Documentation of Projekt II.

Kirill Rassudikhin
email: kpassudin@gmail.com
Supervisor: Ing. Jaroslav Bušek, Ph.D.

October 2021

1 Assignment:

Implementation of navigation of the wheel robot in virtual environment:

- Make work the simulation of TurtleBot3 with ROS + Gazebo or Rviz
- Create detailed documentation of the development
- Create and implement algorithm for robot to travel from point A to point B

2 Introduction:

What is ROS:

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. [1]

ROS stands for Robot Operating System. Even if it says so, ROS is not a real operating system since it goes on top of Linux Ubuntu. ROS is a framework on top of the O.S. that allows it to abstract the hardware from the software. This means you can think in terms of software for all the hardware of the robot. And that's good news for you because this implies that you can actually create programs for robots without having to deal with the hardware. Yeah! [2]

As I would say ROS is a tool which allows you to unite programs (nodes), which all take care of different parts of robot (hardware: sensors, actuators, motors etc.) and make them work together. ROS takes care of the communication between nodes, mentioned earlier (ROS documentation refers to them as "packages") and simply connects them and let them work correctly with each other and exchange data.

3 Installation and Configuration

For correct working of ROS and it's environment it's necessary to run it on Ubuntu Linux and not on virtual machine! It's necessary to run ROS on OC with enough RAM and ROM memory with system resources, which are abscent, when running it inside second OC. Tutorials on installing Linux double boot: [3]. Doble boot For that project I had not been using real robot yet, I had started with simulation. Software for simulation is available for installation with ROS. Tutorials on full installation of ROS and simulation software: [4], [5]. Next step after ROS installation is configure an environment for it and configure its tools and their workspace. Tutorials on configuring ROS: [6], [7].

Steps on correct installation of the ROS:

- Installation of Ubuntu Linux on double boot
- Installation of ROS
- Configuring ROS

My configuration: Ubuntu Linux 20.04 LTS + ROS Noetic

4 Created program:

For creating my program, I used those tutorials: [8], [9], [10], [11], [12].

Link to the git, containing the program: `https://github.com/kipariss1/warehouse_robot/tree/ProjectII_final`

The program itself contains class "TurtleBot". Class has its function "move2goal", which makes turtlebot move to the goal coordinates. When the function "move2goal" is being called, algorithm performs. Simplified algorithm looks like this: the robot always begins at the coordinates [0, 0]. As user inputs goal coordinates, the robot calculates an appropriate angle with function "goal angle" and rotates this degree with function "rotation move". Then calculates distance to the goal in the right direction with the function "goal distance" and starts to publish velocity to the turtlebot with function "linear move". When goal is achieved, turtlebot stops. If turtlebot sees obstacle with its laser sensors, then function "naive obstacle avoidance" is being called. The explanation of its algorithm and general algorithm will be in the block diagram below.

