2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), 7-9 February, 2019

Manipulation and Analysis of a 4-DOF Robotic Arm Using a Peer-Peer Messaging System

Md. Kaiser Raihan
Department of Electrical and Electronic Engineering
Chittagong University of Engineering and Technology
Chittagong, Bangladesh
E-mail: rahikaiser@gmail.com

Hameed Hasan Alvee
Department of Electrical and Electronic Engineering
Chittagong University of Engineering and Technology
Chittagong, Bangladesh
E-mail: alvee.cuet@gmail.com

Mohammad Rubaiyat Tanvir Hossain
Department of Electrical and Electronic Engineering
Chittagong University of Engineering and Technology
Chittagong, Bangladesh
E-mail: mrth.eee.cuet@gmail.com

***Abstract*— The robotic arm is a mechanical complex system noticeably used for multiple industrial applications, medical and personal applications to make complex task easier and safer. This research work depicts the simulation and implementation of a robotic arm that sorts objects based on its color using color image processing. The platform was implemented using Robot Operating System which enabled the real time monitoring of the robotic arm movement. Robot Operating System is one of the system that utilizes peer-peer messaging system to control, simulate and analyze a robotic arm operation for precise and robust operation.**

***Keywords*— ROS, Degree of Freedom, Publisher, Subscriber, Node, Color Detection, Pneumatic gripper**

I. INTRODUCTION

Robotic arm is an electromechanical device consisting of joints and links, driven by motors or other actuators and it is able to do special task as per instruction. The design parameters, size and performance of the system is set based on its applications. It is guided by sensors and controlled through a software program hosted in a microcontroller. It helps to manipulate and handle parts or tools for performing various operations in different kinds of real world applications. A robotic arm with the correlation of mechanical structure and electronic control can carry out a complex, tedious and unsafe task with great precision and accuracy. Robot arm is widely used in many applications like material handling, welding, assembling, dispensing, thermal spraying, painting and drilling. For positioning the arm in robot work space based on the links arrangement and joints movement five basic robot configurations are available. Cartesian, cylindrical, spherical, SCARA and articulated. The most preferred robot arm for pick and place application has articulated configuration. This type of configuration is very useful as it can simulate a human arm as a result it is also called anthropomorphic arm. The work space of this configuration is spherical in shape. This anthropomorphic structure is the most dexterous one, because all the joints are revolute and widely used in industrial applications. The existing robotic arm models are used for achieving specific tasks mainly in industries. The fact that cost of such robot arms is high makes robotic algorithm studies with hardware models difficult. It is necessary to have robot arm models that provides user friendly environment for students, which creates interest and helps in learning different application specific algorithms.

Among some of the previous works related to robotic arm, Robotic arm using stereo vision and Lab Windows is noteworthy [1]. Here the robotic arm position was detected through color markings on the arm which helps to detect the location of the arm in co-ordinate system. Another work used the 3D kinetic and kinematic properties of arms movement to be imitated by the robotic arm [2]. These approach uses the Inverse Kinematic system for pinning the exact location of the arm. The system analyses the trajectory of the arm movement and performs the evaluation. Use of accelerometer for the control of Robotic Arm is also noteworthy [3]. The accelerometer gives feedback to the computer which controls the robotic arm. In this approach the accelerometer is connected to the robotic arm which during operation gives the value of acceleration of the different joints of the Robotic Arm just enabling user to define the position of the joints. Robotic Arm movement through programmed manipulation path-planning is also notable [4]. User defined path-planned robotic arm has some limitations as it can cause percentage error during picking and placing due to wear and tear. User control robot through speech and voice recognition system is also a good approach to perform mediocre task [5]. In 2016 Hwi-Su Kim and Jae-Kyung Min developed an approach that used multi-degree of freedom counter balance mechanism based on spring and slide-crank mechanism to reduce payload on the actuators [6]. Moreover, famous robots such as ASIMO [7] and WABIAN-2R [8] are noteworthy though very expensive and agile. In low cost robotic arm Development University of Cassino became pioneer through development of a 1-DOF robotic arm capable of single and combined motion through the manipulation of clutch states [9].

In this paper we have implemented a ROS based robotic arm that classifies and picks object using color image processing.

It also enables the real time monitoring of the movement of robotic arm through simulation. The arm also utilizes pneumatic gripper which eases the pick and place task.

