Research Track II – First assignment

Carmine Tommaso Recchiuto, Phd

Package Updated

The pioneer_ctrl package has been updated with a new scene, which can be integrated with the turtlesim_controller package developed during the previous weeks:

CarmineD8/pioneer_ctrl: Control package for the pioneer p3dx in VRep (github.com)

First assignment

Download the starting package for the assignment, by cloning the repository CarmineD8/rt2_assignment1: Package for the first assignment of the Research Track 2 course (github.com) and building the workspace.

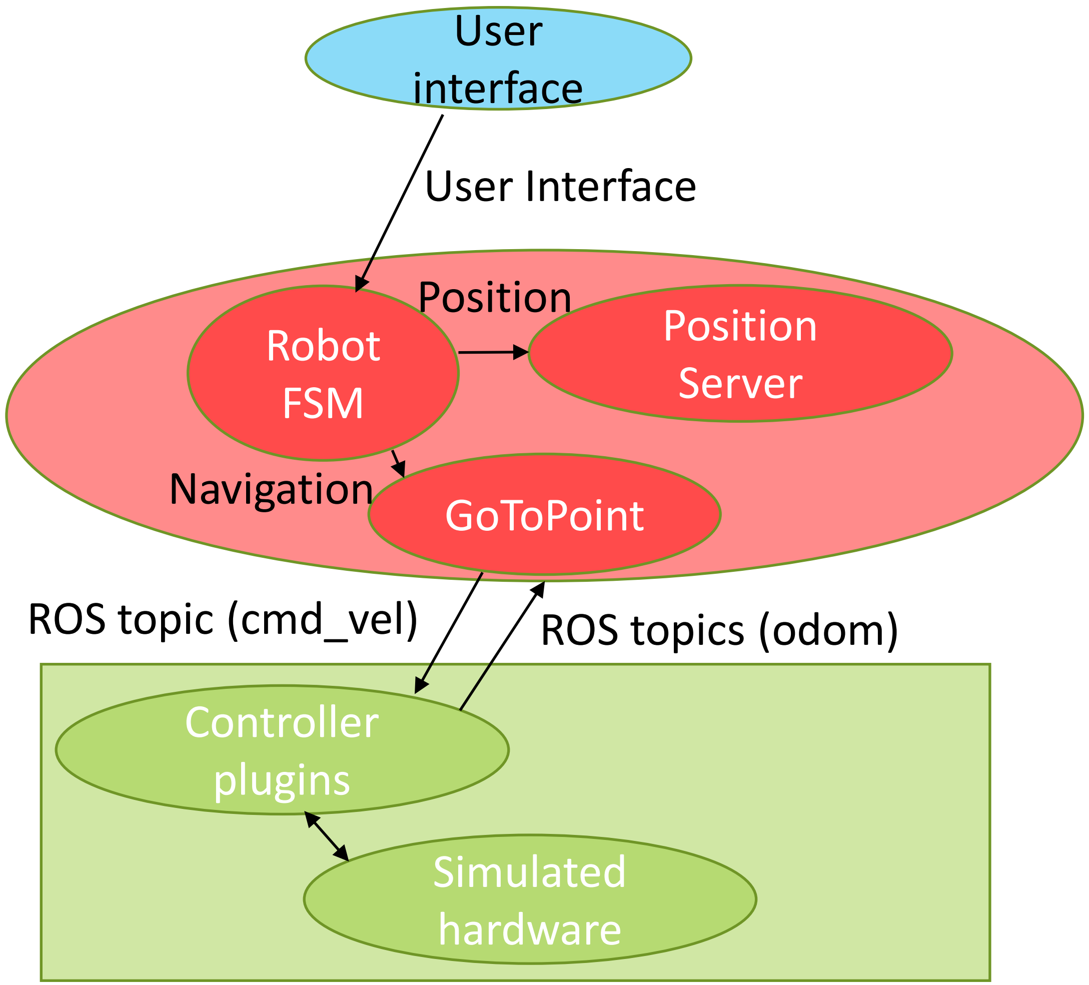
You can then run:

roslaunch rt2_assignment1 sim.launch

The launch file will open:

- a Gazebo simulation with a simple mobile robot
- four nodes which implement the robot's behaviour

