THE CONSTRUCT

COURSE CATALOG

ROS eLearning Solutions Built for Campus

November 2020

The Construct
Gran Vía de les Corts Catalanes 608, 3-D,
Barcelona, SPAIN
www.theconstructsim.com
info@theconstructsim.com

ROBOTICS THEORY

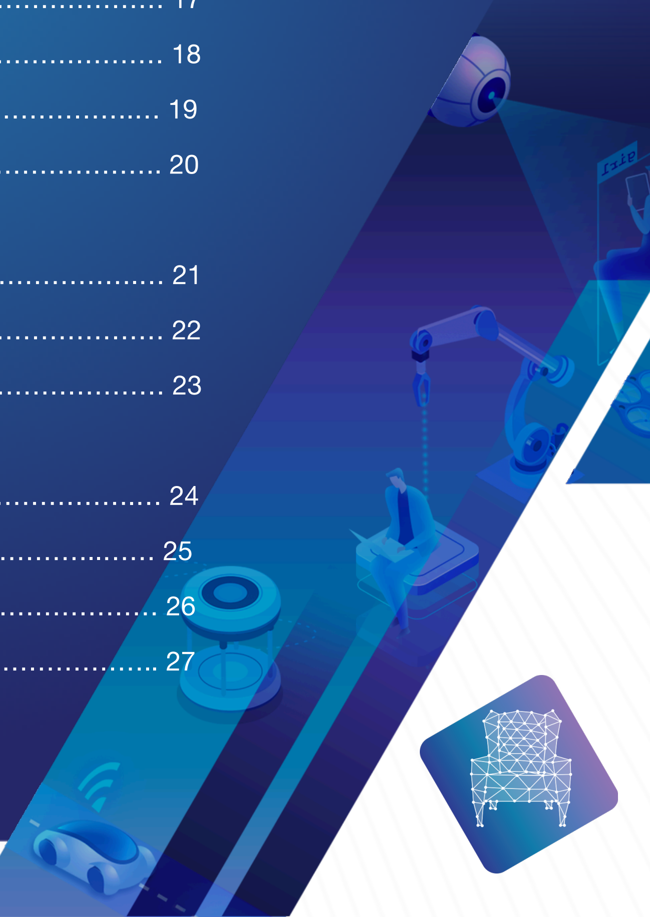
ROBOT AND ARTIFICIAL INTELLIGENCE

CODE FOUNDATION FOR ROS

BASIC ROS

ROS DEBUG

NAVIGATION



PERCEPTION

ROBOTICS HARDWARE

MANIPULATION

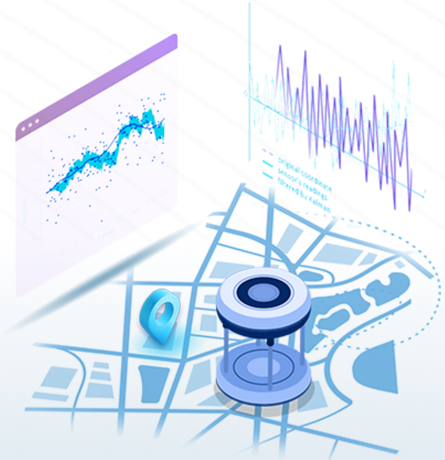
WEB DEVELOPMENT FOR ROBOTS

COURSES OF ROBOTICS PRODUCTS

Kalman Filters

ABOUT THIS COURSE

Learn how Kalman filters work and how to apply them to mobile robots using ROS.



One of the most common problems in robot navigation is knowing where your robot is localized in the environment (known as robot localization). In this field, Kalman Filters are one of the most important tools that we can use.

With this course, you will understand the importance of Kalman Filters in robotics, and how they work. You will learn the theoretical meaning, and also the Python implementation. Finally, you will apply the studied filters to mobile robots using ROS.

Learning Objectives

- What a Kalman Filter is and why they are required
- Different types of Kalman Filters and when to apply each one
- Bayesian Filters
- One-dimensional Kalman Filters
- Multivariate Kalman Filters
- Unscented Kalman Filters
- Extended Kalman Filters
- Particle Filters

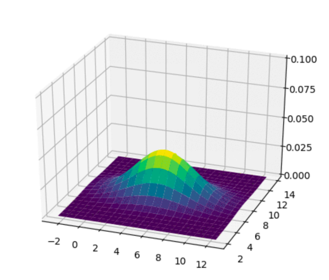
Intermediate

30 hours

6 lectures

TurtleBot 2

COURSE EXERCISES



- **Bayes filter**: Implement a Bayes filter in order to know a robot's position.
- **EKF and UKF**: Compare the EKF and UKF filters' performance using the robot_localization ROS package.
- **Kalman filter**: Implement a Kalman filter and test it in a simulated robot.
- **Course Project**: Implement a pose estimation system combining different nonlinear filtering methods.

Basic Mobile Robots Kinematics

ABOUT THIS COURSE

Understand the concepts of rigid motions in 3D space, and learn the movement equations for holonomic and non-holonomic robots.

This course will teach you the basic motion equations required to work with robots.

You will understand the different types of movement, orientation and equations that describe movement of wheeled robots, as well as, how to make a wheeled robot follow a trajectory.

You will be learning the mathematical equations, but also program them in the simulated robots, so you will be able to see the results on an actual robot.

The final project will ask you to apply all that knowledge to create a program to make a floor cleaner robot cover the whole space and clean it.

Learning Objectives

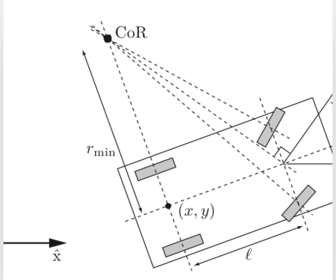
- Learn and understand rigid body motions in 3D: Frames of reference, Rotations in 3D, and Homogeneous Transformations.
- Kinematics for non-holonomic robots
- Kinematics for holonomic robots
- Kinematics control

Basic

14 hours

6 lectures

COURSE EXERCISES



- **Non-Holonomic Motion**: Program the equations of a wheeled robot.
- **Odometry Computation**: Compute and visualize odometry for holonomic, non-holonomic and car-like robots.
- **Car-Like Kinematics**: Understand the kinematic equations of car-like robots.
- **Program a Floor Cleaning Robot:** On the final project, program a Roomba robot to cover the whole space while cleaning.

Basic Arm Kinematics

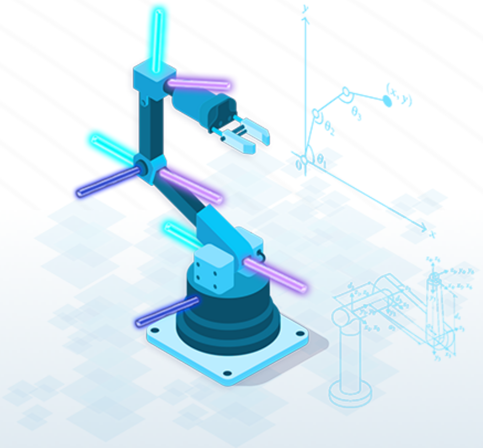
ABOUT THIS COURSE

Learn the basic kinematic concepts of robotic arms

The aim of this course is to provide you with the concepts and tools that will enable you to understand how the motion of a manipulator works. This is the cornerstone of robot manipulation, as it can be later expanded upon with the addition of external sensors in other courses.

Learning Objectives

- Learn some basic concepts needed in order to know how to deal with a manipulator robot.
- Learn how to compute the Forward Kinematics of a manipulator robot.
- Learn how to compute the Inverse Kinematics of a manipulator robot.

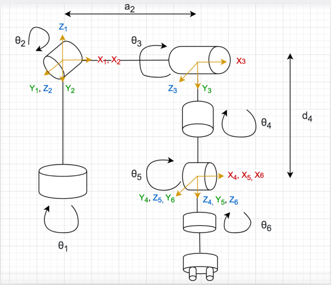


Intermediate

30 hours

5 lectures

COURSE EXERCISES

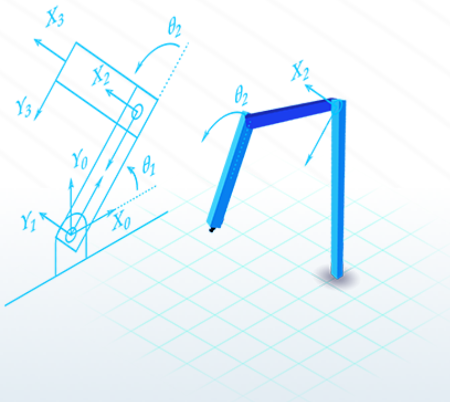


- **Robot Configuration:** Analyze the kinematic configuration of your manipulator robot for further analysis.
- **Forward Kinematics:** Compute the Forward Kinematics of your robot both theoretically and using ROS tools.
- **Inverse Kinematics:** Compute the Inverse Kinematics of your robot both theoretically and using ROS tools.
- **Final Project:** Solve the Forward and Inverse Kinematics of the UR3 robot manipulator.

Robot Dynamics and Control

ABOUT THIS COURSE

Learn to develop dynamic models and intelligent control systems for simple robots.



Robot Dynamics is really important since it will give you a complete understanding not only how robots move (kinematics) but also WHY they move (dynamics). In this course, you will learn to develop the dynamics models of basic robotic systems, as well as create intelligent controllers for them.