II. ROBOTIC ARM SYSTEM

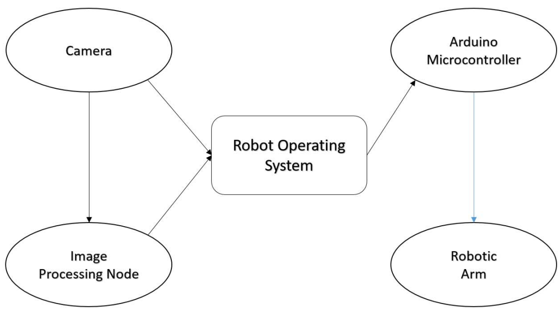

Fig. 01: Block Diagram of the Robotic System

A. Block Diagram

The whole system runs using Robot Operating System. As a result, each and every steps can be depicted using nodes. Fig. 01 depicts a simple block diagram of the project. Which consists of basic components, which are; Camera Node, Image Processing Node, A Robot Operating System, An Arduino and the Robotic Arm.

B. The Nodes

The master node is the Robot Operating System which controls the publication and subscription of all the other nodes. In LINUX operating system the Operating System can be initiated using the terminal window by calling ROSCORE. ROS starts with the ROS Master. The Master allows all other ROS pieces of software (Nodes) to find and talk to each other. That way, we do not have to ever specifically state “Send this sensor data to that computer at 127.0.0.1. We can simply tell Node 1 to send messages to Node 2.

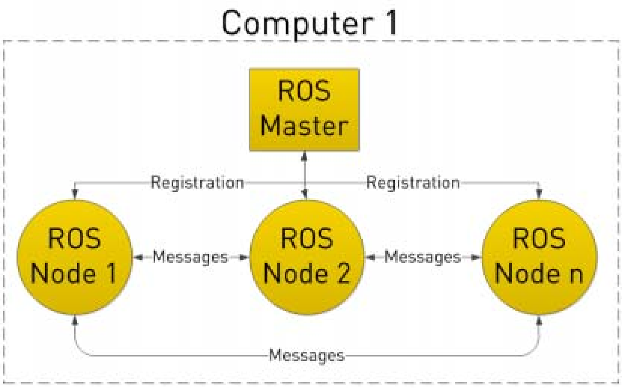

Fig. 02: Nodal Representation of the Robot

Nodes do this by publishing and subscribing to Topics. Since there is a camera on our Robot shown in Fig. 02 We can be able to see the images from the camera to process the image [10]. Here we have a Camera Node that takes care of communication with the camera, an Image Processing Node on the robot that process image data, and an Image Display Node that displays images on a screen. To start with, all Nodes have registered with the Master. The Master Node acts a hub where the nodes come to see where they are to send the messages. In registering with the ROS Master, the Camera Node states that it will Publish a Topic called /image_data (for example). Both of the other Nodes register that they are Subscribed to the Topic /image_data. Thus, once the Camera Node receives some data from the Camera, it sends the /image_data message directly to the other two nodes. (Through what is essentially TCP/IP)

III. Robot Structure

A. 3D Design

The Robotic arm structure was designed in two steps. At first a 3D model was designed for simulation in RVIZ using URDF. Unified Robot Description Format (URDF) is an XML based system for designing robot structure in ROS. The URDF model was used to decide the length of the different parts of the arm and their movement according to their position. Fig. 03 shows the URDF model of the robot. And to design the robotic arm in real world with precision Solidworks model was designed. Solidworks is an CAD design software. The different part of the CAD design was printed on plastic wood and cut accordingly to get the accurate measurement of the arms of the Robot. Fig. 04 shows the 3D design of the robot designed in Solidworks.

