# ROBOTICS SOFTWARE ENGINEER NANODEGREE SYLLABUS

## Course 1: Introduction to Robotics

In this course, you'll get an introduction to your Nanodegree program, and explore the three essential elements of robotics: perception, decision making, and actuation. You'll also build your first project, which is modeled after the NASA Mars Rover Challenge.

| Lesson | Title | Description |
|---|---|---|
| 1 | Welcome | In this first lesson,  you'll meet your instructors, learn about the structure of this program and about the services available to you as a student. |
| 2 | What is a Robot? | Ask three people what a robot is and you'll get three different answers!  Here we ask your instructors, three expert roboticists from Electric Movement. |
| 3 | Search and Sample Return | In this lesson, you'll learn the skills you need to tackle the first project, where you'll experience the three essential elements of robotics -- perception, decision making, and actuation. |

| Project 1 : Search and Sample Return |
|---|
| In this project, you will write code to autonomously map a simulated environment and search for samples of interest. |

| | | |
|---|---|---|
| 4 | Careers: Orientation | As you learn the skills you'll need in order to work in the robotics industry, you'll see optional Career Lessons and Projects that will help you prepare for interviews, craft your resumé and more. |

## Course 2: ROS Essentials

ROS provides a flexible and unified software environment for developing robots in a modular and reusable manner. In this course, you'll learn how to manage existing ROS packages within a project, and how to write ROS Nodes of your own in Python.

| Lesson | Title | Description |
|---|---|---|

| | | |
|---|---|---|
| **1** | **Introduction to  ROS** | Obtain an architectural overview of the Robot Operating System Framework and set up your own ROS environment on your computer. |
| **2** | **Packages & Catkin Workspaces** | Learn about ROS workspace structure, essential command line utilities, and how to manage software packages within a project. Harnessing these will be key to building shippable software using ROS. |
| **3** | **Write ROS Nodes** | ROS Nodes are a key abstraction that allows a robot system to be built modularly. In this lesson,  you'll learn how to write them using Python. |

## Course 3: Kinematics

Movement is one of the most exciting elements of building a Robot that interacts with the physical world. In this course, you'll learn to control a robotic arm with six degrees of freedom to perform pick and place actions using Inverse Kinematics.

| **Lesson** | **Title** | **Description** |
|---|---|---|
| **1** | **Intro to Kinematics** | In this lesson you'll get an introduction to the exciting field of kinematics and learn about the most important types of serial manipulators used in the robotics industry. |
| **2** | **Forward and Inverse Kinematics** | Here you'll dive deep into the details of solving the forward and inverse kinematics problem for serial manipulators. |
| **3** | **Robotic Arm: Pick & Place** | In this lesson you will learn how to control a robotic arm with six degrees of freedom to perform pick and place actions using Inverse Kinematics. |

| **Project 2 : Robotic Arm: Pick & Place** |
|---|
| In this project, you will write code to perform Inverse Kinematics. Given a list of end-effector poses, you will calculate joint angles for the Kuka KR210. |

## Course 4: Perception

Robots perceive the world around them by using sensors. Working with sensor data for perception is a core element of robotics. Here you'll work with 3D point cloud data to perform segmentation tasks using techniques like RANSAC and clustering.

| Lesson | Title | Description |
|---|---|---|
| 1 | **Perception Overview** | Here's a quick look at what to expect in the upcoming lessons and project. |
| 2 | **Introduction to 3D Perception** | Dive into the world of perception in three dimensions! After a brief tour of 3D sensors used in robotics we'll explore the capabilities of RGB-D cameras, which you'll use in these lessons. |
| 3 | **Calibration, Filtering, and Segmentation** | To understand your sensor data you first need to calibrate! Here you'll get a handle on RGB-D camera calibration and how to do filtering and basic segmentation on your point cloud data. |
| 4 | **Clustering for Segmentation** | Clustering is a powerful machine learning method for segmenting objects of any arbitrary shape in your point cloud data. Here you'll compare K-means and Euclidean clustering for object segmentation. |
| 5 | **Object Recognition** | In this lesson, you'll take your segmented point cloud and isolate features you can use to train a machine learning algorithm to recognize the object you're looking for! |
| 6 | **3D Perception Project** | In the project at the end of this lesson, you'll bring together everything you know about perception in three dimensions, from filtering and segmentation to feature extraction and object recognition! |

| **Project 3: 3D Perception** |
|---|
| In this project, you will complete a tabletop pick and place operation using PR2 in simulation. The PR2 is a common hardware and software platform for robot researchers. This one has been outfitted with a noisy RGB-D sensor that your robot must use to identify and acquire objects from a cluttered space. |

