The following colour represents the syllabus revision, skill, employability and entrepreneurship.

Green : Skill

Pink : Employability

Sky : Entrepreneurship

Industrial IoT and Automation

Code	Course Title	Credit	T-P-PJ
CUTM1017	Industrial IoT and Automation	6	3-2-1

Objective

- Students will learn the new evolution in hardware, software, and data.
- While the promise of the Industrial Internet of Things (IIoT) brings many new business prospects, it also presents significant challenges ranging from technology architectural choices to security concerns.
- Students acquire upcoming Industrial Internet of Things: Roadmap to the Connected World Course offers important insights on how to overcome these challenges and thrive in this exciting space.

Learning outcome

- Discover key IIoT concepts including identification, sensors, localization, wireless protocols, data storage and security
- Explore IoT technologies, architectures, standards, and regulation
- Realize the value created by collecting, communicating, coordinating, and leveraging the data from connected devices
- Examine technological developments that will likely shape the industrial landscape in the future
- Understand how to develop and implement own IoT technologies, solutions, and applications
- At the end of the program, students will be able to understand how to develop and implement their own IoT technologies, solutions, and applications.

Course content

Module I: Introduction and Architecture **(6 hours)**

Theory

What is IIoT and connected world? the difference between IoT and IIoT, the web of things, architecture of IIoT.

Practice: Hardware/MATLAB

1. Simulation of RFID using Matlab/Dymola.

Module II: Communication Technologies of IIoT **(10 hours)**

Theory

Industry standards communication technology (LoRAWAN, OPC UA, MQTT), connecting into existing Modbus and Profibus technology, wireless network communication, security issues in IIoT.

Practice:

1. Demonstration of MQTT communication using Matlab/Dymola.
2. Site visit to Apparel factory in the Bhubaneswar campus.
3. Wireless communication demonstration using Matlab/Dymola.

Module III: Visualization and Data Types of IIoT **(8Hours)**

Theory

HMI in an Industrial IoT world, front end EDGE devices, enterprise data for IIoT, emerging descriptive data standards for IIoT, cloud data base.

Practice:

1. Assembling the HMI for IIoT environment using Matlab/Dymola.
2. Measurement of temperature & pressure values of the process using sensors.

Module IV: Automation **(8 Hours)**

Theory

Automation definition, automation pyramid, field level sensors, Embedded sensors, HMI in an automation process.

Practice:

1. Visualization of diverse sensor data using dashboard (part of IoT's 'control panel')
2. Wearable sensing for IoT (future user interfaces for IoT - new ways to control and interact with your environment)

Module V:Control & Supervisory Level of Automation **(8 Hours)**

Theory

Programmable logic controller (PLC), real-time control system, Supervisory Control & Data Acquisition (SCADA).

Practice:

1. Simulation of PLC to understand the control concept.
2. SCADA HMI demonstration using Matlab.
3. SCADA simulation using Matlab/Dymola.

Module VI: Planning Level & Management Level **(4Hours)**

Theory

Manufacturing execution system (MES), enterprise resource planning(ERP), production control.

Practice:

1. Designing MES system by using Adobe.

Projects:

1.A smart meter is an internet-capable device that measures energy.
2.Building connectioninto existing Modbus & Profibus networks.
3.Monitoring environmental conditions in an apparel factory space.
4. Predictive monitoring of CNC machine operation.

Text Books:

- The Internet of Things in the Industrial Sector, Mahmood, Zaigham (Ed.)(Springer Publication)
- Industrial Internet of Things: Cybermanufacturing System, Sabina Jeschke, Christian Brecher, Houbing Song, Danda B. Rawat(Springer Publication)

Note: 1 credit theory=10 hrs lecture, 1 credit practice/project=12.5 hrs lab/workshop/field work in a semester

Code	Course Title	Credit	T-P-PJ
CUTM1018	Data Analysis and Visualisation Using Python	4	0-1-3

Objective

- How to tell a story from data
- How to marshal the data for storyline
- The ability to develop visualisation to tell the story
- The focus is on analysis of data using visualisation as a tool

Learning outcome

- To create impactful visualization with good story line.

Course content

Module-I

STORY BOARD DEVELOPMENT

The objective and flow of the story to be understood through cases

Module-II

DATA READING USING PYTHON FUNCTIONS

Python libraries: Pandas, NumPy, Plotly, Matplotlib, Seaborn, Dash

Data collection from online data sources, Web scrap, data formats such as HTML, CSV, MS Excel, data compilation, arranging and reading data, data munging

Module-III

DATA VISUALSATION USING PYTHON LIBRARIES

Different graphs such as Scatterplot, Line chart, Histogram, Bar chart, Bubble chart, Heatmaps etc.

Dashboard Basics – Layout, Reporting, Infographics, Interactive components, live updating

Projects List

1. COVID 19
2. World Development Indicators
3. ERP dashboarding
4. Details of Social/ Empowerment schemes of Govt. etc.

References:

https://www.programmer-books.com/wp-content/uploads/2019/04/Python-for-Data-Analysis-2nd-Edition.pdf

https://towardsdatascience.com/data-visualization/home

Reading materials and videos available on internet on how to use ANACONDA, JUPYTER NOTEBOOK and Python Libraries

Machine Learning using Python

Code	Course Title	Credit	T-P-PJ
CUTM1019	Machine Learning using Python	4	1-2-1

Objective

- Understand the meaning, purpose, scope, stages, applications, and effects of ML.
- Explore important packages of python, such as numpy, scipy, OpenCV and scikit-learn.

Learning outcome

- Students will able to Create and incorporate ML solutions in their respective fields of study.

Course content

Module 1 – Application and Environmental-setup (12 hrs)

- Applications of Machine Learning In different fields (Medical science, Agriculture, Automobile, mining and many more).
- Supervised vs Unsupervised Learning based on problem Definition.
- Understanding the problem and its possible solutions using IRIS datasets.
- Python libraries suitable for Machine Learning(numpy, scipy, scikit-learn, opencv)
- Environmental setup and Installation of important libraries.

Module 2 - Regression (8 hrs)

- Linear Regression
- Non-linear Regression
- Model Evaluation in Regression
- Evaluation Metrics in Regression Models
- Multiple Linear Regression
- Feature Reduction using PCA
- Implementation of regression model on IRIS datasets.