Learning Objectives

- How to solve the dynamics for the motion of rigid bodies in 3D space with the use of Newton's laws of motion
- How to model the dynamics of a simple robotic system and how to derive its equations of motion
- How to create a full state feedback controller to allow a robotic system to balance

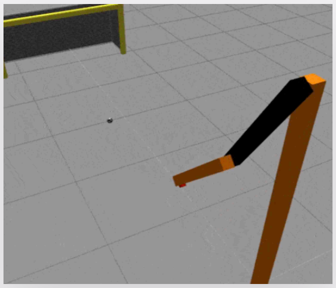
Intermediate

15 hours

5 lectures

RRBot robot manipulator

COURSE EXERCISES



- **Rigid Body Dynamics**: Learn the basic dynamics of rigid bodies using a box and a table.
- **Dynamic Modeling**: Learn how to model the dynamics of a simple robotic system using an RRbot.
- **Feedback Control:** Create a full state feedback controller to allow the RRbot to balance.
- **Score a goal!:** Program a simple dynamic controller for RRBot that will move the robot so that it kicks a ball on the floor.

Deep Learning with Domain Randomization

ABOUT THIS COURSE

Learn how to train any robot to recognize an object and pinpoint its 3D location with only an RGB camera and a lot of training with Keras.

Welcome to this micro-course! This course is intended for the people that want to learn about deep learning using Keras.

In this case, we use a very interesting approach to learning, which is Environment Randomization. This method exploits the versatility of environment generation in simulations to train a robot in a way that the resulting model is very robust, no matter the lighting conditions. It also makes the transition from simulated learning to reality much smoother and faster. Learn through hands-on experience how to train a robot for 3D object recognition using random environments.

Keras will be the cornerstone of this system and you will learn all the necessary skills to generate training data, convert it to a database, train a MobileNetV2 model, retrain it, and make predictions with it.

The final project is the training of a garbage picking robot, from training data generation to the final garbage detection and picking program. Dive into the fantastic world of Deep Learning with Keras right now!

Advanced

7 hours

6 lectures

Fetch Robot
GarbageCollector Robot

COURSE EXERCISES

- **Create a Simple Random Environment:** Start with a simple environment training.
- **Exercises For XY motion Spam:** Train the simulation to find the Spam anywhere on the table.
- **Exercises for a Distractor & Random Environment:** Train the model with a random environment.
- **Microproject Garbage Collector:** Use the two-wheeled Magician robot to find an object amidst other distractions.

ROS Autonomous Vehicles 101

ABOUT THIS COURSE

Introduction to Autonomous Vehicles in the ROS Ecosystem

The goal of this course is to show you the basic knowledge that you need to master in order to program Autonomous Cars for a Level 3 of autonomy. This means that it is expected that all tasks should be performed autonomously, but at the same time, it is expected that a human driver will intervene whenever necessary. This level of autonomy is called conditional automation.

We would like to explain some more advanced subjects to you, like obstacles and traffic signal identification based on deep learning. Those subjects are required for a full Level 4 of autonomy (there is no need for a driver, everything will be handled by an automated system). There is no need for human driver intervention at this level, which is called High Automation. This kind of autonomous system will work in a particular area under specified weather conditions. But you will have learn these topics in a more advanced course.

Learning Objectives

- What are the sensors required for an autonomous car and how to access them using ROS
- How to do autonomous navigation using a GPS
- How to create an obstacle avoider for an autonomous car
- How to interface ROS with a car that follows the DBW interface

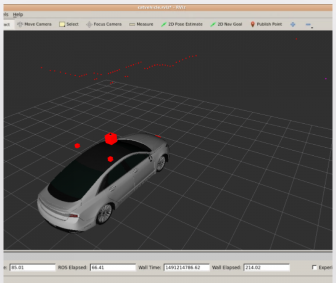
Advanced

12 hours

6 lectures

Simulated DBW MKZ Autonomous Vehicle

COURSE EXERCISES



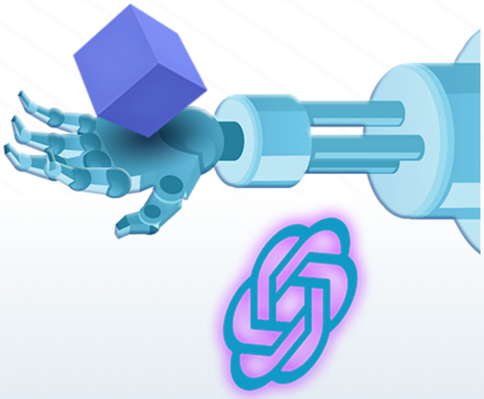
e: 85.01 ROS Elapsed: 66.41 Wall Time: 1491214786.62 Wall Elapsed: 214.02

- **Learn the Most Essential Sensors for autonomous vehicle control:** Laser, Camera, and GPS.
- **Practice Moving the Car with GPS Data:** Have exercises that will help you to get used to the movement of the car and navigating with GPS.
- **Detect Obstacles & Security:** Implement an obstacle detection system using the laser data
- **Use of a bus standard named CAN:** Using CAN-Bus messages instead of topics to move the car and retrieve the GPS data.

Using OpenAI with ROS

ABOUT THIS COURSE

Use the power of OpenAI combined with ROS simulations in the easiest way



In this Course, you are going to learn how to use the OpenAI ROS structure developed by The Construct and how to generate new code for it. The OpenAI ROS structure will allow you to develop for OpenAI with ROS in a much easier way.

Learning Objectives

- Understand the basic concepts of the OpenAI ROS structure
- Set up the OpenAI ROS structure for a CartPole environment
- Train the Cartpole with the Qlearn algorithm
- Set up the OpenAI ROS structure for a Moving Cube environment
- Train the Cube with the Qlearn algorithm
- Modify the learning algorithm: DeepQ
- Set up the OpenAI ROS structure for a Fetch Robot
- Train a Fetch robot with the HER algorithm from OpenAI baselines

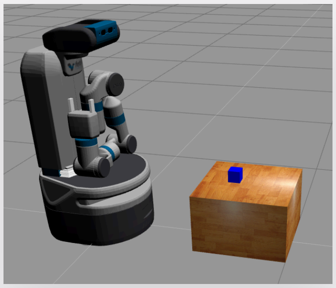
Advanced

26 hours

11 lectures

Fetch Robot,
CartPole
Hopper
RoboCube.

COURSE EXERCISES

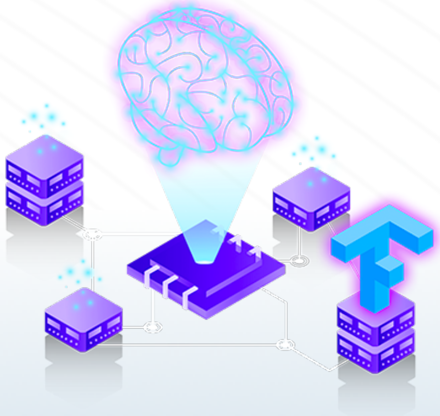


- **Training a Fetch Robot:** Build the Task Environment for training a Fetch robot.
- **Training a Hopper robot:** Create environments to train the one-legged Hopper robot.
- **Apply openai_ros to a robot:** Apply the openai_ros package to a cube robot.
- **Using the CartPole simulated environment:** Understanding the ROS + OpenAI structure.

ROS Deep Learning with TensorFlow 101

ABOUT THIS COURSE

The first step with ROS, Deep Learning, Tensor Flow, and Image Recognition



DeepLearning… A topic that we hear a lot about and that promises a lot. Deep learning is the technology behind intelligent drone flight, self-driving cars, robots recognizing a huge number of objects, people tracking video feeds, etc. But how can it be used?

In this course, you will focus on learning the essentials for doing image recognition with Deep Learning. You won't learn each and every nook and cranny of deep learning, but the few elements you learn you will be able to put into use and apply in real life, integrated with ROS.

Learning Objectives

- Use the Google TensorFlow Image Recognition DB to recognize hundreds of different objects with ROS
- Generate your own TensorFlow Inference graph to make it learn a custom object.
- Use TensorBoard Web visualizer to monitor how the learning process is going.

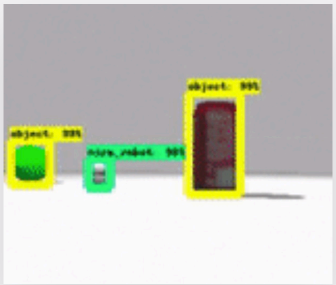
Advanced

9 hours

6 lectures

Mira Robot
GarbageCollector Robot

COURSE EXERCISES



- **Integrate ROS + TensorFlow:** Create your own ROS package that recognizes images with TensorFlow.
- **Train your model:** Train your own TensorFlow Image Recognition Model.
- **Image Detection:** Detect with your new retrained model in ROS.
- **Garbage Collecting Robot Challenge:** Image recognition/ learning with TensorFlow.

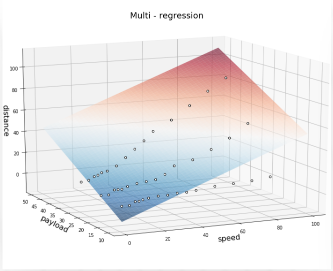
Machine Learning for Robotics

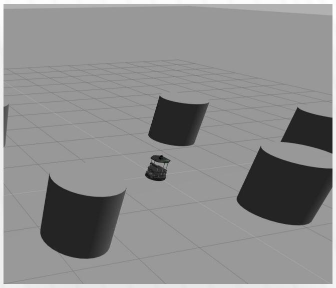
ABOUT THIS COURSE

Learn about machine learning concepts applicable to robotics, and understand the fundamental principles of artificial intelligence.

This course is dedicated to robot enthusiasts and all the others who would like to stay abreast of current technology trends, which change the world and have tremendous impact on human life. This course will give you a state-of-the-art opportunity to familiarize yourself with the general concepts of machine learning and see how the theory is executed in practice. We will discuss the main machine learning techniques and concepts. We will deploy discussed techniques (algorithms) in our domain in order to boost robot performance. Moreover, we are going to play with neural networks and understand this concept from scratch.

This course is an amazing journey and opportunity to capture important knowledge, which accompanies robotics daily performance process, control, and development more and more.





