chassis, and ADAS applications.

Module 1:

ECU Design and Engine Control

Introduction to Electronic Control Units (ECU), Need and evolution of ECUs in automotive applications, ECU architecture: analog and digital interfaces, ECU design complexities and V-model development process, Basics of IC engines (Petrol & Diesel), Need for engine controls for efficiency, performance, and emission control, Objectives and advantages of electronic engine management systems.

Module 2:

Petrol and Diesel Engine Management Systems

Petrol engine controls: evolution and architecture, Electronic ignition systems, Multi-point and direct fuel injection systems, Diesel engine controls: basics and evolution, In-line and rotary fuel pumps, Electric fuel pumps and EGR control, Electronic control of fuel injection timing.

Module 3:

Emission Control and Alternative Fuels

Sources and types of automotive emissions, Indian emission standards and testing methods, Exhaust after-treatment: Exhaust Gas Recirculation (EGR), Catalytic converters.

Introduction to alternative fuels: Hydrogen, CNG, LPG, Biodiesel, Impact of alternative fuels on engine management systems.

Module 4:

Transducers and Sensors

Classification and working principles of transducers, Input-output configuration, Static and dynamic characteristics, Variable resistance transducers, Strain gauges (metal and semiconductor) and signal conditioning, Inductive, electromagnetic, Hall-effect, capacitive, and piezoelectric sensors, Ultrasonic sensor principles and signal processing.

Module 5:

Automotive Sensor Applications and Integration

Sensors used in automotive systems:

- Vehicle Body: Torque, flap airflow, temperature, ultrasonic, radar (ACC)

Powertrain: Fuel level, speed & RPM, Lambda (O_2), hotwire air mass meter
Chassis: Steering angle, vibration & acceleration, pressure sensors

Sensor integration with ECU and control systems, Trends in automotive sensing for ADAS and smart powertrains.

Labs / Practicals:

N/A.

References:

Text Book
H. N. Gupta, Fundamentals of Internal Combustion Engines, 2nd ed. New Delhi, India: PHI Learning Pvt. Ltd., 2015.
V. Ganesan, Internal Combustion Engines, New Delhi, India: Tata McGraw-Hill Education, 2012.
J. Turner and J. Watson, Automotive Sensors (Sensors Technology). London, U.K.: Momentum Press, 2009.

Reference Books
W. Ribbens, Understanding Automotive Electronics, 8th ed. Oxford, U.K.: Butterworth-Heinemann, 2017.
R. Bosch GmbH, Bosch Automotive Handbook, 10th ed. Stuttgart, Germany: Springer Vieweg, 2018.
T. Denton, Automobile Electrical and Electronic Systems, 5th ed. London, U.K.: Routledge, 2017.
C. Blake and F. T. Ulaby, Sensors and Control Systems in Manufacturing, 1st ed. Hoboken, NJ, USA: CRC Press, 1998.
S. M. Boser, Analog Design Essentials, Dordrecht, Netherlands: Springer, 2007.