Module 3 - Classification (24 hrs)

- Defining Classification Problem with IRIS datasets.
- Mathematical formulation of K-Nearest Neighbour Algorithm for binary classification.
- Implementation of K-Nearest Neighbour Algorithm using sci-kit learn.
- Classification using Decision tree.
- Construction of decision trees based on entropy.
- Implementation of Decision Trees for Iris datasets .
- Classification using Support Vector Machines.
- SVM for Binary classification
- Regulating different functional parameters of SVM using sci-kit learn.
- SVM for multi class classification.
- Implementation of SVM using Iris datasets .
- Implementation of Model Evaluation Metrics using sci-kit learn and IRIS datasets.

Module 4 - Unsupervised Learning (12 hrs)

- Defining clustering and its application in ML .
- Mathematical formulation of K-Means Clustering.
- Defining K value and its importance in K-Means Clustering.
- Finding appropriate K value using elbow technique for a particular problem.
- Implementation of K-Means clustering for IRIS datasets

Projects

- To be defined based on respective study area of student.

References:

Text Book:

1. EthemAlpaydin, Introduction to Machine Learning, Second Edition, http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=12012.

Web Resource:

1. https://towardsdatascience.com/beginners-guide-to-machine-learning-with-python-b9ff35bc9c51

Embedded System Programming with ARM-Cortex

Code	Course Title	Credit	T-P-PJ
CUTM1039	Embedded System Programming with ARM-Cortex	6	3-2-1

Objective

- To allow students in Embedded System sectors to learn programming / Interfacing peripherals to ARM Cortex based Microcontroller

Learning outcome

- Describe the architectural features and instructions of 32 bit ARM Cortex M3 microcontroller.
- Understand the basic hardware components and their selection method based on the characteristics and attributes of an Embedded System.
- Understand various Sensors, Actuators & Interfacing Modules.

Course content

Module I: EMBEDDED C **(4 Hrs)**

Embedded System, Programming Embedded system, Factor for selecting the Programing language, Embedded C programming Language, Embedded C vs C.

Practice:

1. Familiarization with tools (STM32CubeMX, KeiluVision IDE, Flash Magic & Proteus Simulator).
2. Programming STM32 using KeiluVision& STM32CubeMX.

Module II: ARM-32 bit MICROCONTROLLER **(6 Hrs)**

ARM Design Philosophy & RISC Architecture, Programmer's Model. ARM Cortex M, Cortex M Architecture, ARM Cortex-M Internals & Debugging.

Practice:

1. Familiarization with Different Processors and Controllers Boards (8, 16, 32, 64 bits)

Module III: STM32 GPIO MANAGEMENT **(14 Hrs)**

GPIO Configuration, Driving De-initialization, Interfacing IO devices and its type – LEDs, Switches, Buzzer, Seven Segment Display, LCD (4 bit, 8 bit Mode), Keypad (4*4), DC Motor, Stepper Motor, Servo motor, Relay.

Practice:

1. Write an Embedded C program to interface LEDs with STM32.
2. Write an Embedded C program to interface Switch with STM32.
3. Write an Embedded C Program to design up counter & down counter using Seven Segment Display. (1 digit, 2 digit)
4. Write an Embedded C program to interface buzzer to control with the help of Switch.

5. Write an Embedded C program to display characters on Alphanumeric LCD.
6. Write an Embedded C program to interface Keypad and LCD with STM32.
7. Write an Embedded C program to interface DC Motors, Stepper Motor, and Servo Motor rotate clockwise, anticlockwise and in angle (45°, 90°, 180°).
8. Write an Embedded C program to interface relay to control the AC Appliances.

Module IV: STM32 INTERRUPT MANAGEMENT & UART (14 Hrs)

NVIC Controller, Enabling Interrupt, Interrupt Priority Levels, UART Initialization, UART communication in polling Mode & in Interrupt Mode. Wireless Technologies- Bluetooth, Wi-Fi, RF.

Practice:

1. Write an embedded C program to generate an Interrupt process using STM32.
2. Write an Embedded C program to interface STM32 to Bluetooth Module to send & receive Data.
3. Write an Embedded C program to interface STM32 to GPS module to get a Location Coordinate.
4. Write an Embedded C program to interface STM32 to GSM module to Send & Receive SMS.
5. Write an Embedded C program RF module with STM32 to send and receive the data wirelessly.
6. Write and Embedded C program to design a system to read the RFID cards using STM32.
7. Write and Embedded C program to connect ESP8266 with STM32 to create a Webserver.

Module V: STM32 TIMERS , ADC, & DAC (10 Hrs)

Timers Basics, General Purpose Timer, SysTick Timer, ADC & DAC Basics, Initialization, DAC Peripherals & Modules. Analog Sensors and its Types(Ultrasonic Sensor, Temperature, Humidity, Soil Moisture Sensor, PIR sensor)

Practice:

1. Write an Embedded C Programs to generate Delay using Timer.
2. Write an Embedded C program to display output for given analog input using internal ADC. (Use of Analog Sensors like Ultrasonic Sensor, Temperature, Humidity, Soil Moisture Sensor, PIR sensor)
3. Write an embedded C program to generate Triangular and Square waves using DAC.

Module VI: STM32 I2C & SPI (10 Hrs)

I2C specification, Protocol configuration, I2C Peripherals. SPI Specification, Protocol configuration, it's Peripheral and Modules.

Practice:

1. Write an Embedded C program to build I2C communication between STM32 and Arduino
2. Write an Embedded C program to build SPI communication STM32 to the Arduino board.

Module VII: PWM & CAN (8 Hrs)

RTC feature and its Module, CAN Protocols Overview, Application, Architecture, Data Transmission & Data Frames.

Practice:

1. Write an Embedded C program to implement a Real-Time Clock.
2. Write an Embedded C program to Speed Control of DC motor using PWM.
3. Write an Embedded C program to change the intensity of Light using PWM.

Text Books:

1. Shibu K V, ―Introduction to Embedded Systems‖, Tata McGraw Hill Education Private Limited, 2nd Edition
2. Noviello, Carmine. "Mastering STM32." Obtenido de http://www2. keil. com/mdk5/uvision,2017.
3. Norris, Donald. Programming with STM32: Getting Started with the Nucleo Board and C/C++. McGraw Hill Professional, 2018.

