Special topics in Robotics: Robot Operating Systems (ROS) and Applications

The objective of this course is to provide students with a comprehensive understanding of Robot Operating Systems (ROS) and its application. Students will learn the fundamentals of ROS architecture and the underlying concepts such as nodes, topics, and messages. They will be able to develop and test ROS packages for various applications including robot navigation, localization, and manipulation. Additionally, students will gain knowledge about various tools and libraries available in ROS for developing robot software.

Upon completion of this course, students will be able to:

- Understand the fundamental concepts of ROS architecture.
- Create and run ROS nodes, topics, and messages.
- Develop ROS packages for various applications including robot navigation, localization, and manipulation.
- Use various tools and libraries available in ROS for developing robot software.
- Apply the knowledge gained to develop robotic applications using ROS.

Overall, this course will equip students with the necessary knowledge and skills to work with Robot Operating Systems and apply it to various real-world scenarios in robotics.

Week Number	**Topics**	**Description**
1	Introduction to Robotics and ROS	• Overview of Robotics and its applications • Introduction to ROS architecture and its components • Setting up the ROS environment on a local machine
2	ROS Topics and Messages	• Understanding ROS topics and messages • Creating and publishing messages • Subscribing to and processing messages **Hands-on:** Create a simple publisher-subscriber scheme (preferably talker-listener) and grasp the concept of topics and messages from rqt_graph.
3	ROS Nodes and Launch Files	• Creating ROS nodes in Python • Launching multiple nodes using launch files • Managing node communication **Hands-on:** Follow the documentation from the official website of ROS to create ROS nodes for specific application scenarios. Also, implement a launch file which initiates multiple ROS nodes.

4	Robot Motion Control	• Overview of robot motion control • Robot kinematics and dynamics • Controlling robot motion using ROS **Hands-on:** Experiment with ROS-enabled robotic agents and learn about the physics behind the control and dynamics of the robot. Also, create a launch file and deploy it into the robot to control the robot manually through keyboard commands.
5	Robot Localization	• Overview of robot localization • ROS interfaces for localization sensors • Using ROS to localize a robot **Hands-on:** Integrate IMU, camera, and LiDAR sensors with a ROS-compatible robot. Create specific ROS-topics for each sensor and visualize the data on RViz. Train the robot to localize itself at a given place using the sensor data.
6	Robot Mapping and Navigation	• Introduction to robot mapping • Simultaneous Localization and Mapping (SLAM) • Using ROS for robot mapping • Overview of robot navigation • Path planning and obstacle avoidance • Using ROS for robot navigation **Hands-on:** Create a map for an enclosed area with the help of LiDAR data and using the saved map try to navigate the robot to any specific place of the area autonomously.
7	Robotic and Network Simulators	Learn about the working basics of Physics based simulators and Network simulators. **Hands-on:** • Install a physics and a network simulator under ROS environment and try to extract the topic, node, and messages generated by those simulators through ROS • Design a terrain and deploy simulated robots inside of it • Design a simple network topology using the network simulator
8	Addressing Synchronization Issue during Co-simulation	• Learn about the phenomena of synchronization issue while doing co-simulation of physics and network simulators • The student is expected to formulate the project on this topic **Hands-on:** Create a co-simulation environment without any synchronization effort to experience the data transmission mismatch between physics and network simulators. Implement existing approaches to solve the issue.

9	Exploring Features of ROS2	• Why is ROS2 important? • ROS1 to ROS2 • Fundamental differences between ROS1 and ROS2 **Hands-on:** Install ROS2 and learn about the fundamental difference with ROS1 at code level. Explore the impact of master-based and masterless architecture of data transmission rate, and packet loss.
10	ROS2 QoS Policy Design	Read about the components of ROS2 QoS policy design and learn about the impact of each one of them. **Hands-on:** Choose a specific communication scenario and design a suitable QoS policy for that scenario to increase packet delivery rate and decrease latency. Deploy that designed policy with actual robots and validate the simulation performance.

Resources:

Reference papers:

- Exploring the performance of ROS2
- Simulation tools for robotics research and assessment
- CPS-Sim: Co-Simulation for Cyber-Physical Systems with Accurate Time Synchronization
- FlyNetSim: An Open Source Synchronized UAV Network Simulator based on ns-3 and Ardupilot
- RoboNetSim: An integrated framework for multi-robot and network simulation
- ROS-NetSim: A Framework for the Integration of Robotic and Network Simulators

Helpful links on ROS:

- Official website
- Tutorials
- ROS index
- ROS discourse
- Getting involved

Implementation on Robots:

- Data collection with ROSbag and analysis
- Swarm robotics with Crazyflie
- Mapping and Navigation with Turtlebot

ROS compatible bots:

- Aerial
- Ground
- Marine