




ROS Robotics Workshop

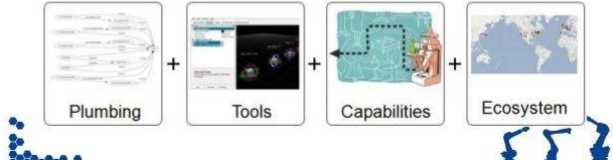
Simulate 7 real-life robots in 4 days!

23rd -26th April, 2019

 ROS: Robot Operating System 



- Open source (BSD)
- Created by Willow Garage
- Maintained by Open Source Robotics Foundation (OSRF)



Plumbing + Tools + Capabilities + Ecosystem

ROS is the default middleware used for commercial and industry robots worldwide and has an active user community along with support from ROS industrial. Originally developed in 2007 at the Stanford Artificial Intelligence Laboratory, today ROS is the de-facto standard for robot programming.

What you will learn:

You will learn how to establish communication with robots using distributed computing techniques, and will learn how to control and simulate a robot in an environment, using a simple joystick as a controller for the robot.

Benefits

- ROS is the meta OS that industrial robots use & hence it is an employable skill. ROS engineer is responsible for R&D, Load testing and design of real life industrial robots.
- With base from Stanford University and a very strong community base, ROS is very popular in Robotics and machine vision R&D work.
- ROS is an applied skill & students can apply theory with real life robotics simulation.

For further enquiries, contact:

Dr. Ankush Ghosh (+91) 9748816774

Course Detail:

Day 1

- ROS Architecture and history
- ROS master nodes and topics
- Console commands
- Catkin workspace and build system
- Launch-files
- ROS messages, ROS node, ROS Topic
- CatKin and CMAKE in details

Day 2

- Robot 1# A software chatter & talker application
- Robot 2# TurtleSIM teleops with XBOX
- Robot 3# Turtle 2 robot through Rviz & keyboard teleop

Day 3

- ROS package structure
- ROS C++ client library (roscpp)
- ROS subscribers and publishers
- ROS parameter server
- ROS Service
- Robot 4# Simulate Star Wars R2D2 Robot
- Robot 5 # Simulate custom made 2 wheel robot with caster wheel
- Talker application in C++ and Python

Day 4

- rqt User Interface
- Robot models (URDF)
- Simulation descriptions (SDF)
- XML in details
- ROS services
- ROS actions (actionlib)
- Robot 6# Simulate Hector quadcopter with X box controller
- Robot 7# Simulate a custom-made robotic arm

The course will use Python and C++ as needed in ROS stack. The students would need to carry a low cost Xbox controller on their own.

Students would need to install the below packages.

Software	Comments
Ubuntu 16.04 LTS	Can be installed using VMWare or Oracle Virtual Box
ROS Kinetic or Indigo	Command line ROS installation
Visual Studio community edition	IDE for coding in C++ and Python
gedit, jstest-gtk	ROS-Linux packages
Xml editor as xml spy	Free version or trail version