Reference Books:

1. STM32F10xx User Manual
2. https://www.udemy.com/course/stm32cubemx-complete-training/learn/lecture/9606338#overview

1.**https://www.udemy.com/course/embedded-c-programming-for-embedded-systems/**

Digital Communication Systems

Code	Course Title	Credit	T-P-PJ
CUTM1044	Digital Communication Systems	3	2-1-0

Objective

- To impart the fundamentals of modern digital communication system design.
- To evaluate the performance of digital signalling schemes on realistic communication channels.
- Know the techniques of digital communication, information theory, and error control coding.

Learning Outcome

- Analyze the performance of a baseband and pass band digital communication system in terms of error rate and spectral efficiency.
- Perform the time and frequency domain analysis of the signals in a digital communication system.
- Develop understanding about performance of digital communication systems.
- Calculate bandwidth and power requirements for digital systems.

Course Outline

ModuleI(05 Hours) **[03 hrs.Theory + 02hrs. Practice]**

Digital Representation of Analog Signal:

Sampling Theorem, Signal Reconstruction from uniform samples,Quantization of Signals, Quantization error, PCM, Electrical representation of binary digits, PCM System, Companding.

Practice: (*using Trainer Kit/MATLAB)*

1. Analysis of Sampling theorem by using Trainer Kit/MATLAB.
2. PCM Modulation and Demodulation using Trainer Kit/ MATLAB.

ModuleII(5 Hours) **[03 hrs. Theory + 02hrs. Practice]**

Issues in Digital transmission: Line coding, Scrambling, T1Digital System, Multiplexing T1 lines – The T2, T3, T4, E1 lines, Differential PCM: Linear predicted design, Delta Modulation, Adaptive Delta Modulation
(Ref Text Book 1: Chapter 5.1.1, 5.4, 5.5 and 5.6)

Practice: (*usingTrainer Kit/MATLAB)*

3. Differential PCM Modulation and Demodulation using Trainer Kit/ MATLAB
4. Delta Modulation and Demodulation using Trainer Kit/ MATLAB

ModuleIII (6 Hours) [03 hrs. Theory + 03 hrs. Practice]

Digital Modulation Technique:

Generation, Transmission, Reception, Spectrum and Geometrical Representation in the Signal Space of BPSK,QPSK, QASK, M-ary PSK, BFSK, M-ary FSK, and Minimum Shifting Keying (MSK), GMSK, 16-QAM, 64-QAM.
(Ref Text Book 1: Chapter 6)
Practice :(*using MATLAB)*

5. PSK Modulation and Demodulation using Trainer Kit/MATLAB.
6. FSK Modulation and Demodulation using Trainer Kit/MATLAB.
7. ASK Modulation and Demodulation using Trainer Kit/MATLAB.

Module IV (5 Hours) [03 hrs. Theory + 02 hrs. Practice]

Noise in PCM and DM: Calculation of Quantization Noise, Output Signal Power, and the Thermal Noise, Output SNR in PCM, Quantization noise in Delta Modulation, output signal power, output SNR, Comparison with PCM and DM
(Ref Text Book 1: Chapter 12.1 -12.3)
Practice: (*using MATLAB)*

8. SNR study of PCM using MATLAB
9. SNR study of Delta Modulation using MATLAB

Module V (3 Hours) [03 hrs. Theory]

Principle of Digital Data Transmission:

Digital Communication Systems – Source, Line coder, Multiplexer, Regenerative repeater; Line Coding: PSD of various line codes, polar signaling, constructing a DC Null in PSD by pulse shaping, On Off signaling, Bipolar signaling; Pulse shaping – ISI and effect, Nyquist first criterion for zero ISI; Scrambling, Digital receiver and regenerative repeaters; Equalizers, Timing extraction, Detection error, Eye Diagram

(Ref Text Book 2: Chapter 7.1, 7.2, 7.3.1, 7.3.2, 7.4, 7.5, 7.6)

Module VI (4.5 Hours) [03hrs.Theory + 1.5 hrs. Practice]

A base band signal Receiver, Peak signal to RMS noise output voltage ratio, probability of error, optimum threshold, optimum receiver for both base band and pass band: calculation of optimum filter transfer function, optimum filter realization using Matched filter.
(Ref Text Book 1: Chapter 11.1 – 11.3)

Practice:(*using MATLAB)*

10. Study of optimum filters realization using MATLAB.

Module VII (04 Hours) [02 hrs.Theory + 02 hrs. Practice]

Discrete Messages and information content:

Information theory: entropy, mutual information and channel capacity theorem. Shannon coding, Hoffman coding. Fundamentals of error correction, hamming codes.

(Ref Text Book 1: Chapter 13.1 – 13.3)

Practice:(*using MATLAB)*

11. Study of Hamming codesusing MATLAB.
12. Study of error correction using MATLAB.

1. **Reference**

- **Text Books:**

1. Principles of Communication Systems by H Taub, D. L. Schilling and G Saha, 3rdEdition 2008, TMH Education Pvt. Ltd, New Delhi,

- **Reference Books:**

1. Modern Digital and Analogue Communication Systems by B.P. Lathi and Z Ding, 4th Edition 2010, Oxford University Press, New Delhi
2. Communication Systems by SimonHaykin, 4thEdition, John Wiley & Sons, Inc.
3. Digital Communications, Proakis and Salehi, 5th Edition, Pearson Edu.

- **Online Source:**

1. http://nptel.ac.in/courses/117101051/Digital Communication by Prof.Bikash Kumar Dey, IIT,Bombay, Video Course.
2. https://www.udemy.com/course/digital-communication-information-theory/
3. https://www.udemy.com/course/digital-analog-introduction-to-modulation-in-communication-systems/

Network Analysis

Code	Course Title	Credit	T-P-PJ
CUTM1043	Network Analysis	3	2-1-0

Objective

- To learn techniques of solving circuits involving different active and passive elements.
- To analyze the behavior of the circuit's response in time domain.
- To analyze the behavior of the circuit's response in frequency domain.
- To synthesize an electrical network from a given impedance/admittance function.

Learning outcome

- Apply the knowledge of basic circuital law and simplify the network using reduction techniques.
- Analyze the circuit using Kirchhoff's law and Network simplification theorems.

- Infer and evaluate transient response, Steady state response, network functions.
- Obtain the maximum power transfer to the load , and Analyze the series resonant and parallel resonant circuit.
- Evaluate two-port network parameters.