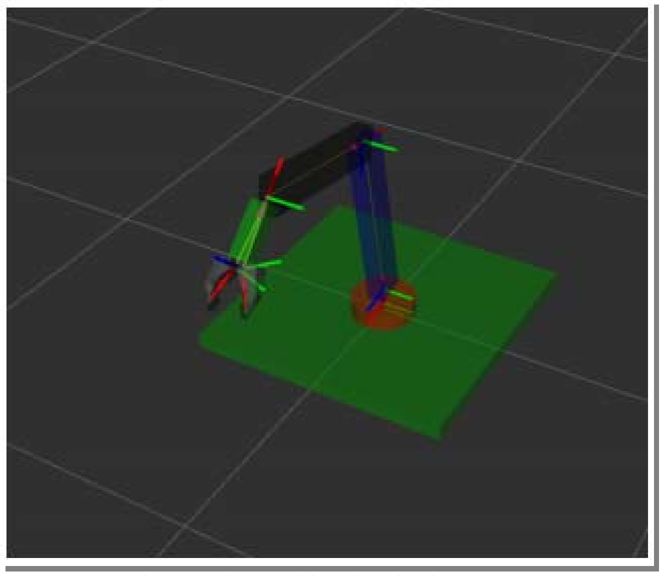

Fig. 03: Robot Model Designed Using URDF

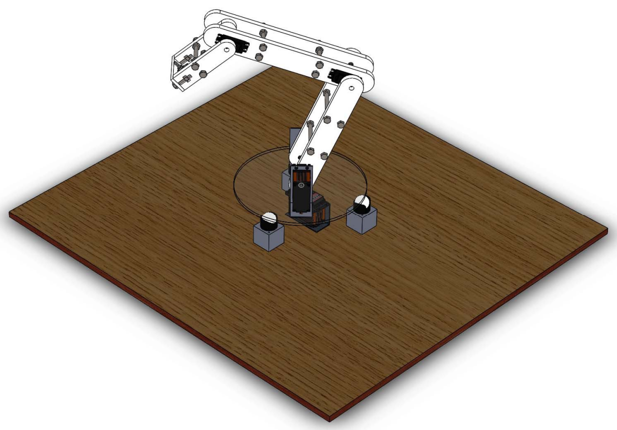

Fig. 04: Robot Model designed in Solidworks

B. Mechanical and Electrical Components

The Robotic Arm was made using plastic-wood which is easier to drill and tap. The plastic wood was 5mm thick and two pieces were stacked for each part of the arm. Pololu caster ball used for support in the base with the servo which enabled easy movement of the base. The clamp for the base motor was cut using CNC machine from the design in Solidworks. Two types of servo motor were used- S8203 and Bluebird 660DMG. The S8203 was used in the base for its higher torque. The gripper used for the robot is pneumatic based. A vacuum suction pump was used for the gripper for easy pick and place. The electrical part of the robot consists of Arduino UNO, 5V Relay module, Lithium polymer battery, Buck converter, Logitech Webcam.

IV. Software Architecture

A. Image Processing

The OpenCV image processing library was used for color detection and classification of the object. The whole system was implemented using python programming. The frames taken from the video input of the webcam was first converted to HSV color format and then upper and lower threshold value was to detect certain color. For the whole image processing an ROS package was made to implement the image processing node. Determining the color of the object this node sends a message to the topic to take necessary action.

B. Rosserial package

The Rosserial package enables the ROS architecture to communicate with the Arduino used in the robot and control the robot to move to certain position. This package uses serial communication at a certain baudrate to communicate with the robot. The baudrate used for the robot is 9600.

C. Joint State to plain package

The Joint state publisher node publishes different parameters like angle, position, velocity etc. for controlling the different joints of the arm. But publishing so many information to the Rosserial node makes it impossible for the robot to work properly. As a result, the Joint State to plain package was made to truncate the message to relevant angle information only.

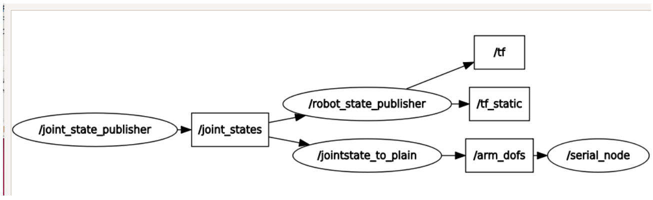

Fig. 05: Final Nodal Representation during Operation

V. SIMULATION

The simulation of the robotic arm was performed along with the feedback that was taken which help in the precise movement of the robotic arm. The Robot Operating System controls the autonomous operation of the Robotic Arm. It follows the following block diagram in Fig. 05 and controls the Robotic Arm.

The simulation result of the Robotic Arm was taken for different position and it was seen that for different position we get different graph. Moreover, the joint_state_publisher shown in Fig.6 shows the angle in radian of different joints and segments of the robotic arm during its operation. The simulation result in Fig. 06 is obtained precisely throughout the robotic arm operation by following the block diagram in Fig. 5 for the entire time.