Advanced

21 hours

11 lectures

TurtleBot 2

COURSE EXERCISES

- **Estimate future positions of the robot:** You will run the robot and run linear regression – to be able to estimate future positions of the robot for the captured data set.
- **Obstacle avoidance with a mobile robot using Neural Network (NNs):** Let the robot learn the NNs to avoid a collision with the wall without using sensors.
- **Obstacle Detection using the K-means Algorithm:** Discover the obstacles in the robot environment (space) and estimate its positions using the k-means algorithm.
- **Use Principal Components Analysis in a ROS Environment:** You are going to perform the simulations in ROS in order to capture the data set that will be used to understand PCA.

Python3 for Robotics

ABOUT THIS COURSE

Master the basics of Python 3 required for robot programming

Do you want to become a ROS Developer? If the answer is yes, then learning Python is mandatory for you.

Python is the most popular programming language for robots, and it is also the faster and easier way to learn ROS. In this Python3 for Robotics course, you will master essential Python3 knowledge in order to get started with ROS smoothly.

Learning Objectives

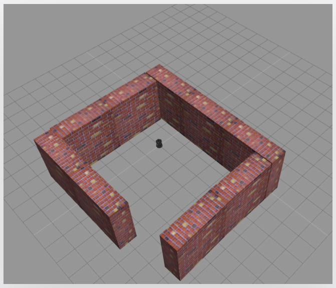
- Store data into Variables
- Operate with the data in the Variables
- Change behavior based on Conditions
- Create Functions that can be called from other parts of the code
- Encapsulate the code into Classes so you can have clean and robust code

Basic

12 hours

6 lectures

COURSE EXERCISES



- **Control Robotic Arm using Python**: Create your first Python program!
- **Define a class in Python**: Create a Python program that makes the robot perform 2 squares.
- **Ultimate Code Foundation Challenge**: Help the TurtleBot Robot get out of the maze (by using Python).

Linux for Robotics

ABOUT THIS COURSE

Learn the Linux fundamentals you'll need for robotics development

At present, ROS is fully supported only by Linux systems. This means any ROS-based robot uses Linux, so if you want to create programs for these robots, you have to get ready with Linux.

In the Linux for Robotics course, you will master essential Linux knowledge in order to get started with ROS.

Learning Objectives

- Navigate through a Linux filesystem
- Interact with a Linux filesystem
- Edit files using the Shell (vi editor)
- Manage access to files (Permissions)
- Create simple Linux programs (Bash Scripts)
- Manage execution of Linux programs (Processes)
- Connect to the remote computer of a robot (ssh)

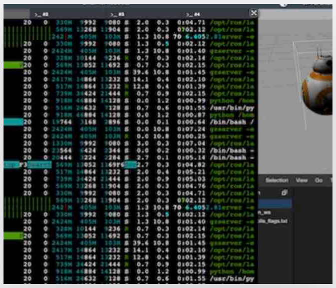
Basic

10 hours

6 lectures

BB-8

COURSE EXERCISES



```
 20   0  330M  9992  9080 S  2.0  0.3  0:04.71 /opt/ros/la
         569M 13268 11904 S  2.0  0.3  0702.12 /opt/ros/la
      242 M  405M  103M S  1.3 10.8 70 6.4052.81erver -e
 20   0  330M  9992  9080 S  1.3  0.5  0:02.12 /opt/ros/la
 20   0 2424M  405M  103M S  1.3 10.8  0:01.40 gzserver -e
 20   0  328M 10144  9236 R  0.7  0.3  0:02.14 /opt/ros/la
 20   0  569M 13052 11692 S  0.7  0.3  0:02.15 /opt/ros/la
 20   0 2424M  405M  103M S 39.6 10.8  0:01.45 gzserver -e
 20   0 2417M 14864 13232 S 14.1  0.4  0:02.10 /opt/ros/la
 20   0  517M 14864 13232 R 12.8  0.4  0:01.39 /opt/ros/la
 20   0  739M 34424 21444 R  0.7  0.9  0:02.15 /opt/ros/la
 20   0  918M 46884 14128 S  0.0  1.2  0:00.99 python /hom
 20   0  516M 24632  7128 S  0.7  0.6  0:01.55 /usr/bin/py
 20   0  918M 46884 14128 S  0.0  1.2  0:00.87 python /hom
 20   0 19764  3168  2896 S  0.0  0.1  0:00.64 /bin/bash /
 20   0 2424M  405M  103M S  0.0 10.8  0:07.24 gzserver -e
 20   0 2424M  405M  103M R  0.0 10.8  0:00.25 gzserver -e
 20   0 1330M  9992  9080 S  3.0  0.3  0:07.04 /opt/ros/la
 20   0 23564  6424  2344 S  0.0  0.2  0:00.32 /bin/bash -
 20   0 20444  3224  2284 S  2.7  0.1  0:05.14 /bin/bash -
    F3       569M 13052 11692 F6  2.7  0.3  0:04.82 /opt/ros/la
 20   0  517M 14864 13232 S  2.0  0.4  0:05.21 /opt/ros/la
 20   0  739M 34424 21444 S  2.0  0.9  0:05.03 /opt/ros/la
 20   0  569M 13268 11904 S  2.0  0.3  0:04.76 /opt/ros/la
 20   0  330M  9992  9080 S  2.0  0.3  0:04.71 /opt/ros/la
         569M 13268 11904 S  2.0  0.3  0702.12 /opt/ros/la
      242 M  405M  103M S  1.3 10.8 70 6.4052.81erver -e
 20   0  330M  9992  9080 S  1.3  0.5  0:02.12 /opt/ros/la
 20   0 2424M  405M  103M S  1.3 10.8  0:01.40 gzserver -e
 20   0  328M 10144  9236 R  0.7  0.3  0:02.14 /opt/ros/la
 20   0  569M 13052 11692 S  0.7  0.3  0:02.15 /opt/ros/la
 20   0 2424M  405M  103M S 39.6 10.8  0:01.45 gzserver -e
 20   0 2417M 14864 13232 S 14.1  0.4  0:02.10 /opt/ros/la
 20   0  517M 14864 13232 R 12.8  0.4  0:01.39 /opt/ros/la
 20   0  739M 34424 21444 R  0.7  0.9  0:02.15 /opt/ros/la
 20   0  918M 46884 14128 S  0.0  1.2  0:00.99 python /hom
 20   0  516M 24632  7128 S  0.7  0.6  0:01.55 /usr/bin/py
```

- **Control BB-8 robot using Linux:** Execute the bash script.
- **Create simple Linux programs:** Work and interact with Linux-based systems.
- **Manage execution of Linux programs:** Learn advanced utilities to interact deeply with a Linux system.
- **Ultimate Code Foundation Challenge:** Program a Parrot AR Drone using Linux.

ROS Basics (Python/C++)

ABOUT THIS COURSE

Learn the fundamentals of ROS to understand and be able to program robots

The ROS BASICS IN 5 DAYS course will take you quickly and smoothly into ROS. You will learn the essential concepts and tools to be able to understand and create any basic ROS related projects.

Learning Objectives:

- Understand key ROS concepts
- Understand and create your own ROS programs
- How to debug your ROS programs
- How to apply theory into real Robotics Challenge and Projects

Who is this for?

- Anyone interested in learning about robotics through practice.
- Beginners who want to quickly learn the fundamental of ROS through practice.

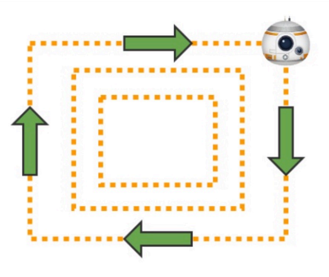
Basic

17 hours

13 lectures

Husky
TurtleBot2
WAM Arm robot
BB-8.
Parrot AR. Drone
Sphero robot

COURSE EXERCISES



- **Mobile robot obstacle avoidance project**: By using Subscribers, Publishers, & Messages.
- **Control the BB-8 robot**: Use ROS Services to change the way BB-8 moves.
- **Control a Drone Robot:** Using ROS Actions to move a parrot drone robot.
- **Win the TurtleBot Race!:** Move a TurtleBot 2 robot through the maze and help it find its way out.

ROS2 Basics for C++

ABOUT THIS COURSE

Learn how to start working with ROS 2

ROS2 is finally here! This introductory course is intended not only for new ROS users, but also original ROS1 users that want to start with ROS2. ROS2 adds a lot of new features that will get traction in the next year. This course will give you the basics for starting ROS2, more than the bells and whistles of ROS2 that are much more advanced topics and will be addressed in their respective specialized courses.

Learning Objectives

- Creation of ROS2 packages
- Management of the new Colcon universal building system.
- Topic Publishers and Subscribers in ROS2 C++.
- New Launch system based on python
- Service servers and client generation for ROS2.
- Basic use of ROS1-Bridge to communicate ROS2 systems with ROS1 systems.
- Use of Debugging tools in ROS2.

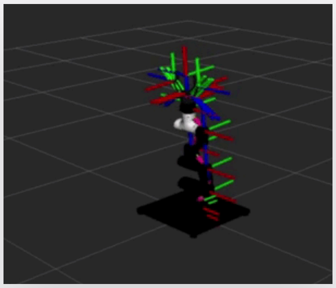
Basic

20 hours

8 lectures

TurtleBot 2
MARA
Husky
WAM Arm robot
BB-8
Parrot AR. drone
Sphero robot

COURSE EXERCISES



- **Move a Robot with ROS2:** Execute a ROS2 program to control the TurtleBot.
- **Launch ROS1-Bridge:** Communicate between ROS1 & ROS2 with ros1_bridge.
- **Visualize Complex data and RViz2:** Learn the basic use of RViz2 debugging tool.