Course content

Module I: Basic Circuit Concepts and Basic Laws (3hrs Theory)

Circuit concepts –Resistor(R)-Inductor(L)-Capacitor(C)-Voltage and Current Sources - Source transformation-Voltage, Current relationship for passive bilateral elements -Ohm's law, Kirchhoff's laws.

Module II: Methods of Analysis (3hrs Theory)

Nodal analysis, mesh analysis, super node and super mesh, Nodal Versus Mesh Analysis for D.C excitations.

Module III: Circuit Theorems (3hrs Theory + 4hr Practice)

Superposition theorem- Thevenin's theorem- Norton's theorem- Maximum power transfer theorem- Reciprocity theorem.

Practice:

1. Verification of Superposition Theorem
2. Verification of Thevenin's Theorem
3. Verification of Maximum Power Transfer Theorem
4. Verification of Reciprocity Theorem

Module IV: Network Topology and Two Port Networks (3hrs Theory + 4hrs Practice)

Network topology, Incidence matrix, Tie-set matrix, Cut-set matrix, Dual networks- Two port network, Impedance Parameter, Admittance Parameter, Transmission line.

Practice:

1. Determination of Z parameters of Port Network
2. Determination of Y parameters of Port Network
3. Determination of h parametersof Port Network
4. Determination of ABCD parametersof Port Network

Module V: Time Response of Circuits (DC Excitation) (3hrs Theory)

Time (Transient) response of R-L, R-C, R-L-C series circuits for Zero input, Step input, pulse input -Initial conditions-solution method using differential equation and Laplace transforms.

Module VI: Single Phase A.C Circuits (3hrs Theory + 3hrs Practice)

Sinusoidal alternating quantities – Phase and Phase difference – Complex and polar forms of representations, J-notation, R.M.S, Average values and form factor for different periodic wave forms - Steady state analysis of R,L and C (in series, parallel and series parallel combinations) with sinusoidal excitation-Concept of Reactance, Impedance Susceptance and Admittance-Power Factor and significance-Real and Reactive power, Complex Power.

Practice:

1. Study of Step Response of R-L Network
2. Study of Step Response of R-L Network
3. Study of Time Response of R-L-C Network.

Module VII: S Domain Analysis (3hrs Theory + 1hr Practice)

Transform Impedance and Transform Circuits, Series and Parallel Combination of Elements, Terminal Pairs, Network Function for the One port and Two port, Poles and Zeros of Network Functions.

Practice:

1. Frequency response of a series and parallel resonant circuit by laboratory set up

Text Books:

1. M. E. VAN VALKENBURG, "Network Analysis", PHI Publications.
2. A Sudhakar and Shyammhoan S Palli, "Network Analysis", MC Graw Hill Publishers.

Reference Books

1. Smarajit Ghosh, "Network Theory Analysis & Synthesis", MC Graw Hill Publishers.
2. B.R.GUPTA, "Network Analysis & Synthesis", S.Chand.
3. BH ferri, "Linear Circuits 2: AC Analysis", Coursera
4. B Tapas Kumar, "Linear Analysis", NPTEL.

Analog Communication Systems

Code	Course Title	Credit	T-P-PJ
CUTM1045	Analog Communication Systems	3	2-1-0

Objective

- Impart the basic concepts of analog modulation schemes.
- Describe different types of noise and predict its effect on various analog communication systems.
- Know the techniques of analog communication and noise analysis in analog communication.

Learning Outcome

- Analyze energy and power spectral density of the signal.
- Develop an understanding of the performance of analog communication systems.
- Calculate bandwidth and power requirements for analog systems.
- Analyze the different characteristics of the receiver.

Course Outline

Module I (5 Hours) [3 hrs.Theory + 2 hrs. Practice]

Basic block diagram of analog communication. Need for modulation, Fourier transform, Properties of Fourier transform: Duality property, Frequency shifting property, Modulation property. Introduction to AM: Time-Domain description, Frequency – Domain description, Generation of AM wave: square law modulator, switching modulator. Detection of AM waves: square law detector, envelop detector.

Practice: (*using Hardware/MATLAB)*

1. DSB+C Modulation using Trainer Kit/ MATLAB
2. DSB+C Demodulation using Trainer Kit/ MATLAB

ModuleII (6 Hours) **[3 hrs.Theory + 3hrs. Practice]**

Double side band suppressed carrier modulation (DSBSC): Time-Domain description, Frequency-Domain representation, Generation of DSBSC waves: balanced modulator, ring modulator. Coherent detection of DSBSC modulated waves.

SINGLE SIDE-BAND MODULATION (SSB):

Quadrature carrier multiplexing, SSB modulation, Frequency-Domain description of SSB wave, Time-Domain description. Phase discrimination method for generating an SSB modulated wave, Demodulation of SSB waves.

Practice: (*using Hardware/MATLAB)*

3. Quadrature Carrier Multiplexing using MATLAB/SCILAB.
4. DSB-SC Modulation and Demodulation using Trainer Kit/ MATLAB.

VESTIGIAL SIDE-BAND MODULATION (VSB):

Frequency – Domain description, Generation of VSB modulated wave, Time – Domain description, Comparison of amplitude modulation techniques, Frequency translation, Frequency division multiplexing, Superheterodyne receiver.

Practice: (*using Hardware/MATLAB).*

5. Frequency Division Multiplexing using Trainer Kit/MATLAB.

Module III (5 Hours) **[3 hrs.Theory + 2hrs. Practice]**

ANGLE MODULATION (FM):

Basic definitions, FM, narrow band FM, wide band FM, transmission bandwidth of FM waves, Generation of FM waves: indirect FM and direct FM.

Practice: (*using Hardware/MATLAB)*

6. FM Modulation using Trainer Kit/MATLAB/SCILAB

Module IV (5 Hours) **[3 hrs. Theory + 2 hrs. Practice]**

DEMODULATION OF FREQUENCY MODULATED SIGNALS:

Demodulation of FM waves, FM stereo multiplexing, Phase-locked loop, Non-linear model of the phase – locked loop, Linear model of the phase-locked loop, Nonlinear effects in FM systems.

Practice: (*using Hardware/MATLAB)*

7. FM Demodulation using Trainer Kit/MATLAB/SCILAB.
8. FM Demodulation using PLL.

Module V (4 Hours) **[2hrs.Theory + 2hrs. Practice]**

RANDOM PROCESS:

Random variables,Statistical averages: Function of random variables, moments, mean, Correlation and Covariance function: Principles of autocorrelation function, cross-correlation functions, Central limit theorem, Properties of Gaussian process.