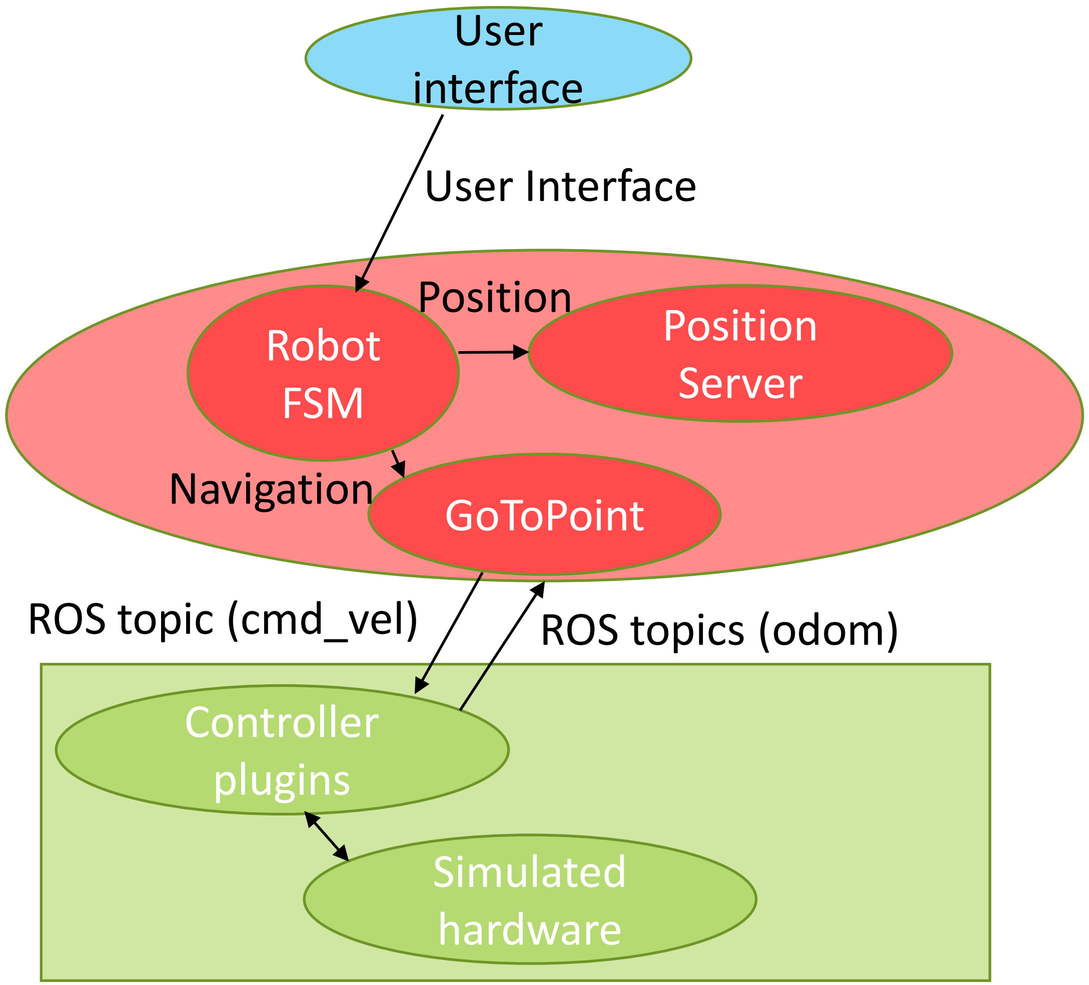
First assignment



The launch file will open:

- the simulation environment
- the node PositionServer, which implements a random position service
- the node GoToPoint, which implements a service to drive a robot toward a point in the environment
- the node FSM, which implements a service to start or stop the robot, and calls the other two services to drive the robot
- the UserInterface, which asks the user to start/stop the robot, and calls the service implemented in the FSM node

First assignment

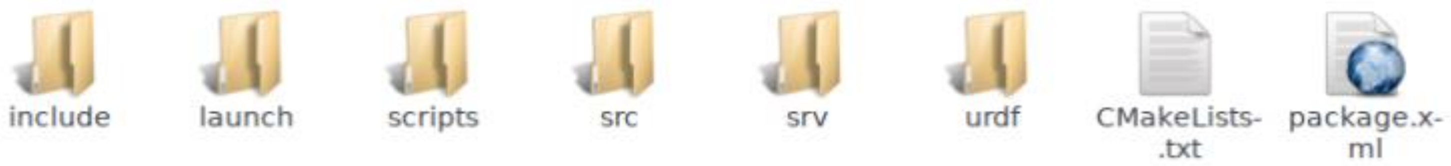


- The service implemented in the GoToPoint node is able to drive the robot towards a certain position in space (x, y) and with a certain angle (theta)

- The service implemented in the Position Server node replies with random values for x, y, and theta, where x and y should be limited between some min and max values

- The service implemented in the Robot FSM node gives the possibility to start or stop the robot behaviour. Please notice that the robot can be stopped only when it reaches a target

Expected Behaviour



- The folder contains:
  - the launch folder, with the launch file
  - the scripts folder, with two nodes (go_to_point.py and user_interface.py)
  - the src folder, with the source codes of the nodes FSM and position_server
  - the srv folder, with the custom messages and services implemented
  - the urdf folder, with the description of the robot

Requirements

The package should be modified by creating your own repository, containing the given package and two additional branches:

- ✓ *action*
- ✓ *ros2*

Moreover, you are required to create a VREP scene which can interact with the simulation.

Requirements

1) The Branch *action* should contain the same package in ROS, but with the go_to_point node modelled as a ROS action server, instead of a “simple” server.