TF ROS 101

ABOUT THIS COURSE

To finally understand TF and Robot State Publishers in ROS

Imagine the following scenario: you have a robot with a laser mounted on it. In order to be able to use the laser data (e.g. to detect an obstacle), the robot needs to know WHERE that laser is mounted in the robot. If you don't provide this information to the robot, when the laser detects that obstacle, how can the robot know where the object is? Is it in front of the robot? Or is it behind it?

Well, this relationship between the different parts (frames) of a robot is what is handled by ROS TFs.

In this TF ROS 101 course, you will be able to understand how TF works by practicing in several hands-on exercises

Learning Objectives

- Visualize TFs
- Publish & Subscribe to TF data
- Understand Transformations & Frames
- Understand Robot State Publishers & Joint State Publishers
- Be comfortable with common TF Command-line Tools (tf_echo, view_frames…)
- Understand Static Transform Publishers

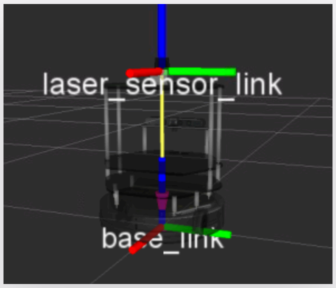
Basic

15 hours

6 lectures

Pi robot,
Multiple Kobuki Robot,
3D Version of the Classical
2D TurtleSim

COURSE EXERCISES



- **Create your own TF publisher:** Create a publisher that publishes the TF of a new model.
- **Move the iRobot:** Create your first TF listener and move the iRobot.
- **Publish the joint_states:** Visualize all the TFs moving at the same time as the real robot.
- **Create your own robot with TF from scratch:** Apply all of the concepts and tools you learned to your robot**.**

Programming Drones with ROS

ABOUT THIS COURSE

Learn all the basics you need in order to start programming autonomous drones

In this Course, you are going to learn all the basics you need in order to start programming autonomous drones using Robot Operating System (ROS). You will learn through hands-on experience from day one! You will be using the Parrot AR Drone in different environments.

Learning Objectives

- Learn basic control of a Drone
- Drone Exploration
- Drone Navigation

Lectures

- Basic Control of a drone
- 2D Navigation with RTABMap
- 3D Navigation with MoveIt!
- PTAM-based visual navigation
- Adapting the PTAM system for other drones

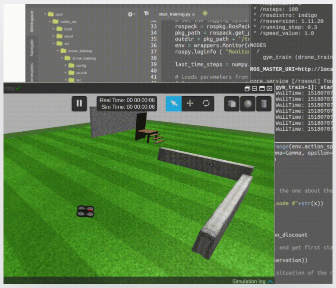
Intermediate

20 hours

5 lectures

Hector Drone,
Parrot AR Drone

COURSE EXERCISES



```
33    rospack = rospkg.RosPack
34    pkg_path = rospack.get_
35    outdir = pkg_path + '/tr
36    env = wrappers.Monitor(e
37    rospy.loginfo ( "Monitor
38
39    last_time_steps = numpy.
40
41    # Loads parameters from
```

```
 * /nsteps: 100
 * /rosdistro: indigo
 * /rosversion: 1.11.20
 * /running_step: 0.3
 * /speed_value: 1.0

NODES
  /
    gym_train (drone_train

ROS_MASTER_URI=http://loca

core service [/rosout] fou
[gym_train-1]: star
WallTime: 15180707
WallTime: 15180707
WallTime: 15180707
WallTime: 15180707
WallTime: 15180707
WallTime: 15180707
WallTime: 15180707
```

```
range(env.action_sp
ma=Gamma, epsilon=

 the one about the
.sode #"+str(x))

n_discount
 and get first sta
servation))
situation of the
```

- **Library Unit Tests:** Create unit tests at a Python-code level**.**
- **ROS unit tests:** Create unit tests at a ROS-node level.
- **ROS Integration Tests:** Create unit tests at a behavioral level.
- **Unit Testing MicroProject:** A small project to put into practice everything you've learned during the course.

ROS Control 101

ABOUT THIS COURSE

Finally understand ROS_Control and how to use it on your robot

ROS Control is a set of packages and tools that allow you, basically, to send commands and communicate with the joints of your robot in order to be able to control them. For instance, you need ROS Control to move the wheels of a mobile robot, or to move the different joints of a robotic arm in an industrial plant. If you don't master ROS Control, you won't be able to make your robots take actions.

The main goal of this course is to teach you how to integrate this ros_control utility within a simulated environment, so you can use this tool to control the joints of your simulated robot.

Learning Objectives

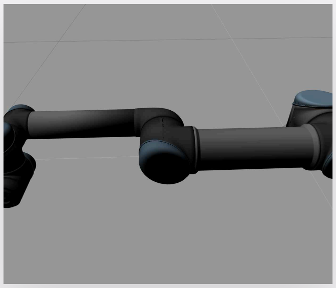
- Learn the essential concepts of ROS_Control
- Configure ROS_Control on a Robot Simulation
- Create a custom controller

Intermediate

7 hours

5 lectures

COURSE EXERCISES



- **Configure the controllers:** Set up your robot's URDF files for using ros_control.
- **Create a controller:** Create a custom ROS controller that sends commands to your joints.
- **ROS Control Challenge:** Apply all the knowledge you've gained during this course to a UR5 robot arm project.

FlexBE with ROS

ABOUT THIS COURSE

Learn the basics of how to use FlexBe with ROS

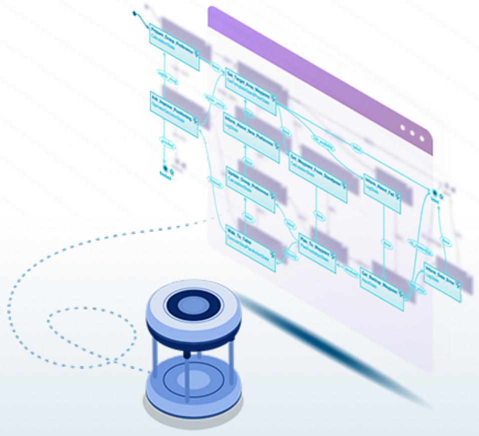
FlexBe is a powerful and user-friendly, high-level behavior engine that helps you to create complex robot behaviors without the need for manually coding them. In this Course, you are going to learn how to use FlexBe with ROS systems.

Learning Objectives

- Basic Concepts of FlexBe Engine
- Create a FlexBe state
- Create a FlexBe behavior
- Create a FlexBe state (Actionlib)
- Understand FlexBe Autonomy Levels
- Understand Unit Testing

Lectures

- Creating a Basic Behavior
- Actionlib States and Autonomy Levels
- Unit Testing
- A Project based on a Parrot AR Drone, where to test everything you have learned during the course.

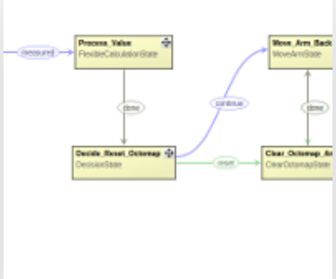


Basics ROS

14 hours

5 lectures

COURSE EXERCISES



- **Create your first FlexBe state:** Create your own FlexBe behavior, and ad new states on it.
- **Create an Actionlib state:** Manage the Autonomy Levels of your behaviors.
- **Create a unit test:** Create Unit Tests for your FlexBe states.
- **FlexBe Challenge:** Apply all of the concepts and tools you learned to a drone.

Unit Testing with ROS

ABOUT THIS COURSE

Learn how to perform Unit Tests with ROS on the 3 main levels of testing: Python tests, ROS tests, and Integration tests

Currently, there exist thousands of ROS packages, and that number is growing every year. As a robotics developer, you will have to constantly update your code. But it is very easy to introduce errors or bugs. So with this in mind, unit testing becomes an essential tool that allows you to make incremental updates to your code more quickly.

Learning Objectives

- Create Python Unit Tests
- Create ROS Unit Tests
- Create ROS Integration Tests

Lectures

- basic concepts related to ROS Unit Testing: testing tools, levels of testing…
- Library Unit Tests
- ROS-Node Level Tests
- ROS Integration Tests

Intermediate

9 hours

5 lectures

TurtleBot2
Parrot AR. Drone

COURSE EXERCISES

ROS Unit Tests

The second level of testing is the ROS Node-Level Unit Tests
working correctly (that is, if it's publishing the expected value
response, etc...

But in order to better understand this, let's practice! In the foll
of our Python Script.

Exercise 3.1

First of all, inside **test** folder, we are going to place a script lik

rotate_robot_test_ros.py

```
In [ ]: #! /usr/bin/env python

        from robot_control.rotate_robot import RobotC
        import rosunit
        import unittest
        import rostest
        import rospy
        from geometry_msgs.msg import Twist
        import time
        #import sys
        PKG = 'robot_control'
        NAME = 'rotate_robot_test_ros'


        class TestRobotControl(unittest.TestCase):
```

- **Library Unit Tests:** Create unit tests at a Python-code level.
- **ROS unit tests:** Create unit tests at a ROS-node level.
- **ROS Integration Tests:** Create unit tests at a behavioral level.
- **Unit Testing MicroProject:** A small project to put into practice everything you've learned during this course.

ROS RViz Advanced Markers

ABOUT THIS COURSE

Learn how to use RViz Advanced Markers for debugging and visualization

Visualizing data in the correct way is vital to extract meaningful conclusions. This is especially true in Robotics.

One of the problems you always tend to have in robotics is knowing what the robot is actually seeing, what is the virtual representation of the world in his mind. It's also very important to represent visually complex data in one place only. That's why RViz and all its markers and plugins have made robotics more user friendly and powerful than ever before!

