DESIGN AND CONTROL OF A SERIAL ROBOTIC ARM USING ROS (ROBOT OPERATING SYSTEM)

İZMİR KÂTİP ÇELEBİ UNIVERSITY 2010

Ecem Betül ELMAS, İsmail Burak ŞATIR, Nurşen ÇALIK, Ruken Irmak DİNLER

Supervisor: Assoc. Prof. Dr. Fatih Cemal CAN

DEPARTMENT OF MECHATRONICS ENGINEERING

Abstract

6-DOF (Degree of Freedom) robot manipulators are six-joint robotic systems that allow a wide range of motion. These manipulators can be programmed using ROS (Robot Operating System), an open-source framework for robot software development. ROS provides several tools and libraries for programming robot manipulators, including motion planning, control, and simulation libraries.

1. Introduction

Robotics is an interdisciplinary branch of engineering that comprises the formation, design, building, and action of robots. This field overlays with mechanical, mechatronics, electrical engineering, computer science and electronics. In addition, Robotics deal with the computer systems for their control, sensory response, and information processing [1]. In this project, a serial robot arm, which is widely used in the industry, was designed. The robotic arm consists of six revolute joints. It is foreseen to be used for simple operations such as pick and place. The mechanical design of the robotic arm was made in the SolidWorks Design program, and all kinematic and dynamic analyzes were provided. The control of the Robotic Arm and its operation in the simulation environment are provided by ROS.

2. Robot Arm Design Methodology

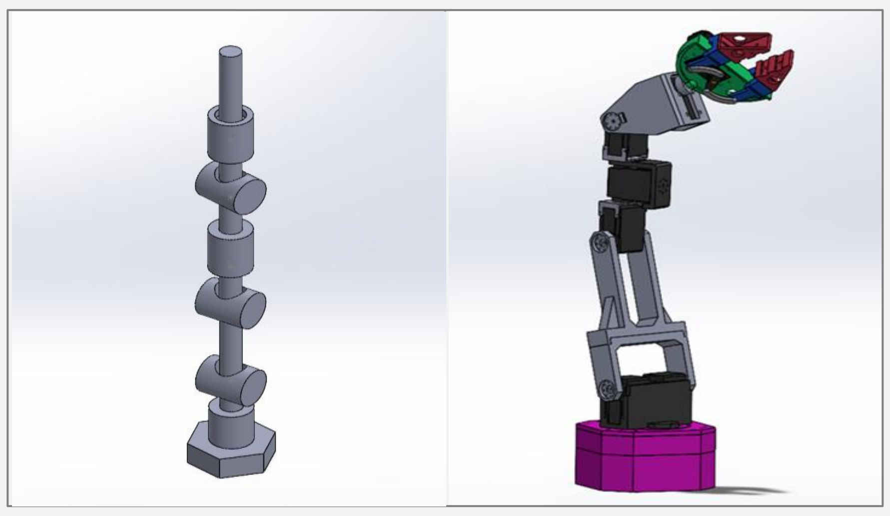

Figure 1: Mechanical Design of Robotic Arm

For the design of the robot arm, first, a conceptual design was prepared in Solidworks CAD program. Later, this design was detailed and the base, bearing, links and gripper were created, and the servo motors were placed.

3. Analysis Of Robot Arm

The analysis of the robot is divided into dynamic and kinematics. Kinematic analysis typically involves the use of mathematical equations, such as forward and inverse kinematics, to describe and calculate the relationships between the input parameters (such as joint angles or displacements) and the resulting motion of the system [2]. Denavit-Hartenberg (DH) method were used for forward kinematic analysis at Figure [2].

