Hungarian University of Agriculture and Life Science
Szent István Campus
MSc Mechanical Engineering Course

DEVELOPMENT OF THE CONTROL OF A ROBOTIC ARM USING ROS2

Primary Supervisor: **Tóth János**
Assistant Professor

Author: **Ibrahim Malli**
BLD9OB

Institute/Department: **Institute of Technology**
MSc Mechanical Engineering

Szent Istvan Campus, Gödöllő

2023

SUMMARY

Development of the Control of a Robotic Arm Using ROS2

Ibrahim Malli

Course, level of education: MSc Mechanical Enginering

Host Department/Institute: Institute of Technology

Primary thesis advisor: Dr. Tóth János, Assistant Professor, Institute of Technology, MSc Mechanical Enginering

In this study, it is aimed to develop the control of a robotic arm using ROS2. The collaborative robot arm, developed using the ROS 2 MoveIt 2 repository and the RViz visualization tool and ROS 2 Development Studio, is intended to be capable to move in the Gazebo simulation toolbox.

As a user of ROS, the goal of ROS is to build the development environment that allows robotic software development to collaborate on a global level. That is to say, ROS is focused on maximizing code reuse in the robotics research and development, rather than orienting towards the so-called robot software platform, middleware, and framework. To support this, ROS has the following characteristics.

To fulfil their purposes, many robots are required to interact with their environment, and the world around them. Sometimes they are required to move or reorient objects from their environments without direct contact by human operators. Despite the name, ROS is not, in fact, an operating system. Rather, it’s an SDK (software development kit) that provides the building blocks you need to build your robot applications. Whether your application is a class project, a scientific experiment, a research prototype, or a final product, ROS will help you to achieve your goal faster. And it’s all open source.

ROS is the most widely used middleware in robotics. It is the interface that allows the robot to process the data it receives from the outside world through sensors and send it back to the robot as a command. It allows to use different languages (C++, Python) on the same robot.

The primary goal of ROS is to support code reuse in robotics research and development which enables the robotic arm to be developed more easily thanks to common repository and documentations, facilitates robot coding in C++ language.

However, ROS supports soft real-time applications. ROS has a repository in which developers can use code that has been already written for robot components like motion planners and vision systems. In addition, it can be integrated easily with other popular open-source software libraries such as OpenCV, Point Cloud Library, Gazebo simulator, etc. As ROS is Linux based software, it cannot guarantee the hard real-time properties of a system.

Tool beside from ROS like Gazebo brings a fresh approach to simulation with a complete toolbox of development libraries and cloud services to make simulation easy. Iterate fast on your new physical designs in realistic environments with high fidelity sensors streams. Test control strategies in safety and take advantage of simulation in continuous integration tests. RVIZ is a ROS graphical interface that allows you to visualize a lot of information, using plugins for many kinds of available topics. Differences between RViz and Gazebo is basically RViz shows you what the robot thinks is happening, while Gazebo shows you what is really happening.

In conclusion, the collaborative robot arm was successfully capable to move in the ROS 2 Foxy Fitzroy environment, simulated in the Gazebo software. First trajected at the ROS MoveIt2 visualisation tool RViz then the necessary adjustments are made for the collaborative robot arm to take its initial position at the virtual environment (see Figure 1). When final planning is complete, it was executed on MoveIt2 graphical interface package and simulated in the Gazebo software (see Figure 2). This study was programmed with C++ and the help of repository ROS 1, ROS 2 and MoveIt2 open-source documentation. And ROS 2 Development Studio was used to make this study easy and fast.

Key Words: Robotic Arm, Robot Operating System-ROS, Programming

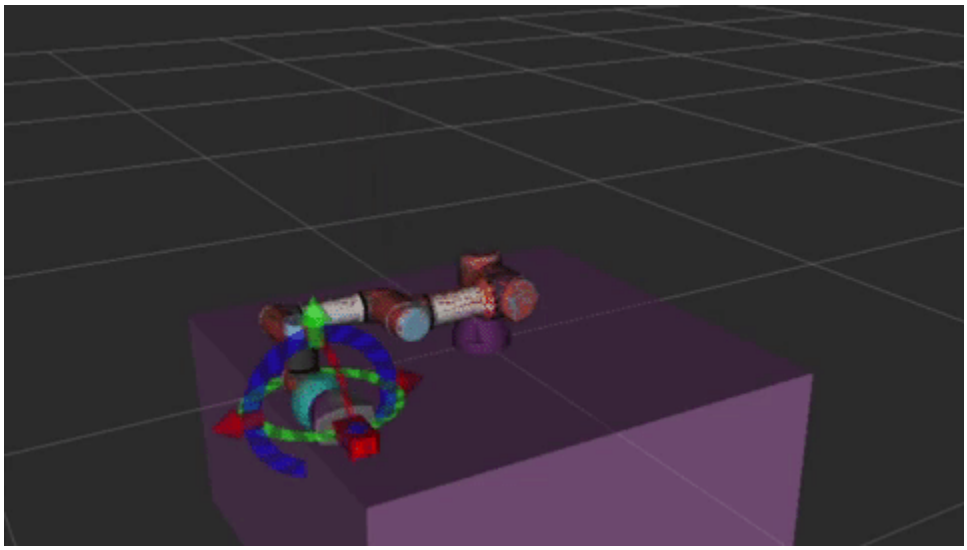

Figure 1. Visualization of the collaborative robot arm using with ROS MoveIt2 visualisation tool RViz

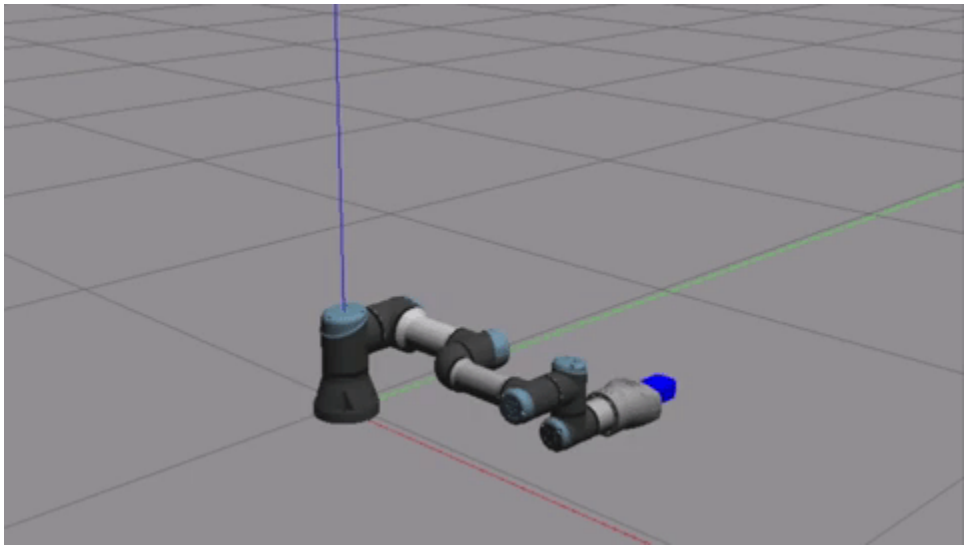

Figure 2. Visualization of the collaborative robot arm on Gazebo