Learning Objectives

- Understand how to use Basic RViz Markers.
- Create BoundingBoxes Arrays that change dynamically.
- Add Overlay text, graphs and menus in RViz.
- Draw TFTRajectories, RobotFootsteps and occupancy grids that change based on real robot data.
- Draw pictograms from FontAwsome to represent detections and real object in the world
- Represent TwistStamped commands issued to the robot
- Create Interactive displays in RViz that allow to execute programs from RViz with custom icons.

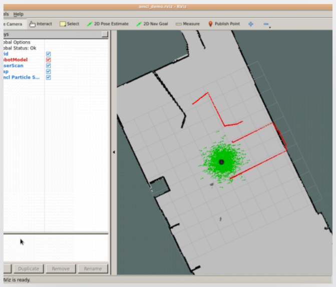
Intermediate

13 hours

5 lectures

Haro Robot

COURSE EXERCISES



- **Create Your First Basic Marker:** Create a publisher that publishes data of a spherical-shaped marker.
- **Use the Special Markers:** Bounding Boxes, Robot Footsteps, Polygon Arrays, Occupancy Grids, and Pictograms.
- **Put Overlays in RVIZ:** Adding Plots, PieCharts, and Menus.
- **Add Custom Panels to RVIZ:** Use panels to turn RVIZ into your robot's GUI command system and give it some interactivity.

Debug Cases

ABOUT THIS COURSE

Learn to Debug ROS code

This Course contains several ROS-related problems that need to be solved by you. Within this Course, you will find several typical ROS-related problems, distributed in different cases, which need to be solved by you. You will find a step-by-step guide on how to find and solve the different issues.

Learning Objectives

- Understand how to detect ROS-related issues
- Understand how to debug ROS-related issues
- Understand how to solve typical ROS issues
- Understand how to get and analyze data from the topics

What You Will Learn

- Case 1: ROS Topics and Messages
- Case 2: ROS Services and Messages
- Case 3: ROS TF's
- Case 4: ROS Actions and Messages
- Case 5: ROS Navigation

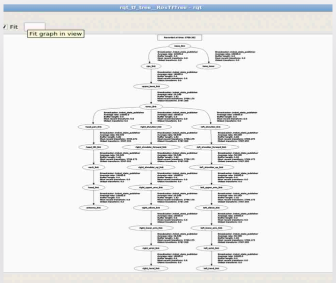
Intermediate

6 hours

6 lectures

Parrot AR.Drone
TurtleBot 2 Robot
Husky
Pi Robot
Summit XL

COURSE EXERCISES



- **ROS Topics & Messages:** Debug ROS issues related to topics and messages.
- **ROS Services & Messages:** Debug ROS issues related to services and service messages
- **Transforms:** Learn to debug ROS issues related to TF’s.
- **ROS Actions & Messages:** Learn to debug ROS issues related to actions and action messages.

Fuse Sensor Data to Improve Localization

ABOUT THIS COURSE

Learn how to fuse GPS, IMU, odometry, and other sources of localization

How can you improve the localization of your robot when you have multiple localization sensors?

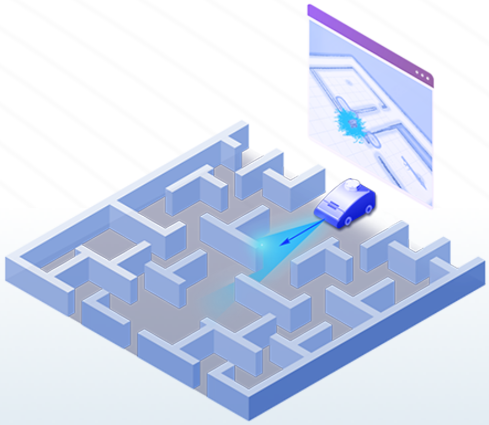
We will solve that localization problem by using the robot localization package.

In this course, you are going to work on the following scenarios:

- Use robot_localization to merge different sensor inputs (IMU, Encoders, etc.).
- Use robot_localization with an external localization system (AMCL).
- Use robot_localization with GPS data.

Learning Objectives

- Merge multiples sensor data using the robot_localization package
- Use the robot_localization package with an AMCL Localization system
- Add GPS data to the robot_localization package.



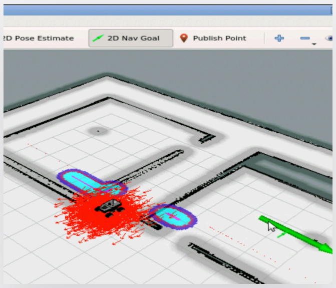
Intermediate

14 hours

6 lectures

Jackal Robot
TurtleBot2
Summit-XL robot

COURSE EXERCISES



- **Merging sensor data:** Use the robot_localization package to merge data from different sensors.
- **Localize the Summit-XL robot:** Use the robot_localization package in combination with an amcl system external to it.
- **GPS Navigation:** Navigate with GPS to unmapped places.

ROS Navigation

ABOUT THIS COURSE

Learn how to make your robot navigate autonomously by using the ROS Navigation Stack

Navigation is one of the most essential tools in ROS. It allows mobile robots to move around autonomously. This is used in many robotics fields: logistic robots for warehouses, domestic robots that perform certain household tasks, entertainment robots, etc.

In ROS NAVIGATION IN 5 DAYS course, you will learn the key concepts involved in ROS Navigation, and how to use it in real robot projects.

Intermediate

8 hours

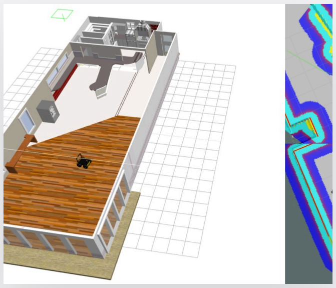
6 lectures

Husky
TurtleBot 2
Summit XL Robot

Learning Objectives

- Set up ROS Navigation Stack on a Robot
- Build a map of the environment from zero
- Perform Robot Localization
- Autonomous Path Planning
- Understand Simultaneous Localization and Mapping (SLAM)
- Obstacle Avoidance

COURSE EXERCISES



- **Mapping:** Build a map of an unknown environment with SLAM.
- **Robot Localization:** Visualize Robot Localization using RViz.
- **Path Planning:** Visualize Path Planning in RViz & Obstacle Avoidance.
- **Robot Navigation Challenge:** Navigate a Summit XL Robot autonomously in an Indoor Environment.

RTAB-Map in ROS 101

ABOUT THIS COURSE

Learn how to use the rtabmap_ros package for performing RGB-D SLAM

RTAB-Map (Real-Time Appearance-Based Mapping) is an RGB-D SLAM approach based on a loop closure detector.

The loop closure detector uses a bag-of-words approach in order to determinate if a new image detected by an RGB-D sensor is from a new location or from a location that has been already visited.

Of course, this is a very summarized explanation. You will get more details on how this loop closure detector works inside this course.

Learning Objectives

- Learn the basics of RTAB-Map
- Use the rtabmap_ros package
- Understand how the loop closure detection works internally
- Create a 3D Map of an environment
- Autonomous Navigation using RGB-D SLAM.

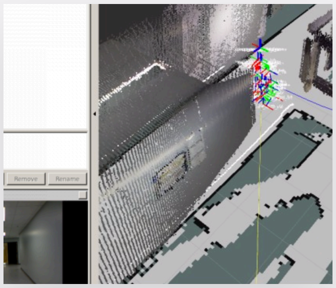
Intermediate

2 hours

3 lectures

TurtleBot 2

COURSE EXERCISES

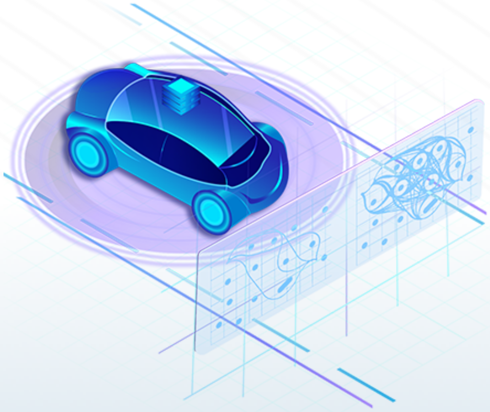


- **3D Mapping:** Create a 3D representation of an environment.
- **Data Visualization (RViz):** Visualize the data that the robot simulation is providing.
- **Autonomous Navigation using RGB-D SLAM:** Perform Autonomous Navigation using the rtabmap_ros.

TEB Local Planner

ABOUT THIS COURSE

Learn how to set up the TEB Local Planner for your Navigation system, including set up for car-like robots



In this small course, you are going to learn how to set up the TEB Local Planner for your ROS Navigation system, including setup for car- like robots, and customization and optimization of the parameters for better performance.

Learning Objectives

- Set up the TEB Local Planner for the Navigation Stack
- Configure and optimize the TEB Local Planner
- Set up the TEB Local Planner for Cars

What You Will Learn

- Setup the Navigation Stack
- Customize Parameters and Optimization
- Set up for car-like robots
- Mini Project to test what you've learned

Intermediate

6 hours

5 lectures

Autonomous Vehicle
Husky Robot,
Carbot

COURSE EXERCISES

- **Set up the gmapping:** Create a map of the environment using the slam_gmapping node.
- **Optimize trajectories:** Obstacle avoidance and customize the planner parameters.
- **Set up TEB Local Planner for a car:** Navigate a car-like robot using ROS and the teb_local_planner.

ROS Perception

ABOUT THIS COURSE

Learn OpenCV, Facial Recognition, Person Tracking, and Object Recognition

Perception is probably one of the most important things when we talk about autonomy. In this course, you will learn how perception is performed by robots using the ROS Framework.