Practice: (*using Hardware/MATLAB)*

9. Study of Autocorrelation&Cross-correlation using MATLAB

ModuleVI(5 Hours) [3 hrs. Theory + 2 hrs. Practice]

NOISE: Introduction, shot noise, thermal noise, white noise, Noise equivalent bandwidth, Narrow bandwidth, Noise Figure, Equivalent noise temperature, cascade connection of two-port networks, Frequency Domain Representation of Noise, Power Spectral Density, Spectral Components of Noise, Response of a Narrow band filter to noise, Effect of a Filter on the Power spectral density of noise, Superposition of Noises, Noise Bandwidth, Narrow band representation of noise and its PSD.

Practice: (*using Hardware/MATLAB/LABVIEW)*

10. Generation of Gaussian Noise using MATLAB/SCILAB

ModuleVII(3 Hours) [3 hrs. Theory]

NOISE IN CONTINUOUS WAVE MODULATION SYSTEMS:

Introduction, Receiver model, Noise in DSB-SC receivers, Noise in SSB receivers, Noise in AM receivers, Threshold effect, Noise in FM receivers, FM threshold effect, Pre-emphasis and De-emphasis in FM.

- **Text Books:**

1. Communication Systems, Simon Haykins, 5th Edition, John Willey, India Pvt. Ltd, 2009.
2. H. Taub, D. L Schilling, G. Saha,Principles of Communication System, 3rd Edition, 2008, Tata McGraw Hill, India; ISBN: 0070648115. (Selected portions from chapters: 1, 2, 3, 4, 5, 7, 8 and 9)

- **Reference Books:**

1. Modern digital and analog Communication systems B. P. Lathi, Oxford University Press., 4th ed, 2010,
2. Communication System Engineering, Second Edition by MasoudSalehi, John G. Proakis, ISBN: 0130950076
3. Principles of Electronic communication Systems,Louis E. Frenzel,3rd Edition, Tata McGraw Hill.
4. Communication Systems: Analog and digital, Singh and Sapre,TMH, 2nd Ed, 2007.

Basket IV: Core Courses Syllabus

OOPs with C++ Programming

Code	Course Title	Credit	T-P-PJ
CUTM1028	OOPs with C++ Programming	4	1-2-1

Objective

- To understand how C++ improves C with object-oriented features
- To learn how to design C++ classes for code reuse
- To learn how inheritance and virtual functions implement dynamic binding with polymorphism
- To learn how to use exception handling in C++ programs

Learning outcome

- Apply the object-oriented programming approach in connection with C++
- Illustrate the process of data file manipulations using C++
- Apply virtual and pure virtual function & complex programming situations
- Write an error free program of minimum 200 lines of code

Course content

Module I: Revision of C programming (8 hrs)

Revision of C Programming, Pointers, Functions (Call by value and reference), Recursion, Arrays using Pointers, Structures, Union, Enumeration and Typedef, File handling.

Programs:

1. Write a Program to perform Parameter passing.
2. Write a program to create a scientific calculator.
3. Write a program to convert a decimal to binary number using recursion.
4. Write a program to Read 'n' employee details and display the top 10 employees as per the salary.
5. Write a program to evaluate MCQ questions of an examination and generate the results using files.

Module II: Basics of Object oriented concepts (8 hrs)

Object oriented concepts Classes and Objects, Encapsulation, Abstraction, Overloading, Inheritance, Polymorphism.

Beginning with C++, Tokens, Static Members, Constant Members, Expressions, Control Structure, Functions: parameter passing, inline function, function overloading.

Programs:

1.Write a program to read a number and check whether the number is Prime number , Palindrome number , Magic number , Armstrong number , Strong number or not.

2. Write definitions for two versions of an overloaded function. This function's 1st version sum() takes an argument, int array, and returns the sum of all the elements of the passed array. The 2nd version of sum() takes two arguments, an int array and a character ('E' or 'O'). If the passed character is 'E', it returns the sum of even elements of the passed array and is the passed character is 'O', it returns the sum of odd elements. In case of any other character, it returns 0 (zero).

Module III: Class-Object-Constructor (10 hrs)

Classes: data members, member function, array of objects, static data members, constant members function, and friend function.

Constructors, Encapsulating into an object, Destructors.

Programs:

1. Define a class to represent a book in a library. Include the following members:

Data Members

Book Number, Book Name, Author, Publisher, Price, No. of copies issued, No. of copies

Member Functions

(i) To assign initial values

(ii) To issue a book after checking for its availability

(iii) To return a book

(iv) To display book information.

2. A bank maintains two kinds of accounts for customers, one called as savings and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level a service charge is imposed.

Define a class to represent a bank account. Include the following members: Data members: 1. Name of the depositor. 2. Account number. 3. Type of account. 4. Balance amount in the account. Member functions: 1. To assign initial values. 2. To deposit an amount. 3. To withdraw an amount after checking the balance. 4. To display the name and balance. Write a main program to test the program

3. Declare a class to represent fixed-deposit account of 10 customers with the following data members:

Name of the depositor, Account Number, Time Period (1 or 3 or 5 years), Amount.

The class also contains following member functions:

(a) To initialize data members.

(b) For withdrawal of money (after alf of the time period has passed).

(c) To display the data members.

4. Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimeters depending on the object on display.

Module IV: Inheritance **(8 hrs)**

Associations, Inner Classes, Memory Management and pointers

Inheritance: Derived classes, member accessibility, forms of inheritance, virtual base classes.

Programs:

1. Write a Program to describe about all types of inheritance.

2. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area.

3. An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in following figure.

The figure also shows the minimum information required for each class. Specify all classes and define functions to create the database and retrieve individual information as and when required.

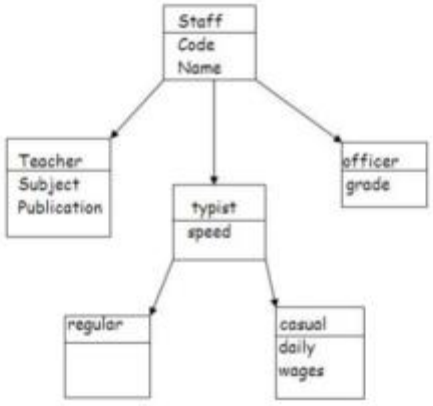

fig: class relationships (for exercise 8.3)

Module V: Polymorphism (8 hrs)

Polymorphism (Compile time Polymorphism, Run time Polymorphism), Virtual Functions, Abstract class, virtual destructors, Interfaces.