	$\alpha_{i-1,i}{}^{a}$	$L_{i-1,i}{}^{a}$	$d_i{}^{a}$	$\theta_i{}^{b}$
1	$\frac{\pi}{2}$	0	0	0°
2	0	62.45	0	0°
3	$-\frac{\pi}{2}$	0	0	0°
4	$\frac{\pi}{2}$	39.90	0	0°
5	$-\frac{\pi}{2}$	82.15	0	0°
6	0	65.9	11.50	0°

Link length (Li): the distance between the axis Z0 and Z1
Link offset (di): distance from origin O0 to the x1 axis
Joint angle (Θi): the angle between the X0 and X1
Link twist (αi): the angle between the Z0 and Z1

$$T_i^{i-1} = \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 & L_{i-1,i} \\ sin\theta_i cos\alpha_{i-1,i} & cos\theta_i cos\alpha_{i-1,i} & -sin\alpha_{i-1,i} & -sin\alpha_{i-1,i}.d_i \\ sin\theta_i sin\alpha_{i-1,i} & cos\theta_i sin\alpha_{i-1,i} & cos\alpha_{i-1,i} & cos\alpha_{i-1,i}.d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6^0 = T_1^0 * T_2^1 * T_3^2 * T_4^3 * T_5^4 * T_6^5 = \begin{bmatrix} 1 & 0 & 0 & 250.4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 11.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 2: Forward Kinematic Analysis Calculations

In the DH method, each joint is associated with a homogeneous transformation matrix that describes the relative position and orientation of the link attached to that joint with respect to the previous link.

Inverse kinematic analysis is the process of determining the required input for a robot arm to reach a specific end-effector position and orientation. This is the opposite of forward kinematic analysis [3].

Dynamic analysis of a robotic arm involves studying the arm's motion and forces, understanding the relationship between control inputs and resulting motion, and analyzing stability, controllability, and performance.

Inverse kinematic and dynamic analysis RoboAnalyzer program was used at Figure [2]. RobAnalyzer is a 3D simulation and analysis program used in the robotics industry.

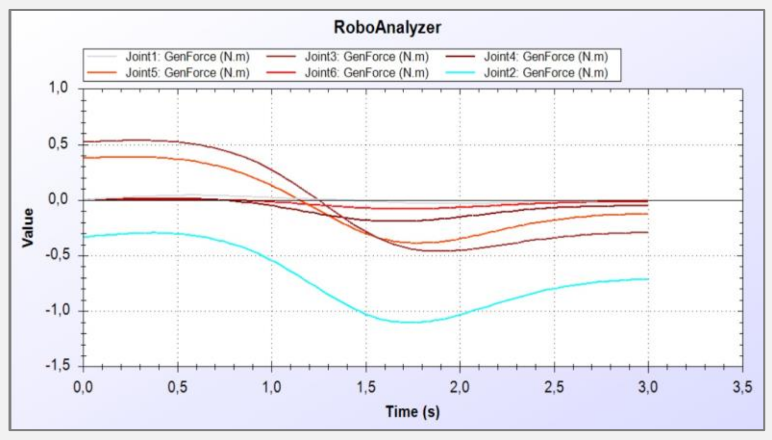

Figure 3 : Dynamic Analysis

4. ROS (Robot Operating System)

Today's robot system is a complex hardware device equipped with numerous sensors and computers which are often controlled by complex distributed software. Robots must navigate and perform successfully specific tasks in various environments and under changing conditions. However, it is costly and time consuming to build different test fields and to examine the robot behavior under multiple conditions. Using a well-developed simulation environment allows safe and cost-effective testing of the robotic system under development. The simulation decreases the development cycle and can be versatility applied for different environments.

ROS is an open-source framework widely used in the field of robotics for developing and controlling robotic systems. ROS provides a collection of tools, libraries, and conventions that aid in building and managing complex robot applications. It offers a flexible and distributed architecture that allows different components of a robot system to communicate and work together seamlessly.

Software in the ROS Ecosystem can be separated into three groups: language-and platform-independent tools used for building and distributing ROS-based software; ROS client library implementations such as roscpp rospy, and roslisp; packages containing application-related code which uses one or more ROS client libraries.

