ROS Robotics: Autonomous Mobile Robots Diploma

Diploma Description:

This course provides the deep concepts to Robotics and the Robot Operating Systems (ROS). Students will learn how to program robots using ROS, a popular open-source software framework for developing robotic software. Topics covered include robot kinematics and dynamics, motion planning, sensor integration, and robot control.

Diploma Outline:

Module 1: Introduction to Robotics and ROS

- Introduction to Robotics and its Applications
- Overview of the Robot Operating System (ROS)
- ROS architecture and components
- Installation and setup of ROS

Module 2: ROS Tools and Concepts [File & Eco systems]

- Introduction to ROS tools, such as RViz, rostopic, and rqt
- Topics and messages in ROS
- ROS nodes and communication
- ROS packages and launch files
- RPLIDAR ROS Drivers
- Kinect360 ROS Drivers
- Install and Run Turtlebot3 Robot in simulation

Module 3: ROS Node Development

- Writing ROS nodes in Python and C++
- Creating ROS publishers for data publishing
- Creating ROS subscribers for data-receiving
- Custom ROS MSG with publisher and subscriber

Module 4: differential drive Robot Modelling with URDF

- Introduction to URDF (Unified Robot Description Format)
- Robot models and joint coordinates
- URDF elements for robot description (links, joints, sensors, etc.)
- URDF visualization with RViz

Module 5: Gazebo Simulation with ROS and Plugins

- Gazebo models and plugins
- Introduction to Gazebo plugins
- Gazebo simulation of URDF robots
- Simulating a Differential Drive Robot Using ROS
- Gazebo models and plugins for differential drive robots

Module 6: Converting SolidWorks Design to URDF

- Challenges and considerations when converting SolidWorks designs to URDF.
- Tools and plugins are available to export SolidWorks models to URDF format.
- Step-by-step process for converting SolidWorks designs to URDF files.
- Validation and debugging techniques for ensuring compatibility and accuracy of URDF models.
- Integration of URDF models into ROS for simulation and visualization.

- Practical exercises involving the conversion of SolidWorks designs to URDF and simulation using ROS and Gazebo.

Module 7: Building Maps Using Gmapping SLAM Algorithm

- Introduction to Simultaneous Localization and Mapping (SLAM) and its significance in robotics.
- Overview of the Gmapping SLAM algorithm and its implementation in ROS.
- Explanation of SLAM concepts including mapping, localization, and loop closure detection.
- Configuring and launching Gmapping within ROS for mapping environments.
- Techniques for optimizing SLAM parameters to improve mapping accuracy and efficiency.

Module 8: Robot Autonomous Navigation using Move Base

- Introduction to autonomous navigation and its importance in robotics applications.
- Overview of the Move Base package in ROS and its role in navigation planning and execution.
- Understanding the components of Move Base including global and local planners, costmaps, and recovery behaviors.
- Configuration and parameter tuning of Move Base for specific robot platforms and environments.
- Integration of sensor data (e.g., LIDAR, RGB-D camera) with Move Base for obstacle detection and avoidance.
- Implementation of global and local path planning algorithms within Move Base for efficient navigation.
- Techniques for handling dynamic obstacles and map updates during navigation.

Module 9: Differential Drive Robot Kinematics

- Mathematical modeling of the robot
- Forward kinematics of a differential robot
- Inverse kinematics

Module 10: Mechanical and Electrical Hardware Assembly with Bill of Materials (BOM).

- Overview of common mechanical components (e.g., frames, actuators, sensors) and electrical components (e.g., motors, controllers, power supplies) used in robotics systems.
- Hands-on experience with mechanical assembly techniques, including fastening, mounting, and alignment of components.
-
- Integration of mechanical and electrical components into robotic systems according to design specifications and BOM.

Module 11: Low-Level Control Arduino and Interface with ROS Control

- Introduction to low-level control and its role in robotics systems for interfacing with hardware components.
- Overview of Arduino microcontrollers and their suitability for low-level control tasks in robotics.
- Understanding the structure and syntax of Arduino code for controlling actuators, sensors, and other hardware peripherals.
- Integration of Arduino-based control systems with ROS through the ROS serial communication protocol.
- Introduction to ROS Control package and its role in providing hardware abstraction and interfacing with robot actuators and sensors.
- Hands-on exercises involving the development and implementation of Arduino code for controlling robot actuators and sensors, interfacing with ROS Control for higher-level motion planning and coordination.