   Given that, the robot FSM node should now implement mechanisms for possibly cancelling the goal, when the related user command is received

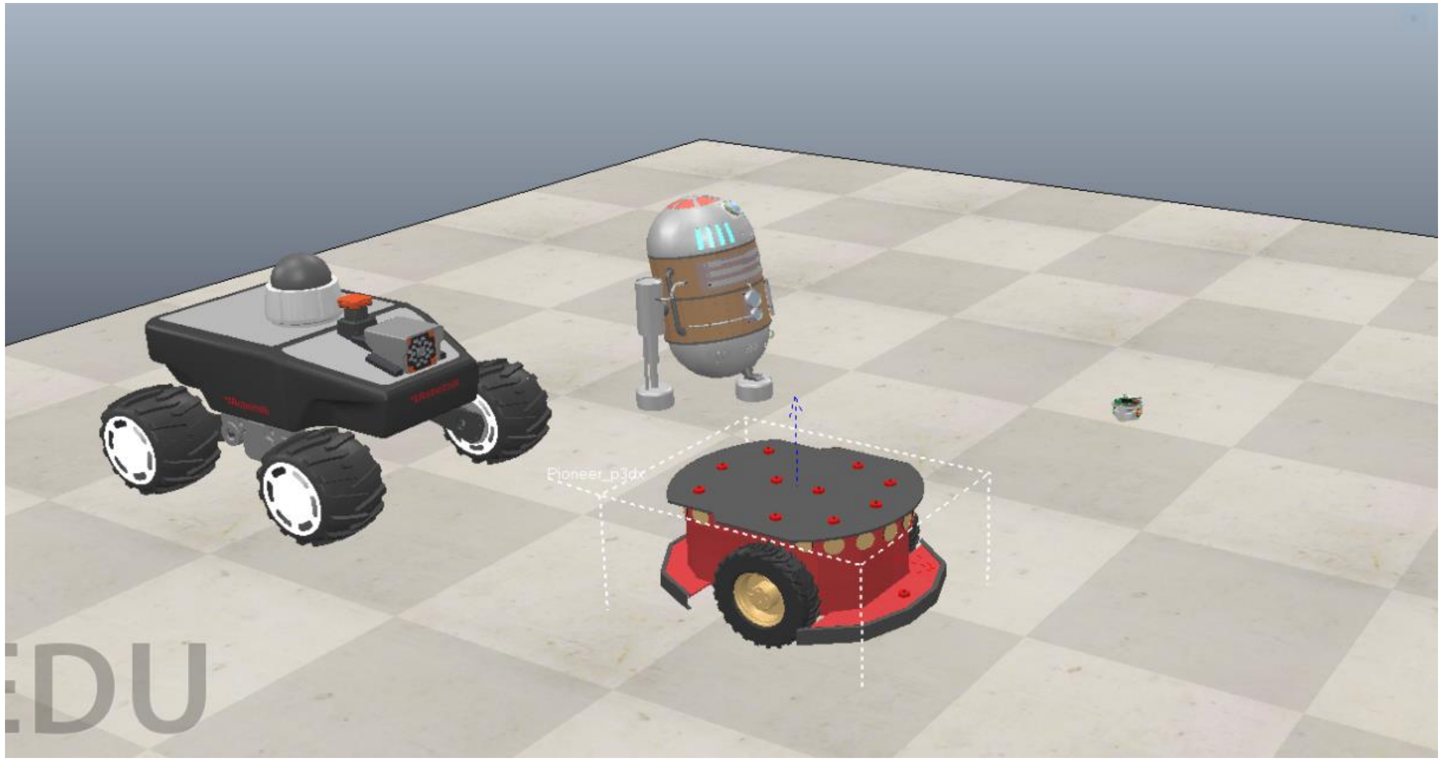
2) In the branch *ros2,* the cpp nodes (Robot FSM and position server) should be written for ROS2, as components, so that, by using the ros1_bridge, they can be interfaces with the ROS nodes and with the simulation in Gazebo. The go_to_point can still be implemented as a service

Also:

- a launch file to start the container manager and the components should be created
- a script to launch all required nodes and the simulation should be implemented

Requirements

3) Finally, in the main branch (or in one of the two added branches), please add a Vrep scene containing a robot interacting with the simulation.
You can choose if using ROS or ROS2 api.



Requirements – how to submit the assignment

- The link of a github repository containing the packages with the main branch and the two additional branches should be provided.

- The repo should have a README.md (only required for the two additional branches) file with:

  - The list of the required packages, if any
  - The description of the system,
  - Instructions about how to run the code.

- Functions and source files should be documented (optional: you can create a docs folder with DoxyGen documentation).

Schedule upcoming Classes and Assignment

22nd April: 1st assignment given (deadline 20th May)

13th May 14th May	Data Visualization in Python
20th May 21st May	Jupyter Notebooks for Robotics
27th May 28th May	Software Documentation + last assignment

Deadline

The deadline (hard) is set to the 20th May.

- In the next two weeks, you can contact me even outside class hours for clarifications about the assignment and the course subjects in general.