Learning Objectives

- Track objects by their color blobs
- Navigate following floor lines with only an RGB camera
- Detect human faces and track them
- Recognize different faces
- Track a person through a 3D environment
- Recognize flat surfaces, like tables, where the object might be placed
- Recognize objects and track them in 3D space with Point Cloud Sensors

Advanced

7 hours

9 lectures

Mira Robot
Aibo
TurtleBot 2
Fetch robot

COURSE EXERCISES

- **Make the robot follow a red ball:** Use cameras in ROS and use the cmvision package for blob tracking.
- **Use OpenCV in ROS:** Make the robot move in this environment, following the yellow line.
- **Face Detection and tracking in ROS:** Detect where a human face is and make the robot track the movements of humans.
- **Aibo Perception Project:** Practice what you have learned about Basic Perception with ROS.

OpenCV Basics for Robotics

ABOUT THIS COURSE

Learn and apply the library most used in computational vision in robotic projects with ROS

The OpenCV Basics For Robotics course will help you to reconcile, understand, and better apply the synergy between OpenCV and ROS. You will learn from basic concepts to widely used tools in computational vision applied in robotic projects.

Learning Objectives

- Know the basics of computational vision with OpenCV
- Understand how OpenCV integrates with ROS
- Understand the scope of OpenCV in ROS in both virtual and real environments
- Learn how to apply vision algorithms in specific robotic applications

Course Challenge

There is a dangerous person in this city, and many possible suspects are close to your robot! You must detect all the people and highlight the dangerous one.

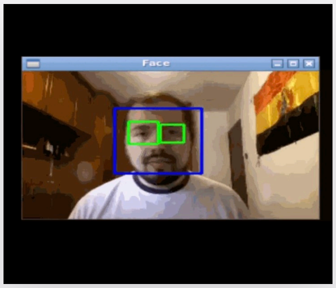
Advanced

19 hours

6 lectures

Hector Quadrotor Drone
ROSbot.

COURSE EXERCISES



- **OpenCV with a Hector Quadrotor:** Apply the basic algorithms of OpenCV using the images taken of a drone.
- **Detecting and Tracking People and Faces:** Use Haar cascade and HOG in order to detect faces and people.
- **Detecting Objects with a Drone:** Detect any kind of object using the ORB algorithm.
- **Start using ArTags:** Use the ArUco library of OpenCV to understand how ArTags work.

Create Your First Robot with ROS

ABOUT THIS COURSE

Creating your first ROS based Robot from Scratch

Have you ever dreamed of creating a robot, a robot that truly belongs to you?

In this course, you are going to make your dream come true. You will go through the whole process of mounting the robot, creating a simulation for the robot so that you can test your ROS programs there, building all the controllers for the robot, and finally, getting it to autonomously navigate using ROS tools.

Intermediate

21 hours

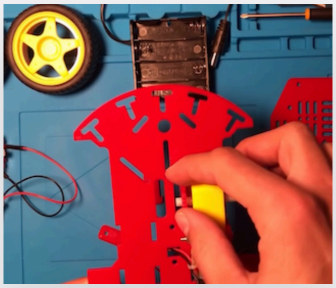
6 lectures

Simulated RIABot
Real RIABot

Learning Objectives

- From idea to real robot
- Mount a real robot
- Create a simulation for the robot
- Control your robot with ROS
- Advanced Utilities: Autonomous Navigation, Deep Learning…

COURSE EXERCISES



- **Build & Mount the Robot:** Mounting your own wheeled robot.
- **Create the Simulation Robot:** Test everything you want in the simulated environment.
- **Autonomous Navigation:** Follow the line and perform SLAM on your robot by using an RGB camera.
- **Apply Deep Learning to the Robot:** Train the robot by using OpenAI_ROS.

Using NVIDIA Jetson Nano with ROS

ABOUT THIS COURSE

Learn Deep Learning using NVIDIA Jetson Nano with IgnisBot

This course will take you from the basics of the NVIDIA Jetbot API, to a Deep Learning-based collision-avoidance system, and ending in a people-following ROS system that uses Deep Learning. And you will learn all that while building your own NVIDIA Deep Learning Robot.

Learning Objectives

- The basics of NVIDIA jetson NANO setup.
- Move a Jetbot based robot
- Train a robot to do obstacle avoidance through deep learning
- Track people and follow them.
- Execute code designed for GPU-CUDA enabled hardware in only CPU systems.
- Build your own IgnisBot, a robot designed for DeepLearning with JetsonNano Hardware.

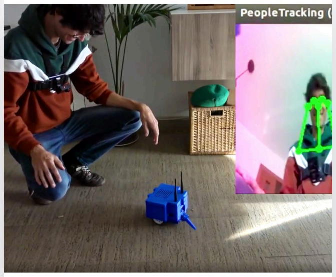
Advanced

23 hours

4 lectures

IgnisBot,
Ignisbot_PRO

COURSE EXERCISES



- **Move the IgnisBot:** Use the Jetbot API to move a two-wheeled robot.
- **Collision Avoidance with Deep Learning:** Train Ignisbot to be able to navigate in a known environment, avoiding obstacles.
- **Create the People Follow ROS Script:** ROSify the people tracker and create a people follower script.
- **Ignisbot Mini Project**
- Combine everything you learned in this project.

URDF for Robot Modeling

ABOUT THIS COURSE

Understanding robot modeling using URDF

As human beings, we learn from a very young age about our body's structure: which bones and muscles are part of it, how they are connected, how we can move each part, etc.

For a robot, though, how can we know all this information? Well, this is exactly what URDF files are going to tell us. URDF files define the structure of a robot, the connection between all the different parts, etc.

In this course, you will understand how URDF files work, and how to create them for any robot.

Learning Objectives

- Build a visual robot model with URDF
- Add physical properties to a URDF Model (Collision, Frictions, etc.)
- Use XACRO to clean up URDF files
- Use URDF in the Gazebo-ROS ecosystem

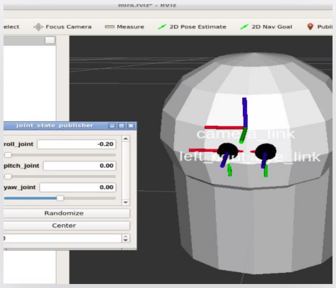
Intermediate

7 hours

6 lectures

Gurdy robot,
Jibo robot
Mira robot

COURSE EXERCISES



roll_joint -0.20
pitch_joint 0.00
yaw_joint 0.00

- **Create your first URDF Model:** Go from a physical robot to a visual virtual model.
- **Adapt URDF for Gazebo Simulator:** Add the collisions, inertias, and gazebo physical properties.
- **URDF + XACRO:** Use XACROs in order to simplify a URDF file.
- **Create your own Jibo:** Create a Jibo robot from scratch and simulate it in Gazebo.

Mastering Mobile Manipulators

ABOUT THIS COURSE

Master how to create ROS applications for autonomous mobile manipulators

Robot manipulators are mobile robots equipped with mobility, one or several robotic arms, and a gripper. They have the ability to autonomously move within an environment, detect objects to grasp, and grasp them to bring them to the proper location. They are widely used in warehouses to locate and bring stuff, in malls and airports to clean, in nuclear areas to access dangerous zones, and even underwater to reach difficult locations. Robot manipulators will be used even more in the near future as their skills improve.

In this course, you will learn how to build a complete ROS application to make a mobile manipulator fill a box with the proper objects obtained from another location.

Learning Objectives

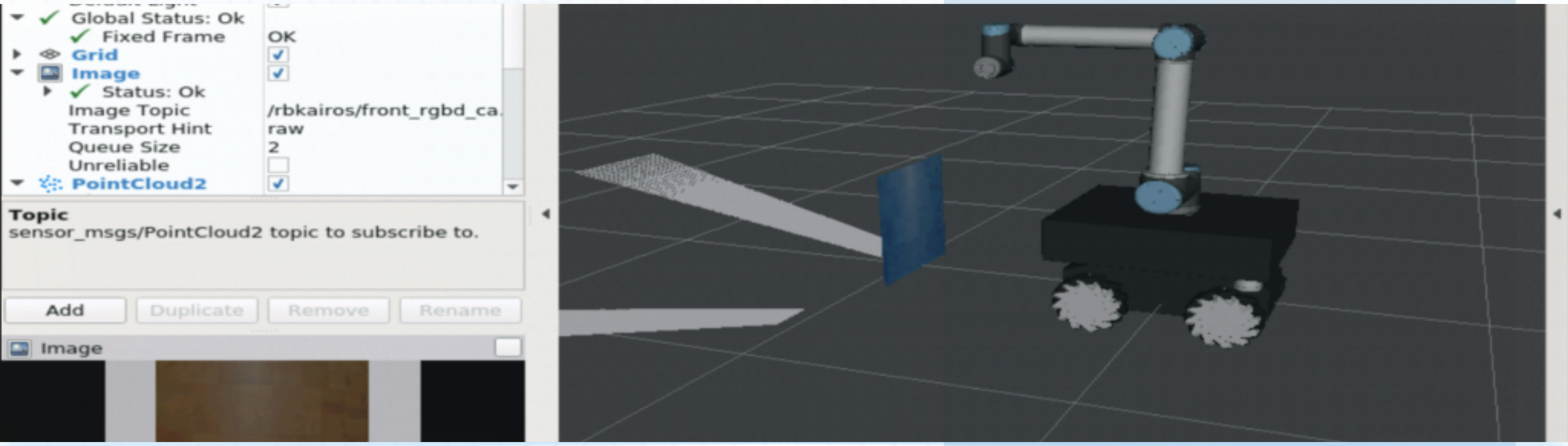
- Do robot navigation in known environments.
- Use perception to detect the objects to grasp in the background.
- Move a robotic arm with a gripper to grasp an object.
- Create a complete application that integrates all these behaviors into a single ROS app based on State Machines.
- Create a web interface that allows people without ROS knowledge to control the robot's operation.