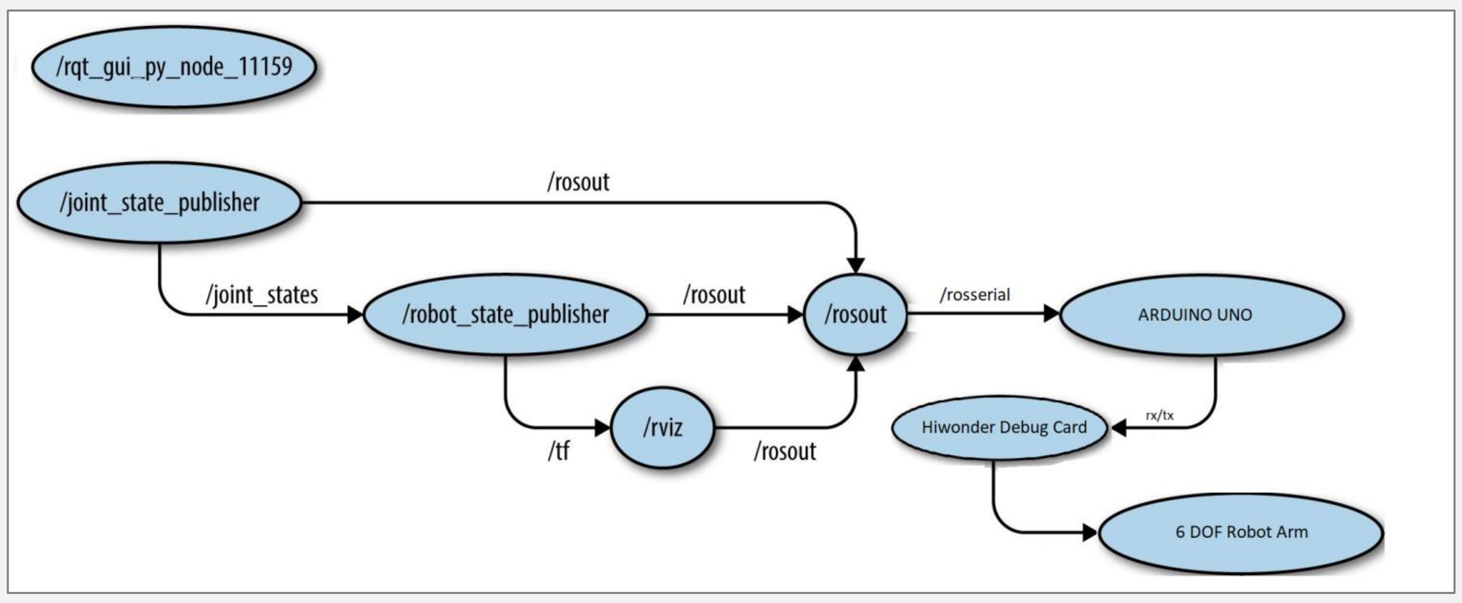

Figure 4: Operation Chart

Overall, ROS simplifies the development of robotic systems by providing a standardized and modular framework for building, controlling, and integrating various components of a robot. It has become a popular choice in academic and industrial settings for developing robotic applications across a wide range of domains.

5. Conclusions

The project includes the development of a series of robotic arms with six degrees of freedom and 6 rotary joints. The primary objectives of the project are to design the robot arm, to perform kinematic and dynamic analyzes on the design, and to control it using the Robot Operating System (ROS) in the simulation environment. The specific task chosen for the robot arm is the pick and place task, which is a common practice for robotic arms in industry. Programming a 6 DoF robot manipulator in ROS involves creating a model in a simulation environment like Gazebo and writing code using ROS's libraries and tools. [4]

References

[1] Design and Control of 6 DOF Robotic Manipulator, UNIVERSITY OF ENGINEERING AND TECHNOLOGY LAHORE FAISALABAD CAMPUS thesis class. (2018, May)
[2] Introduction to Robotics: Mechanics and Control. Pearson Education. Craig, J. J. (2004).
[3] Forward and Inverse Kinematic Analysis of Robotic Manipulators. (2017, Feb), International Research Journal of Engineering and Technology (IRJET)
[4] Programming Robots with ROS Morgan Quigley, Brian Gerkey, and William D. Smart Chapter 18. Your Own Robot Arm