## Course 5: Controls

Control systems are a central component to most robots. In this course, you'll learn how a mechanical system can be described in terms of the equations that govern it. You'll then learn how to manage the behavior of the system using a controller. Lastly, you'll have an opportunity to observe your controller in simulation.

| Lesson | Title | Description |
|---|---|---|
| 1 | **Introduction to Controls** | In this lesson you'll learn about Controls and how to create and tune PID controllers. |
| 2 | **Quadrotor Control using PID** | In this lesson, you'll learn how to control a Quadrotor inside a Unity environment using a PID based Positional Controller within a ROS node. |

## Course 6: Deep Learning

Many recent developments in robotics can be attributed to advances in Deep Learning.
In this course, you'll learn about Convolutional Neural Networks (CNN), Fully Convolutional Networks (FCN), and Semantic Segmentation. You will then integrate a Deep Neural Network with a simulated quadcopter drone.

| Lesson | Title | Description |
|---|---|---|
| 1 | **Intro to Neural Networks** | In this lesson, Luis Serrano provides you with a solid foundation for understanding how you build powerful neural networks from the ground up. |
| 2 | **TensorFlow for Deep Learning** | Vincent Vanhoucke, Principal Scientist at Google Brain, introduces you to deep learning and TensorFlow, Google's deep learning framework. |
| 3 | **Deep Neural Networks** | Vincent walks you through how to go from a simple neural network to a deep neural network. You'll learn about why additional layers can help and how to prevent overfitting. |
| 4 | **Convolutional Neural Networks** | Vincent explains the theory behind Convolutional Neural Networks and shows you how to dramatically improve performance in image classification. |
| 5 | **Fully Convolutional Networks** | In this lesson, you'll learn the motivation for Fully Convolutional Networks and how they are structured. |
| 6 | **Semantic Segmentation** | In this lesson you'll be introduced to the problem of Scene Understanding and the role FCNs play. |
| 7 | **Project: Follow Me** | How to setup your environment and collect data for the |

| | | |
|---|---|---|
| | | Follow Me project. |

| **Project 4: Follow Me** |
|---|
| In this project, you will build and train a Fully Convolutional Network (FCN) to find a specific sim-person in images using semantic segmentation. Your simulated quadcopter will then run your trained model with an inference engine in real time, to find the sim-person in video as the quadcopter patrols, and follows. |

| | | |
|---|---|---|
| **8** | **Term 1 Outro** | Wrapping up your first term! |

## Course 7: Introduction to Term 2

In this course, you'll get an introduction to Term 2, and explore hardware commonly used in robotics. You'll learn the uses of common sensors, and which ROS packages you need to support them. Those students with TX2 hardware will learn how to set up the system and interact with external hardware.

| Lesson | Title | Description |
|---|---|---|
| 1 | **Introduction to Term 2** | Term 2 brings a new direction to the program. Robotics applications today are adding machine learning techniques to the traditional robotics portfolio. You'll learn the latest reinforcement learning techniques as well as how they can be deployed on TX2 hardware platforms. |
| 2 | **Introduction to the TX2** | Students will also get a brief introduction to the Jetson TX2 and how to setup the system. |
| 3 | **Interacting with Robotics Hardware** | Introduce students to simple hardware I/O connections, communication, and simple electrical theory. |
| 4 | **Lab: Hardware Hello World** | Everyone who learns programming starts with a basic Hello World program. In hardware, the Hello World version is to blink an LED. Learn how to do that with the TX2. |
| 5 | **Introduction to Robotics Sensor Options** | Brief introduction to various common robotics sensors, their I/O, and their purpose. |

## Course 8: Robotic Systems Deployment

In this course, you'll learn new tools, and the embedded workflow as you move from code on a host system to code on a target system. You'll work through a familiar problem with these new tools, then extend what you've learned in a project.

| Lesson | Title | Description |
|---|---|---|
| 1 | **TX2 Development** | Meet Kelvin Lwin of Nvidia. Following Nvidia's "Two Days to a Demo", tutorial, learn about the tools and workflow for developing and deploying robotics software to the TX2. |
| 2 | **Inference Applications in Robotics** | Learn about the many and varied applications for inference engines in robotics and the real-time considerations for these systems. |
| 3 | **Project: Robotic Inference** | Learn the steps to set up data and tune a deep neural network with DIGITS. Then apply this to your own version of the Robotic Inference project. |

| **Project 5: Robotic Inference** |
|---|
| Design your own robotic system using inference. You will create a project idea, collect your own data set for classification, and justify network design choices based on your analysis of accuracy and speed on the target system. |