Advanced

53 hours

7 lectures

RB-KAIROS Mobile Manipulator Robot

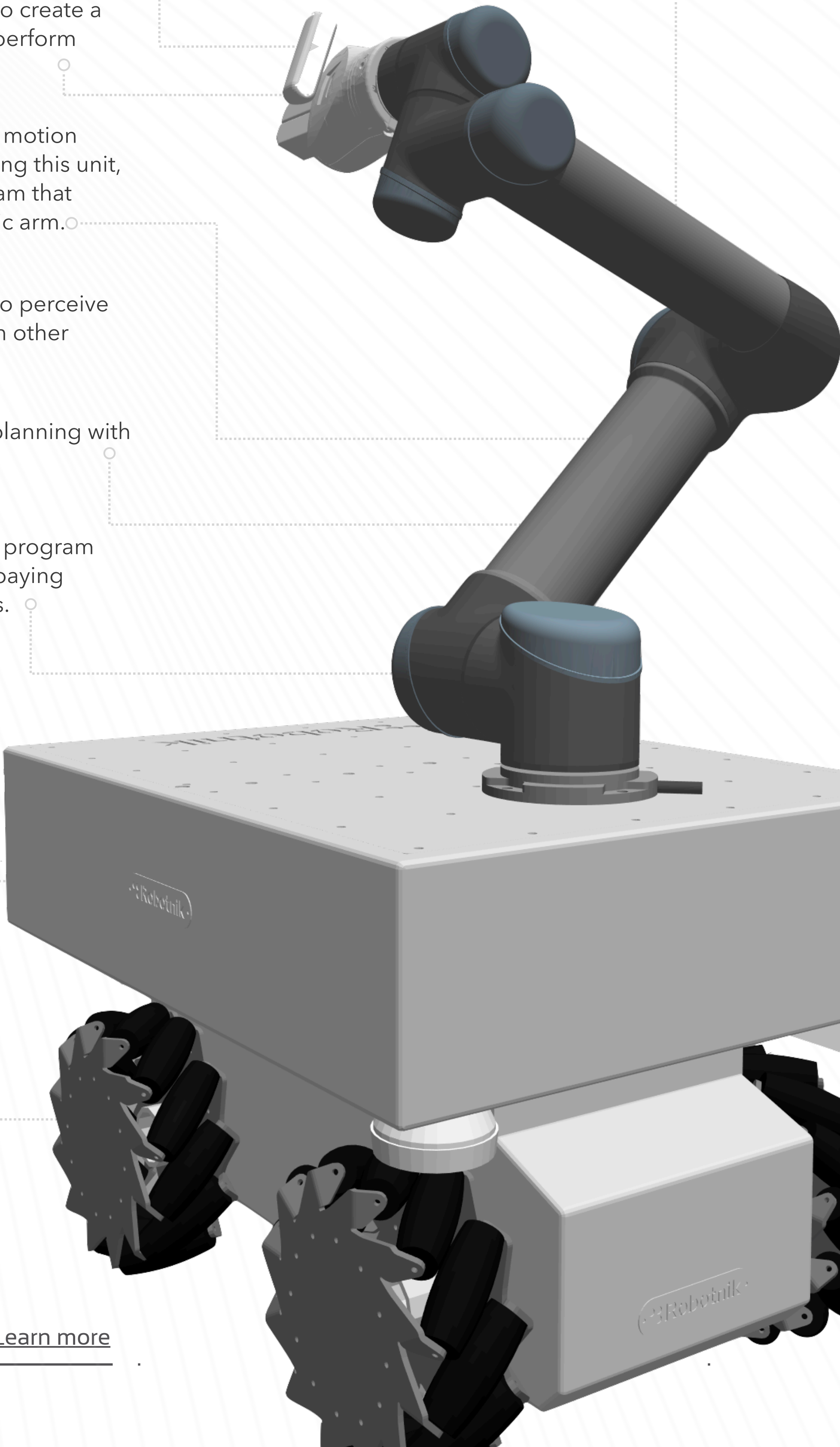


Global Status: Ok
Fixed Frame OK
Grid ☑
Image ☑
Status: Ok
Image Topic /rbkairos/front_rgbd_ca
Transport Hint raw
Queue Size 2
Unreliable □
PointCloud2 ☑

Topic
sensor_msgs/PointCloud2 topic to subscribe to.

COURSE EXERCISES

- **Set Up the Navigation System:** Learn how to set up a Navigation System (using the ROS Navigation Stack) for a Mobile Manipulator robot.
- **Create a MoveIt package:** Learn how to create a MoveIt Package for your mobile manipulator robot. By completing this unit, you will be able to create a package that allows your robotic arm to perform motion planning.
- **Motion Planning:** Learn how to perform motion planning with Python scripts. By completing this unit, you will be able to create a Python program that performs motion planning on your robotic arm.
- **Perception in MoveIt:** Allow the robot to perceive the environment and avoid colliding with other objects.
- **Set Up Manipulation:** Perform motion planning with Python.
- **Set Up Grasping:** Learn how to create a program that performs a full pick and place task, paying special attention to the grasping process.
- **Design Web Interfaces:** Learn how to create a Web Interface for the robot. You will be able to see the navigation map, the robot camera, control the robot through a web-joystick, and start/stop the pick and place algorithm.
- **Program the Robot Behavior:** Learn how to create and manage state machines, using FlexBe, in order to generate different behaviors for your robot.
- **Project using the real robot RB-Kairos:** Test all the programs created during the course into a real robot, the RB-Kairos mobile manipulator.



ROS Manipulation

ABOUT THIS COURSE

Learn how to make your manipulator interact with the environment using ROS

ROS Manipulation is the term used to refer to any robot that manipulates something in its environment.

The main goal of this Course is to teach you the basic tools you need to know in order to be able to understand how ROS Manipulation works, and teach you how to implement it for any manipulator robot.

Learning Objectives

- Learn the basics of ROS Manipulation
- Create and configure a MoveIt! package for a manipulator robot
- Perform Motion Planning
- Perform Grasping

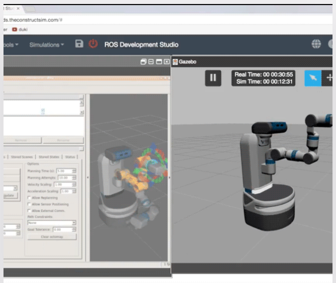
Intermediate

12 hours

7 lectures

Fetch Robot
RB-1 Robot
Universal Robot Arm

COURSE EXERCISES



Real Time: 00 00:30:55
Sim Time: 00 00:12:31

Options
Planning Time (s): 5.00
Planning Attempts: 10.00
Velocity Scaling: 1.00
Acceleration Scaling: 1.00
☐ Allow Replanning
☐ Allow Sensor Positioning
☐ Allow External Comm.
Path Constraints: None
Goal Tolerance: 0.00

- **Motion Planning:** Create a MoveIt Package to make the industrial robot perform motion planning.
- **Moving the RB-KAIROS robot:** Plan and execute trajectories for your robot using MoveIt.
- **Perform motion planning with Python:** Do Motion Planning Programmatically.
- **Robotic Grasping:** Make the robot arm detect and grasp an object.

ROS for Industrial Robots 101

ABOUT THIS COURSE

Introduction to some basic ROS tools to control industrial robots with ROS

ROS for Industrial Robots is a project where the main goal is to bring ROS closer to the industrial robotics world. It is a HUGE project, composed of many packages and tools.

This Course is not meant to make you learn all the things you can achieve with ROS-Industrial, but just to introduce you to some basic concepts you need to know if you want to begin exploring all the ROS-Industrial capabilities.

Learning Objectives

- Overview of how to create a URDF file for an industrial robot.
- Create a MoveIt! package for your industrial robot.
- Perform motion planning using Python.

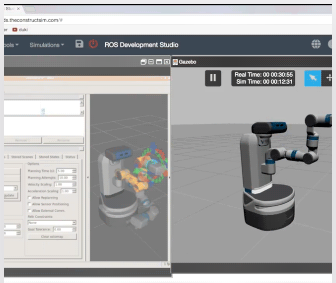
Intermediate

7 hours

5 lectures

Motoman SIA
Universal Robot Arm

COURSE EXERCISES



- **Building the URDF:** Create a Unified Robot Description Format (URDF) file for an industrial robot.
- **Build a MoveIt Package:** Create a Moveit Package for your industrial robot.
- **Motion Planning:** Perform motion planning with Python.

Developing Web Interfaces for ROS

ABOUT THIS HANDS-ON PROJECT

From the essential to advanced widgets, learn how to control and monitor robots with ROS using just your web browser, all on the web!

Would you like to control and monitor robots with ROS using just your web browser?

The "Developing Web Interfaces for ROS" is the right course for you! We dive, step-by-step, into many different widgets that communicate directly to ROS using only HTML and JavaScript code. You will learn the essential and be able to develop more using the knowledge acquired.

Learning Objectives

- Understand how to make ROS data available to other environments
- Understand how to create simple but efficient web pages
- Understand how to publish to topics and control robots from the web
- Understand how to subscribe to topics and monitor ROS data from the web
- Understand how to work with ROS params from the web
- Understand how to consume ROS services and action servers from the web
- Create powerful interfaces that show: 3D models, maps and camera images

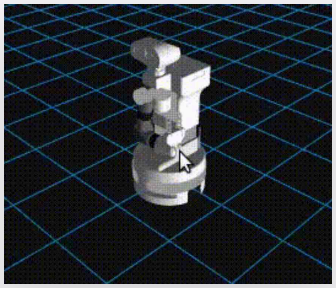
Intermediate

53 hours

12 lectures

Hector Quadrotor,
Fetch
Husky
Turtlebot 2

COURSE EXERCISES



- **Setup the Development Environment:** Web Development, Bootstrap and Javascript.
- **Move the Robot:** Configure your JavaScript code to connect from a Vue.js application.
- **Tracking the Robot:** Subscribe to an odometry topic, which will provide us with real-time data on the robot's moving position.
- **3D Visualization:** Create powerful interfaces that show: 3D models, maps and camera images.