Programs:

1. Write a Program to overload ++ operator.
2. Write a program to overload + operator by concatenating strings.
3. Write a program to describe about virtual function.

Module VI: Exception Handling **(8 hrs)**

Exception Handling, Managing Console I/O Operations, Streams & Files: streams, hierarchy of stream classes, working with files

Programs:

1. Write a Program to describe about exception handling mechanism.
2. Write a Program to describe multi catch statement.
3. Write a program to read a list containing item name, item code, and cost interactively and produce a three column output as shown below.

Name	Code	Cost
Turbo C++	1001	250.95
C primer	905	95.70
.....	.....	.....
.....	.....	.....

Note that the name and code are left-justified and the cost is right justified with a precision of two digits. Trailing zeros are shown.

4. Write a program that reads a text file and creates another file that is identical except that every sequence of consecutive blank spaces is replaced by a single space.

5. Write a program that reads character from the keyboard one by one. All lower case characters get store inside the file LOWER, all upper case characters get stored inside the file UPPER and all other characters get stored inside OTHERS.

Module VII: Templates **(8 hrs)**

Advance Topics in C++ Object Design and Templates STL (Standard Type Libraries)RTTI (Run Time Type Identification) Advanced Typecasting ,new data types, new operators, class implementation, namespace scope , operator keywords, new headers , C++ Containers

Programs:

1. Write a function template for finding the minimum value contained in an array.
2. Imagine a publishing company that markets both books and audio-cassette versions of its works. Create a class called Publication that stores the title (a string) and price of a publication. From this class derive two classes: Book, which adds a page count (type int); and Tape, which adds a playing time in minutes (type float). Each of the three class should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display the data. Write a main() program that creates an array of pointers to Publication. In a loop, ask the user for data about a particular book or Tape, and use new to create a object of type Book or Tape to hold the data. Put the pointer to the object in the data for all books and tapes, display the resulting data for all the books and taps entered, using a for loop and a single statement such as pubarr[i]->putdata();to display the data from each object in the array.

Text Books:

1. E Balagurusamy, "Object Oriented Programming with C++", Tata McGraw Hill, Sixth Edition.
2. Herbert Schlitz, "The Compete Reference C++", Tata McGraw Hill, Fourth Edition.

Reference Books:

1. Ashok Kamthane, "Object Oriented Programming with ANSI and Turbo C++", Pearson.
2. Behrouz A. Forouzan & Richard F. Gilberg "A Structured approach using C++" Cengage Learning Indian Edition.

Data Structures using C++

Code	Course Title	Credit	T-P-PJ
CUTM1029	Data Structures using C++	3	1-2-1

Objective

- Be familiar with techniques of algorithm analysis and Recursive method
- Be familiar with implementation of linked data structures such as linked lists and binary trees
- Be familiar with several sub-quadratic sorting algorithms including quick sort, merge sort and heap sort
- Be familiar with some graph algorithms such as shortest path and minimum spanning tree

Learning outcome

- Evaluate algorithms and data structures in terms of time and memory complexity of basic operations
- Define basic static and dynamic data structures and relevant standard algorithms for them: stack, queue, dynamically linked lists, trees, graphs, heap, priority queue, hash tables, sorting algorithms, min-max algorithm
- Determine and demonstrate bugs in program, recognize needed basic operations with data structures
- Formulate new solutions for programming problems or improve existing code using learned algorithms and data structures

Course content

Module I: Problem Solving Analysis (6 hrs)

Define the problem, Identify the problem, Introduction to Problem Solving, Problem solving basics, Defining creativity v/s innovation

Find Creative Solutions using creativity tools

Effective problem solving approaches , Critical thinking and information analysis , Brainstorming, Reverse Brainstorming, Imagineering, Mind Mapping, Six Thinking Hats: A Tool to Strengthen Critical Thinking, Collaboration, Communication, and Creativity Skills , Analyzing the situation, Gathering information, Identifying solution criteria , Decision Making Methods , Charts and Diagrams , Applying outcome-based thinking

Evaluate and Select solution

Pro's and Con's, Force field analysis, Feasibility/Capability Analysis, Decision analysis, evaluating problems , Choosing among alternatives , Qualitative analysis, discussing qualitative analysis techniques , Establishing objectives , Assigning weight to objectives in order to make the best decision , Creating a satisfaction scale to choose between alternatives

Implementing Decisions

Create an action plan, Break solution into action steps, Prioritize actions and assign roles (setting priorities for taking action) ,Follow-up at milestones

Programs:

1. Problem solving (Control structures, Arrays) using Raptor Tool.

Module II: Array & Stack (9 hrs)

Analysis of different Algorithms, Asymptotic analysis, Algorithm analysis, Complexity Analysis, Application of Data structures

Basic Data Structures, Arrays, Stacks and its applications (Recursion, Infix to Postfix Conversion and Postfix Evolution

Programs:

1. Write a program to perform the following menu driven program on the input array.

a. Insertion
b. Deletion
c. Searching
d. Sorting
e. Merging
f. Display
g. Exit

2. Write a program to perform the following menu driven program on the STACK.

a. Push
b. Pop
c. Display
d. Exit

Module III: Queue & Linked List **(9 hrs)**

Queues, Priority Queues, Dequees.

Linked lists: Single Linked List and Operations on Single Linked List (Creation Insertion , Deletion , Sorting and Reverse).

Programs:

1. Write a program to perform the following menu driven program on the Queue.
    a. Insertion
b. Deletion
c. Display
d. Exit

2. Write a program to create a single linked list performs the following menu driven program.
a. Insertion at front
b. Insertion at end
c. Insertion at particular position
d. Deletion at front
e. Deletion at end
f. Deletion at particular position
g. Display

Module IV: Stack & Queue Using Linked List **(8 hrs)**

Circular linked list and Double linked list, Stack implementation using Linked List and Queue implementation using Linked List

Programs:

1. Write a program to create a Double linked list performs the following menu driven program.
a. Insertion at front
b. Insertion at end
c. Insertion at particular position
d. Deletion at front
e. Deletion at end
f. Deletion at particular position
g. Display

2. Write a program to create a circular linked list and display it.
3. Write a program to implement Stack Using Linked List.
4. Write a program to implement Queue Using Linked List.