## Course 9: Localization

Learn how Gaussian filters can be used to estimate noisy sensor readings, and how to estimate a robot's pose relative to a known map of the environment with Monte Carlo Localization (MCL).

| Lesson | Title | Description |
|---|---|---|
| 1 | **Introduction to Localization** | Learn what it means to localize and the challenges behind it. |
| 2 | **Kalman Filters** | Learn what a Kalman filter is, and its importance in estimating noisy data. |
| 3 | **Lab: Kalman Filters** | Implement an Extended Kalman Filter package with ROS to estimate the position of a robot. |
| 4 | **Monte Carlo Localization** | Introduction to the MCL (Monte Carlo Localization) algorithm to localize robots. |

| | | |
|---|---|---|
| **5** | **Build MCL in C++** | Learn how to code the MCL algorithm in C++ |
| **6** | **Project: Where am I?** | Set up and explore the steps for the Where am I? Project using MCL with ROS in C++. |

| **Project 6: Where am I?** |
|---|
| You will use the Monte Carlo Localization algorithm in ROS in conjunction with sensor data and a map of the world to estimate a mobile robot's position and orientation so that your robot can answer the question "Where am I?" |

## Course 10: SLAM

Learn how to create a Simultaneous Localization and Mapping (SLAM) implementation with ROS packages and C++. You'll achieve this by combining mapping algorithms with what you learned in the localization lessons.

| **Lesson** | **Title** | **Description** |
|---|---|---|
| **1** | **Mapping Algorithms** | Learn about probabilistic occupancy grid mapping. |
| **2** | **Combining Localization and Mapping for SLAM** | The intuition and conceptual background of Simultaneous Localization and Mapping (SLAM). |
| **3** | **SLAM and ROS** | Learn about SLAM packages available in ROS and how to use them. |
| **4** | **Project: Map My World Robot** | Set up and explore the steps you need to do the project with SLAM and ROS in C++. |

| **Project 7: Map My World Robot** |
|---|
| Simultaneous Localization and Mapping (SLAM) can be implemented a number of ways in robotics depending on the sensors used via various ROS packages that exist. Here, you will use a ROS SLAM package and simulated sensor data to create an agent that can both map the world around it and localize within it. |

## Course 11: Reinforcement Learning for Robotics

Begin by learning how to build a basic end-to-end reinforcement learning agent, termed a deep Q-network (DQN). Then, enhance it to create a more complex agent that can pick and place from visual input.

| **Lesson** | **Title** | **Description** |
|---|---|---|

| | | |
|---|---|---|
| 1 | **RL Basics** | Nvidia presents the basics of RL with DQN agents and the OpenAI Gym to train an agent to play a simple game. |
| 2 | **RL Manipulation** | Building on the same RL engine, adapt it to solve robotics arm manipulation problems in a Gazebo simulation environment. |
| 3 | **Project: RL Pick and Place Robot** | Set up a pick and place project using the RL engine. Compare this more general method of learning to manipulate a robotic arm with more traditional methods. |

| **Project 8: RL Pick and Place Robot** |
|---|
| Build an RL agent to pick, grip, stack, and pack, using a manipulator arm. |

## Course 12: Path Planning and Navigation

Extend your RL Engine with more advanced techniques for navigation, and compare these with classic approaches. Finally, combine SLAM and navigation into a single comprehensive project!

| **Lesson** | **Title** | **Description** |
|---|---|---|
| 1 | **Classic Path Planning** | Introduction to classic 2D and 3D path planning and the ROS modules that implement them. |
| 2 | **Lab: Path Planning** | Path planning with classic algorithms. |
| 3 | **Navigation with RL** | Extending the RL engine to navigation. |
| 4 | **Comparisons of Classic vs Deep Learning Approaches in Robotics** | In this changing field, it's important to understand the advantages and disadvantages of different approaches to robotics problems. Sometimes the best solution is a combination of solutions. |
| 5 | **Project: Home Service Robot** | Set up a project that combines advanced RL navigation with SLAM |

| **Project 9: Home Service Robot** |
|---|
| You've already used probabilistic methods with SLAM to map and localize a robot. You've also designed deep RL engines to solve end-to-end sense-to-action problems, which can now be applied to navigation. In this project, you'll combine both AI paradigms to build a home service robot that can map, localize, and navigate to perform household tasks, moving from one room to another autonomously. |