Module 12: Setting up Raspberry Pi with Ubuntu Server and ROS from Scratch

- Overview of ROS compatibility with Ubuntu Server on Raspberry Pi and considerations for selecting appropriate ROS distributions.
- Understanding the installation process of Ubuntu Server on Raspberry Pi, including image download, flashing, and initial configuration.
- Configuring Ubuntu Server for ROS installation, including setting up repositories, updating packages, and installing dependencies.

- Installation of ROS on Ubuntu Server running on Raspberry Pi using standard installation methods (e.g., binaries, source compilation) and selecting suitable ROS distributions (e.g., Melodic, Noetic).
- Configuring network settings and ROS environment variables on the Ubuntu Server for seamless integration with the ROS ecosystem.
- Integration of additional hardware components commonly used in robotics projects with Raspberry Pi running Ubuntu Server, including sensors, actuators, and motor controllers.

Module 13: Setting up Jetson Nano with jetpack Ubuntu and ROS from Scratch **New*

- Introduction to Jetson Nano and operating system (Ubuntu Jetpack)
- Select the right Release of Jetpack and Install it on Jetson
- Power modes for jetson
- Internet connection for Jetson nano
- Install ROS on Jetson Nano
- Install All dependencies and Run the robot in Real using Jetson

Module 14: Skid Steering Robot Modelling with URDF and Gazebo Simulation **New*

- Introduction to macros and parameters in Xacro
- Robot models and joint coordinates using macros
- URDF elements for robot description (links, joints, sensors, etc.)
- URDF visualization with RViz
- Gazebo simulation of URDF robot
- Simulating a Skid Steering Robot Using ROS
- Gazebo models and plugins for Skid Steering robots

Module 15: Skid Steering Robot Kinematics **New*

- Mathematical modeling of the robot
- Forward kinematics
- Inverse kinematics
- Implementation in Code

Module 16: Advanced Navigation *New

- Tunning For move base parameters for Real Robot
- Troubleshooting move-base issues
- Add depth camera data to the navigation stack for obstacle avoidance

Module 17: SLAM and localization using ROS cartographer pkg *New

- Introduction to Grapth SLAM.
- Overview of the Cartographer SLAM algorithm and its implementation in ROS.
- Explanation of SLAM concepts including mapping, localization, and loop closure detection.
- Configuring and launching cartographer within ROS for mapping environments.
- Setup cartographer on PC and Jetson nano
- Run cartographer without encoder data (IMU and Lidar)only
- Custom configuration to run cartographer with navigation stack (Move base)

Extra Topics

Modern C++ for Robotics *New

- STL library
- ROS nodes in OOP

Start Working with ROS2 *New

- Introduction to ROS2.
- ROS1 VS. ROS2
- ROS2 echo system and Middleware
- Create ros2 C++ pkg
- write nodes in ros2 Pubsub system

Diploma Projects:

Project1: Simple Obstacle Avoidance Robot in Simulation

- Clone, build and prepare the environment of Turtlebot3
- Develop ROS program that makes the robot avoid obstacles using Laser scan data

Project2: Ball following Robot in Simulation **New*

- Create a URDF model of the mobile robot
- Integrate the robot model into the Gazebo simulation
- Add Gazebo plugins for wheel controllers and sensor readings
- Develop ROS program that makes the robot follow the white ball in gazebo

Project3: differential drive(Two wheels) Autonomous Mobile Robot in Simulation and Real

- Create a URDF model of the mobile robot
- Add wheels and sensors to the robot model
- Visualize the robot model in RViz
- Integrate the robot model into the Gazebo simulation
- Add Gazebo plugins for wheel controllers and sensor readings
- Verify the robot's movement and sensor data in the Gazebo
- Implement autonomous navigation using the ROS navigation stack
- Add a map and localization to the robot in Gazebo and Real
- Test the robot's navigation in Gazebo and Real

Project4: skid steering (Four Wheels) Autonomous Mobile Robot in Simulation and Real

- Create a URDF model of the mobile robot
- Add wheels and sensors to the robot model
- Visualize the robot model in RViz

- Integrate the robot model into the Gazebo simulation
- Add Gazebo plugins for wheel controllers and sensor readings
- Verify the robot's movement and sensor data in the Gazebo
- Implement autonomous navigation using the ROS navigation stack
- Add a map and localization to the robot in Gazebo and Real
- Test the robot's navigation in Gazebo and Real

Prerequisites:

Basic programming skills in Python or C++. Familiarity with the Linux operating system is recommended. Prior experience in Robotics or ROS is optional, but basic physics and linear algebra knowledge is helpful.