Module V: Trees **(10 hrs)**

Trees and hierarchical orders ,Introduction to trees , Abstract trees , Tree traversals , Forests , Ordered trees , Binary trees , Perfect binary trees , Complete binary trees , Search trees , Binary search trees , AVL trees

Programs:

1. Write a program to create Binary tree and display it.
2. Write a program to create a BST and display it.
3. Write a program to print all pairs from two BSTs whose sum is greater than the given value.

4. Write a program to remove duplicate entries from the BST.
5. Write a program to create a AVL tree and display it.

Module VI: Searching & Sorting **(8 hrs)**

Searching & Sorting algorithms , Objectives of Searching , The Sequential Search , Analysis of Sequential Search , The Binary Search , Analysis of Binary Search , Introduction to sorting , Insertion sort , Bubble sort , Heap sort ,Merge sort ,Quick sort

Programs:

1. Write a program to perform linear and binary search.
2. Write a program to perform selection sort, Bubble sort and Insertion sort.
3. Write a program to perform merge and quick sort.
4. Write a program to perform Heap sort.

Module VII: Hashing **(8 hrs)**

Hash functions and hash tables ,Hashing & Introduction to hash tables ,Hash functions , Mapping down to 0 ... M – 1 , Chained hash tables , Scatter tables , Open addressing , Linear probing , Quadratic probing , Double hashing, Poisson distribution , Collision Resolution
Graph Terminology and Traversals.

Programs:

1. Write a program to perform Linear Probing.
2. Write a program to perform Double Hashing

Text Books:

1. Data Structures, Algorithms and Applications in C++, Sartaj Sahani, 2nd Edition.
2. Data Structures and Algorithms in C++, Michael T.Goodrich, R, Tamassia and D.Mount, wiley Student Edition, 7th edition, John Wiley and Sons.

Reference Books:

1. Data Structures and Algorithms Analysis in C++ by Mark Allen Weiss.
2. Data Structures and Algorithms in C++, 3rd edition, Adam Drozdek, Cengage Learning.

Source of reference;http://courseware.cutm.ac.in/courses/data-structures-using-c/

System Integration with DYMOLA

Code	Course Title	Credit	T-P-PJ
CUTM1022	System Integration with DYMOLA	2	0-0-2

Course Objectives

- To provide powerful multi-disciplinary systems engineering through compatible model libraries for a large number of engineering domains.
- To design high-fidelity modeling of complex integrated systems.
- To design intuitive modeling i.e. advanced, formally defined object-oriented modeling language.
- To enable users to easily build their own components or adapt existing ones to match their unique needs.
- To provide hardware-in-the-loop simulations (HILS) i.e. real-time simulation with AurdinoUno, Python, Matlab, 3D real-time animation, CAD files import capability.

- To increase the ability to integrate with complex 3D geometry for integrated simulation.
- To increase powerful model management, calibration & optimization capabilities.

Learning Outcomes

- The use of open standards such as DYMOLA (Modelica and FMI) is a key enabler to better understand the behavior of systems and to work and communicate accurately with partners and suppliers.
- DYMOLA is not only capable to support an ad-hoc modeling level, such as functional behavior or detailed design, but is also able to convert these predictive models into real-time models.
- The user can able to create new elements in an easy and intuitive way, to answer to its own modeling requirements.
- Future Centurions are ready for operating in many industries including automotive, aerospace, architecture, Motorsport, energy, and high tech.

Course Syllabus

Module 1 - Introduction Dymola and Modelica library

Package Browser, Component Browser, Parameter and Variable Editor Simulation Window, Modeling, and Simulation.
The Modeling window is used to compose models and model components.
The Simulation experiment on the model, plot results and animate the behavior.
Creating user-defined models and scripting using Modelica language.

Role Play – Explore the pre-defined libraries and Models, Creating a Package
Practice Project - Preparation of animated projects

https://www.youtube.com/watch?v=39xyI0k
https://www.youtube.com/watch?v=FN8LlnTwzVE&t=314s

Module 2 – Physical Modeling using DYMOLA

Import of user-defined libraries and packages, interfacing with physical models using ArduinoUno.
The Simulation experiment on the model using multi-domain libraries such as mechanical, electrical, control, thermal, pneumatic, hydraulic, powertrain, thermodynamics, vehicle dynamics, air-conditioning domains
Dymola interface that is stored in the Python package
Role Play – Explore the pre-defined libraries and Models, Creating a Package
Practice Project - Preparation of projects using user-defined packages,

Systems Physics with Modelica/Dymola

https://www.youtube.com/watch?v=xlpHwX-W3Ns

Module 3 – Animation and 3D view Using DYMOLA

MultiBody Frame Connector, Building a Mechanical Model, Concept of Furuta
Role Play - Practical session by students for students
Practice Project - Modeling of animated projects using the MultiBody library.

https://www.youtube.com/watch?v=c9Ar2b4X5rQ

https://www.youtube.com/watch?v=k7lLBASaEJg

Session Plan

Session 1

Project 1

Simulating a model – Modeling of Integrated circuits

Description: Use of Electrical and Electronics components.

Workbench Use: Behavior Modelling, Functional and Logical Design.

Session 2

Project 2

Simulating a model -Creating a model for Electric DC Motor

Description: Design a DC Motor Model, Test, and Simulation, Creating a library for components, Creating a model for motor drive, Scripting.

Workbench Use: Behavior Modelling, Modelica Standard Library.

Session 3

Project 3

Simulating a model -Simple Pendulum with Frictionless joint Using Multi-Body Library

Description: Design the Simple pendulum and the Furuta joint using Dymola and Modelica language. Friction joint for the Mechanical equipment.

Workbench Use: Behavior Modelling.

Session 4

Project 4

Simulating a model – Pick and Place Robot

Description: 5 Axis Pick and Place Robot Design, Validation, and Optimization in the 3DS platform.

Workbench Use: Behavior Modelling, Functional and Logical Design. Part design and Assembly Design.

https://www.youtube.com/watch?v=9RgdZUvEjPw

Session 5

Project 5

Simulating a model – 3D Printer Design

Description: Design All System and Sub System of the 3D Printer, Validation and Simulation using 3Ds Platform.