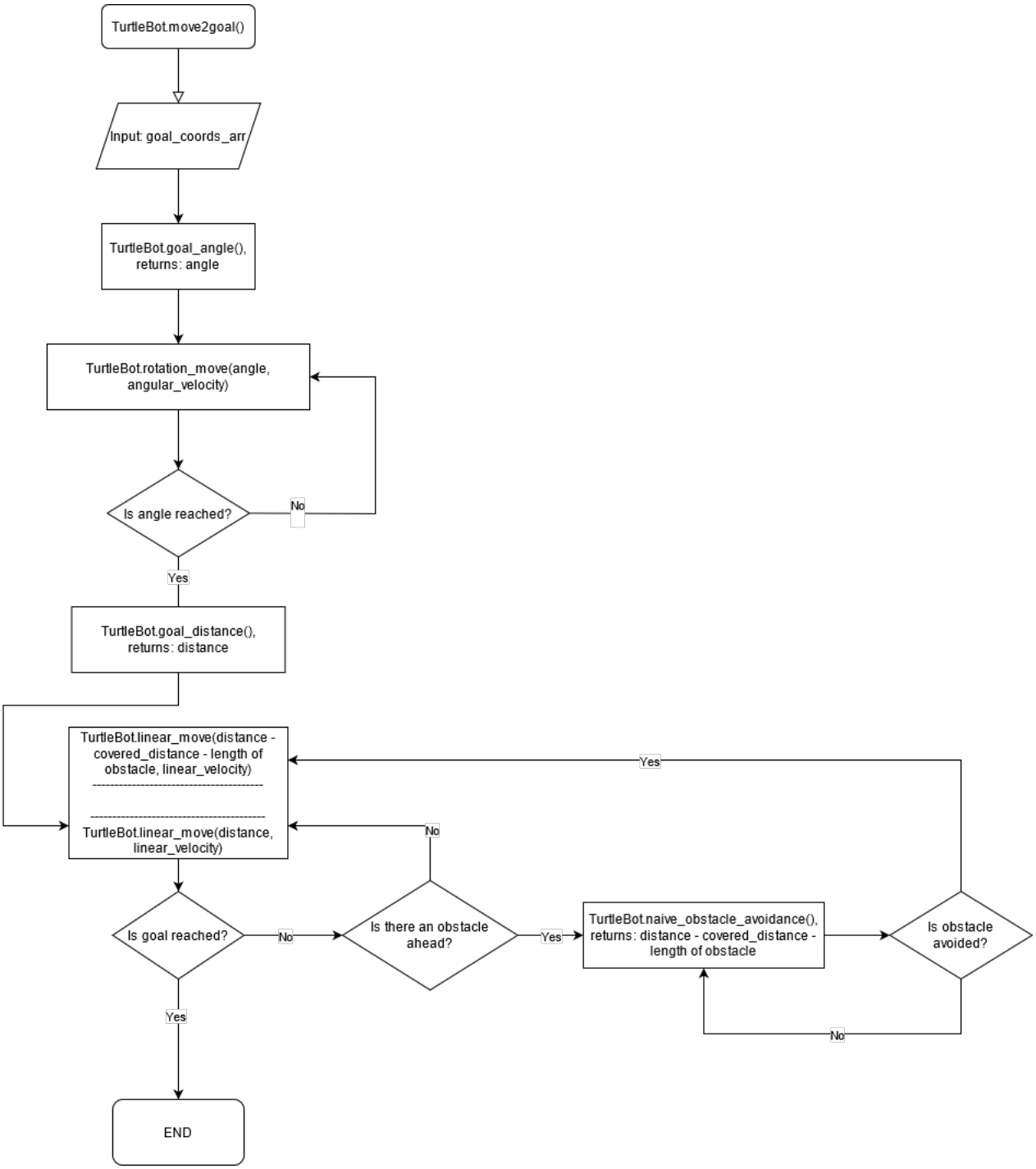

Figure 1: General diagram

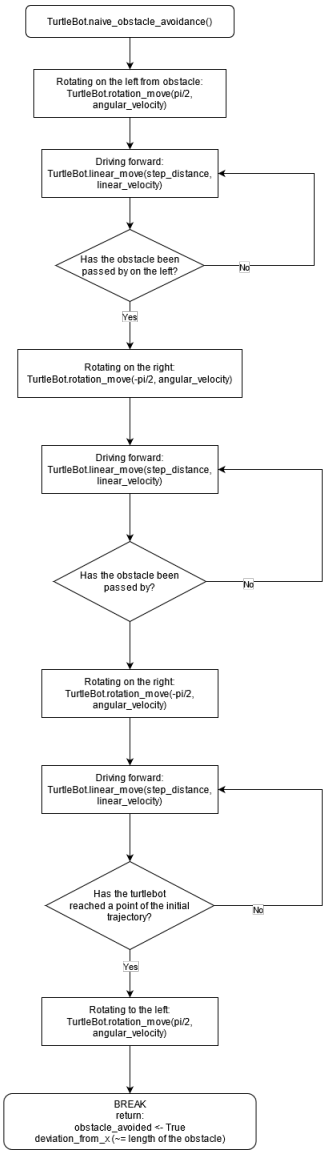

Figure 2: Naive obstacle avoidance function diagram

5 Links to sources

:

[1] - https://www.ros.org/about-ros/ - About ROS

[2] https://www.theconstructsim.com/what-is-ros/ - More about ROS

[3] https://www.youtube.com/watch?v=-iSAyiicyQY&t=488s&ab_channel=KskRoyal - Doble boot Linux+Windows

[4] https://www.youtube.com/watch?v=9U6GDonGFHw&list=PLJNGprAk4DF5PY0kB866fEZfz6zMLJTF8&index=1&ab_channel=JustinHuang - Installing ROS

[5] https://automaticaddison.com/how-to-launch-the-turtlebot3-simulation-with-ros/ - Launching Turtlebot3 simulation

[6] https://www.youtube.com/playlist?list=PLJNGprAk4DF5PY0kB866fEZfz6zMLJTF8 - Full course ROS for beginners

[7] http://wiki.ros.org/catkin/Tutorials/create_a_workspace - Creating catkin work space, necessary thing for working with ROS

[8] http://wiki.ros.org/ROS/Tutorials - General documentation of ROS

[9] http://wiki.ros.org/turtlesim/Tutorials then go → Moving in a straight line - Simple tutorial on moving in a straight line. Under capital number 9

[10] http://wiki.ros.org/turtlesim/Tutorials/ then go → Rotating left and right - Simple tutorial on rotating. Under capital number 9

[11] https://www.youtube.com/watch?v=eJ4QPrYqMlw&ab_channel=TheConstruct - Moving turtlebot

[12] https://www.theconstructsim.com/read-laserscan-data/?utm_source=youtube&utm_medium=ros_qa&utm_campaign=31 - Read laser scan data