Learn Summit XL and ROS

COURSE OF PRODUCT

Learn all the basics to work with the Summit XL robot

In this course of project, you will **learn all you need to operate and use the Summit XL robot platform in the real world**. You will learn it through hands-on experience with a simulated version of the real robot, which will have the same exact interface as the real one.

Learning Objectives

- Set up the navigation stack to make it navigate in an indoor environment, generating maps on its own.
- Create a program to navigate in outdoor environments through GPS data.
- Detect persons with the Hokuyo laser sensor.
- Detect persons with its PTZ RGB camera.
- Recognize persons with its PTZ RGB camera and tell if it has permission to be there or not.
- Set WayPoints in a map to make it follow that path to patrol.
- Create a reactive program based on all previously mentioned and create a patrolling program that reacts to person detection.

PROJECT TASKS

- **Set Indoor Navigation Stack:**
  - How to move the robot
  - How to set the navigation stack to be able to send him a location and be able to reach that destination.
  - How to send it a sequence of waypoints and execute those movements.
- **Detect and localise person.** Learn how to detect a person using two different systems:
  - Detect through leg detection with laser sensor: This system detects persons based on laser readings of leg patterns. This allows you to have even the position of the person, not only detect if there is a person or not.
  - Detect person through a PTZ RGB normal camera: This system doesn't give you directly the position of the person but detects the persons through Video with OpenCV.
  - As extra, in this unit you will have to learn how to move the PTZ camera around.
- **Set Outdoors Navigation:** Here you will learn how to access the GPS data and use it to navigate around without a mapping system. You will have to rely upon the lasers to detect any obstacles and navigate around them to get to your destination in GPS coordinates. You will also learn how to visualize real maps in RVIZ based on GPS data.
- **Patrole with Summit XL Micro Project:** Apply all that you learned in previous units to create a program that makes the omnidirectional wheels mode Summit XL patrol around a building using indoor navigation when inside a building and GPS data when outdoors.

Learn Smart Grasping System and ROS

COURSE OF PRODUCT

Learn to work with the Smart Grasping System

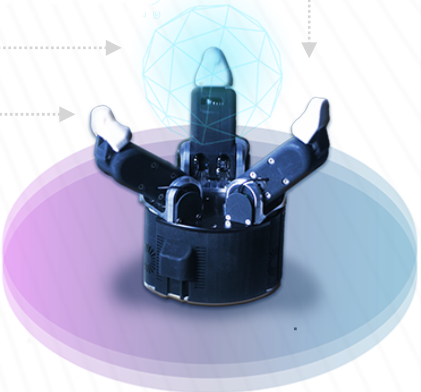
Within this course of product, you are going to learn how you can start working with one of the robotic hands developed by the Shadow Robot Company, as well as how to use their Smart Grasping System.

Learning Objectives

- Basic Usage and control of a Shadow Hand.
- How to attach a robotic hand to a manipulated arm.
- How to create a MoveIt package for a manipulated robot.
- How to interact with the Smart Grasping System
- How to integrate Perception with the Smart Grasping System

PROJECT TASKS

- **Attach the hand to a robotic arm:** See how to attach the Shadow Hand to an industrial robot (UR5). You will go through some basic knowledge regarding URDF files and how to modify them.
- **Motion Planning with MoveIt:** Learn how to create a Moveit package for your industrial robot. By completing this Unit, you will be able to create a package that allows your robot to perform motion planning.
- **Smart Grasping System:** Learn how to interact and what is the purpose of the different functions integrated in the Smart Grasping System.
- **Perception and Object Recognition:** This Unit will show you how to use Perception and Object Recognition to get the position of graspable objects.
- **Shadow Hand Micro Project:** Apply all that you learned in previous units to read the position of the object, publish the position into a topic and create a service for reading this data. Finally you will need to integrate your code with the Smart Grasping System and test and correct errors.



Learn TurtleBot 3 and ROS

COURSE OF PRODUCT

Learn how to work with a TurtleBot3 robot

Within this Course, you are going to learn how you can start working with a TurtleBot3 robot using its both versions, Burger and Waffle.

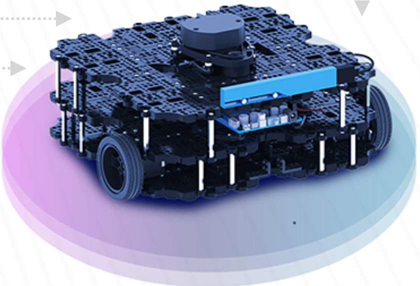
Learning Objectives

- Basic Usage and control of the TurtleBot 3 robot.
- How to perform Navigation with TurtleBot 3.
- Follow a line with TurtleBot 3.
- Object Recognition with TurtleBot 3
- Motion Planning in Moveit with TurtleBot 3

PROJECT TASKS

- **Navigation with TurtleBot 3:** The first thing you will need for a robot that patrols is the ability to move around without crashing into everything, for this you will learn:
  - How to create a map of an environment
  - Localize the robot within the map
  - Path Planning with Obstacle Avoidance
  - Send a sequence of waypoints and execute those movements
- **Follow a line:** Learn how to start using the most basic and also the most powerful tool for perception in ROS: OpenCV.
- **Blob Tracking:** You will start using cameras in ROS and use the cmvision package for blob tracking.
- **Perception and Object Recognition:** Learn how to use Perception and Object Recognition to get the position of graspable objects.

- **Motion Planning with MoveIt:** Learn how to create a Moveit package for your manipulator robot. By completing this unit, you will be able to create a package that allows your robot to perform motion planning.
- **TurtleBot3 Micro Project:** To practice some of the knowledge that you've gained during this course. You will do this by doing a project, which will be based on 2 parts:
  - The first part will consist of applying the Navigation concepts you learned during the Course
  - The second part will consist on perform some Motion Planning using code, instead of MoveIt.



Learn TIAGO and ROS

COURSE OF PRODUCT

Learn how to work with a TIAGo robot from PAL Robotics

TIAGo is a service robot designed to work in indoor environments. TIAGo’s features make it the ideal platform for research, especially on ambient assisted living or light industry. It combines mobility, perception, manipulation, and human-robot interaction capabilities for one specific goal: to be able to assist in research.

In this course, you are going to learn how you can start working with the TIAGo robot, explore its functionalities, and how to build some interesting ROS applications for it.

Learning Objectives

- Control of TIAGos joints.
- Navigation with TIAGo.
- Motion Planning with MoveIt!
- Perception with OpenCV
- Perception with PCL

PROJECT TASKS

- **Autonomous Navigation with TIAGo:** The first thing you will need for a robot that patrols are the ability to move around without crashing into everything, for this you will learn:
  - How to create a map of an environment
  - Localize the robot within the map
  - Path Planning with Obstacle Avoidance
  - Send a sequence of waypoints and execute those movements
- **Motion Planning with MoveIt #1:** You are going to check the different way of performing Motion Planning with TIAGo using MoveIt:
  - Planning in joint space
  - Planning in cartesian space
- **Motion Planning with MoveIt #2:** Use Octomap in MoveIt to compute the collision checking with the environment around the robot during the motion planning of poses given in cartesian space.
- **Motion Planning with MoveIt #3:** Tabletop pick & place demo using monocular model-based object reconstruction based on ArUco markers and the pick and place pipeline in MoveIt!
- **Perception with OpenCV:** Perform various Perception tasks using the OpenCV library.
- **Perception with PCL:** Perform various Perception tasks using the PCL library.
- **TIAGo Micro Project:** You will have to apply all you have learned during the course in order to create a program that makes TIAGo patrol around an environment.

Learn Jackal and ROS

COURSE OF PRODUCT

Learn how to create real world applications for a Jackal robot from ClearPath Robotics

In this course you will learn all you need to operate and use the Jackal robot platform in the real world. You will learn it through hands on experience with a simulated version of the real robot which will have the same exact interface as the real one, thanks to ROS infrastructure.

Learning Objectives

- Set up the navigation stack to make it navigate in an indoor environment, generating maps by its own.
- Create a program to navigate in outdoors environments through GPS data.
- Detect persons with the laser sensor.
- Detect person with its RGB stereo camera.
- Set WayPoints in a map to make it follow that path to patrole.
- Use the StereoCam to generate PointCloud Data
- Create a reactive programs based on all previously mentioned and create a patroling program that reacts to person detections.

PROJECT TASKS

- **Navigation Indoor:** The first thing you will need for a robot that patrols are the ability to move around without crashing into everything, for this you will learn:
  - How to move the robot
  - How to navigate without a map.
  - How to map the environment
  - How to set the navigation stack to be able to send him a location and be able to reach that destination.
  - How to send it a sequence of waypoints and execute those movements.
- **Set Outdoors Navigation:** Learn how to access the GPS data and use it to navigate around without a mapping system. You will have to rely upon the lasers to detect any obstacles and navigate around them to get to your destination in GPS coordinates. You will also learn how to visualize real maps in RVIZ based on GPS data.
- **Detect and Localize a Person:** Detect a person using two different systems:
  - Detect through leg detection with laser sensor: This system detect persons based on laser readings of leg patterns. This allows you to have even the position of the person, not only detect if there is a person or not.
  - Detect person through an RGB camera from the stereo-cam: This system doesn't give you directly the position of the person, but detects the persons through Video with OpenCV.
  - As extra, you are going to learn how to Get Point cloud from the stereo camera: Use the stereo camera to extract point cloud data from the scene.
- **Jackal Micro Project:** Create a program that makes Jackal patrol around using indoor navigation when inside a building and GPS data when it is outdoors.