Workbench Use: Behavior Modelling, Functional and Logical Design. Part design and Assembly Design.

Session 6

Project 6

Simulating a model – Bicycle Behavior Modeling

Description: Design Power Train, Driving Cycle, part design, and Simulation.
Workbench Use: Behavior Modelling, Functional and Logical Design. Part design and Assembly Design.

Session 7

Project 7

Simulating a model – Refrigerator Compartment Door Design using Thermal Library

Description: This component model the airflow through the door of a refrigerator or freezer compartment.
Workbench Use: Behavior Modelling, Functional and Logical Design. Part design and Assembly Design.

Session 8

Project 8

Simulating a Model – Engine Analytic Using MultBody Library.

Description: Engine analytic, an engine with 6 cylinders, 6 planar loops, 1 degree of freedom, and analytic handling of kinematic loops.
Workbench Use: Behavior Modeling.

Session 9

Project 9

Simulating a model – Control the real and Digital servo motor ArduinoUno Library

Description: Control the Real and Digital Servo motor with simulation.
Workbench Use: Behavior Modelling, Arduino based System Design, and Functional and logical design.

Session 10

Project 10

Simulating a model – Virtual Universes with Poppy Humanoid Using ArduinoUno Library

Description: Virtual universes with a human assistant robot with simulation.
Workbench Use: Behavior Modelling, Arduino based System Design, Functional, and logical design.

Session 11

Project 11

Simulating a model – Implementation of Model using Python Library

Description: Modeling using python library, validation and optimization in the 3Ds platform.
Workbench Use: Behavior Modelling, Functional and Logical Design. Part design and Assembly Design.

Session 12

Project 12

Simulating a model – Industrial Robot Design

Description: 6 Axis industrial robot design, validation, and optimization in the 3Ds platform.
Workbench Use: Behavior Modelling, Functional and Logical Design. Part design and Assembly Design.

Session 13

Project 13

Simulating a model – Temperature Control System Using State Graph

Description: The model contains an electric circuit with a heating resistor and a switch.
Workbench Use: Behavior Modelling.
https://www.youtube.com/watch?v=zz-_crJOGo0&t=26s
https://www.youtube.com/watch?v=Zl592ARjnpU

Session 14

Project 14

Simulating a model – Magnetic Ball System using Magnetic Library

Description: The electronic circuit consists of a voltage source, a resistor, and an inductor in the form of a tightly wound coil. An iron ball beneath the inductor experiences a gravitational force as well as an induced magnetic force (from the inductor) that opposes the gravitational force.
Workbench Use: Behavior Modelling.

Session 15

Project 15

Simulating a Model – Design of Water to Steam Converter Using Fluid Library

Description: Create a package under Fluid_Package called Water_To_Steam using temperature sensors.
Workbench Use: Behavior Modelling.

Session 16

Project 16

Simulating a Model – Design of Liquid Valve Control Using Fluid Library

Description: Building a simple circuit with two valves and a volume block.
Workbench Use: Behavior Modelling.
https://www.youtube.com/watch?v=P_YI3RiTI14

Antennas Analysis & Design

Code	Course Title	Credit	T-P-PJ
CUTM1049	tennas Analysis & Design	3	2-1-0

Objective

- To understand the theory and fundamentals of antenna design.
- This course helps the students to learn key aspects of practical antenna design.
- A broad range of antennas such as dipole, loop, microstrip patch, horn, smart etc are studied during the course

Learning Outcome

- Design and analyze antenna arrays
- Design and analyze wire and aperture antennas
- Identify the characteristics of radio-wave propagation

Course content

Module I: Fundamental Concepts: 8 Hours

Physical concept of radiation, Radiation pattern, near- and far-field regions, reciprocity, directivity and gain, effective aperture, polarization, input impedance, efficiency, Friis transmission equation, radiation integrals and auxiliary potential functions.

Module II: Radiation from Wires and Loops: (7 Hours)

Infinitesimal dipole, finite-length dipole, linear elements near conductors, dipoles for mobile communication, small circular loop. Polarization – Linear, Circular and Elliptical, Radiated Fields, Radiation resistance, Field regions & Directivity, Current distribution, Radiated Fields.

Practice:

Design of Half wave Dipole Antenna

Design of Monopole Antenna

Module III: Module 3: Aperture Antennas:(4 Hours)

Huygens ' Principle, radiation from rectangular and circular apertures, design considerations, Babinet's principle, Radiation from sectoral and pyramidal horns, design concepts.

Practice:

Design of Horn Antenna

Design of Parabolic Antenna

Module IV: Module 4: Broadband Antennas: (3 Hours)

Broadband concept, Log-periodic antennas, frequency independent antennas,

Antennas For Satellite communication.

Practice

Design of Circular antenna Simulation for UWB

Design of Log Periodic Dipole Antenna

Module V: Microstrip Antennas: (3 Hours)

Basic characteristics of microstrip antennas, feeding methods, methods of analysis, design of rectangular and circular patch antennas

Practice

Design of Microstrip Antenna Array Simulation

Design of Microstrip Antenna Simulation

Module VI: Module 6: Antenna Arrays: (6 Hours)

Analysis of uniformly spaced arrays with uniform and non-uniform excitation amplitudes, extension to planar arrays.

Practice

Design of Helix Antenna Simulation

Module VII: Basic Concepts of Smart Antennas: (3 Hours)

Concept and benefits of smart antennas, fixed weight beamforming basics, Adaptive beamforming

Practice

Design of 5G phased array antenna design and beamforming

Text Books:

1. C. A. Balanis, "Antenna Theory and Design", 3rd Ed., John Wiley & Sons., 2005.

2. W. L. Stutzman, and G. A. Thiele, "Antenna Theory and Design", 2nd Ed., John Wiley & Sons., 1998.

3. R. S. Elliot, "Antenna Theory and Design", Revised edition, Wiley-IEEE Press., 2003.

Reference Books:

1. G.S.N. Raju, “Antennas and Wave Propagation”, Person Education.

Source of reference

https://volakis.eng.fiu.edu/teaching

https://www.udemy.com/course/horn-antennas-design-simulation-optimization

https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-661-receivers-antennas-and-signals-spring-2003/lecture-notes

https://nptel.ac.in/content/syllabus_pdf/117107035.pdf