

DESCRIPTION

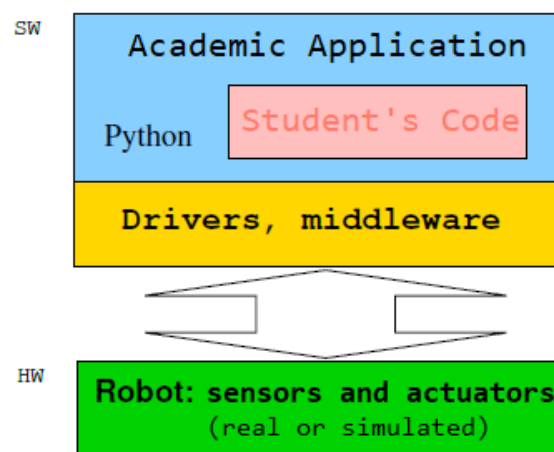
INTRODUCTION

The JdeRobot-Academy environment for learning robotics is presented. It is based on the ROS middleware and Gazebo, and uses Python as a programming language for the robot logic. It is open source and multiplatform.

JdeRobot-Academy aims to bring the field of robotics closer to the university student, in a practical way and emphasizing in the programming of the robot needed to address a specific robotic problem.

PHILOSOPHY

The environment consists of a set of independent and attractive practices, according to applications that are recently reaching society (autonomous cars, mobile robots such as vacuum cleaners, drones...) where the use of standard libraries in robotics such as OpenCV or PCL is encouraged. The practices do not focus on a specific robot, but cover different aspects of robotics with simulated versions and in some cases with real hardware including Turtlebot, autonomous cars and humanoids. Each exercise has three layers:



The lower layer is the robot that wants to perform some task in a certain environment, simulated or real. The intermediate layer are the respective drivers that give software access to the sensors and actuators of the robot. The upper layer is the academic node that analyzes the sensory data and makes decisions of action, planning if necessary.

The academic node is connected to the real robot or simulated interchangeably, without the students having to make changes in their code, due to the framework use of configuration files where the simulation features or parameters of use of a real robot are specified.

INFRASTRUCTURE

For each practice an academic node has been created, which contains a specific part already programmed that solves auxiliary tasks, and that also includes a single template file for the students to enter their code there. This academic node solves the graphic interface of the practice and the template for the student's code, also provides a programming interface to access that graphical interface debugging tools (GUI API) and another programming interface for accessing the sensors and actuators of the robot (HAL API) in the form of simple Python methods. In addition, it has a temporary skeleton

based on a continuous loop of iterations in which sensory data are collected, then processed, then the way to act is decided and finally orders are sent to the actuators.

GOALS

With the presentation of this environment, it is intended that attendees can discover its utilities and usage possibilities for incorporating it to their work. Given its teaching purpose, the aim is –as well– to convey to listeners the educational idea on which JdeRobot-Academy is based, in order to enrich the robotic teaching with practical material. In addition, the intention also is to demonstrate the ease of integration of ROS middleware in robotic projects, emphasizing its power and usability. There will be a general presentation of the environment, going into some implementation details, enriched by specifying and describing some of the practices which the environment has, such as: *car junction*, *global navigation*, *follow line* and *vacuum cleaner with SLAM*.

PRACTICES

As an example we present the *follow line* practice, where the objective is that a Turtle-Bot follows the red line in a circuit in the shortest time possible, by extracting information from the pixels of a camera on board the robot and ordering its motors the proper movement. This practice has both real hardware and simulated versions. The student has to use color filters and ad-hoc processes that extract some measures including whether there is a line in the image. A typical solution is to process four or five lines of each frame and filter by color only in those lines, measuring the deviation from the situation in which the robot is centered on the line. This practice illustrates very well the usefulness of a PID control. Its graphical interface facilitates the visualization of processing done on the image.

Another existing practice is *vacuum cleaner with SLAM*, whose objective is to implement a navigation algorithm for an autonomous vacuum with autolocation to cover the largest area of a known house, which a map has been provided from. The solution of the exercise begins with the planning of the route, creating a grid on the map to carry out a zigzag path by checking through the value of each pixel if the adjoining cells belong to an obstacle, if it has already gone through them, or if they are areas that are still to be cleaned. The graphical interface will show the map of the house, and over it the path followed by the robot. This practice has an automatic evaluator that connects with the simulator to establish an objective note according to the performance of the algorithm.

BALANCE

The environment presented is free software, downloadable from GitHub¹. It is open to collaborations, extensions, modifications and its use is free in its entirety or in the parts that are needed. It has been used successfully in some degree subjects at Spanish universities (such as the URJC) and in postgraduate courses, as well as in several editions of the PROGRAM-A-ROBOT CHALLENGE² (IROS³) competition. You can also find demonstration videos of the execution of all practices in YouTube. Videos for the four practices mentioned above can be found in [4], [5], [6] and [7].

¹ <https://github.com/JdeRobot/Academy>

² <https://jderobot.org/Program-A-Robot-2018>

³ <https://www.iros2018.org>

⁴ <https://www.youtube.com/watch?v=nSYJ0KDLBvg>

⁵ <https://www.youtube.com/watch?v=uaufpVLdRA>

⁶ <https://www.youtube.com/watch?v=QGO9oaoBVoA>

⁷ https://www.youtube.com/watch?v=sUT5ru4Ew_E

INSTALLATION

The installation of the environment is simple, having prepared binary packages for Ubuntu and Debian that include the drivers, the simulator and the academic applications of each of the practices. This package installs the *ros-kinetic-desktop* package.

FUTURE LINES

JdeRobot-Academy currently uses ROS-Kinetic and Gazebo 7, but it is intended to adapt the infrastructure to work with ROS-Melodic and Gazebo 9.0.0 in the near future. The use of MAVROS nodes for drones is also being studied and incorporated into the project. In addition, the environment already has a functional first web version, which is still in the process of development and enrichment.