The Analysis from the simulation depicts the operation of the robotic arm. The graph shows the precise position of the joints of the robotic arm while slider shows the angle.

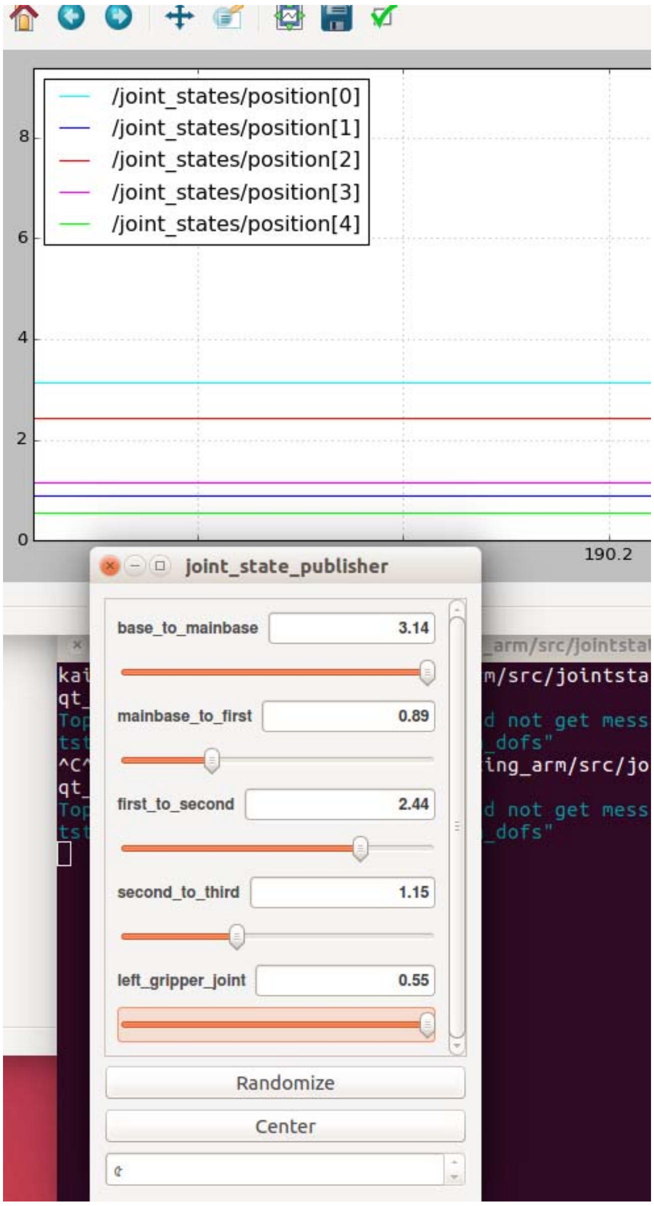

Fig. 06: Simulation Analysis for a particular position

VI. IMPLEMENTATION & EXPERIMENTAL RESULT

The robot was implemented to pick object of different color and place the objects to their respective position. Fig. 07 demonstrates the implemented robot. Fig. 08 shows the control circuit of the robot drawn using Fritzing software. The robot was able to classify the object according to their color. The setup was tested for several light conditions and the robot worked well except for extremely low light condition.

The robot gripper's picking capacity was also tested different object. The gripper worked well for objects of weight up to 1 kg. The gripper was unable to pick objects with too uneven surface. Fig. 09 depicts the operation of the arm while picking a blue color object.

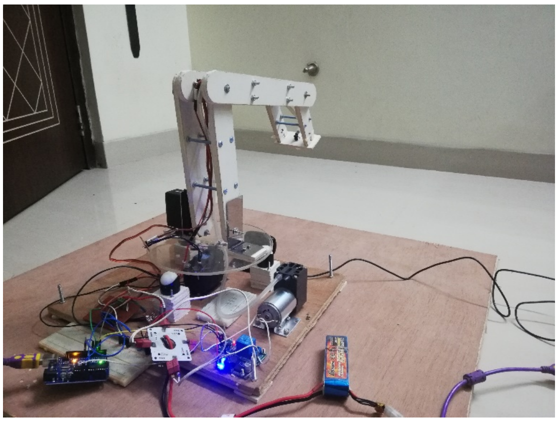

Fig.07: Implementation of the robotic arm

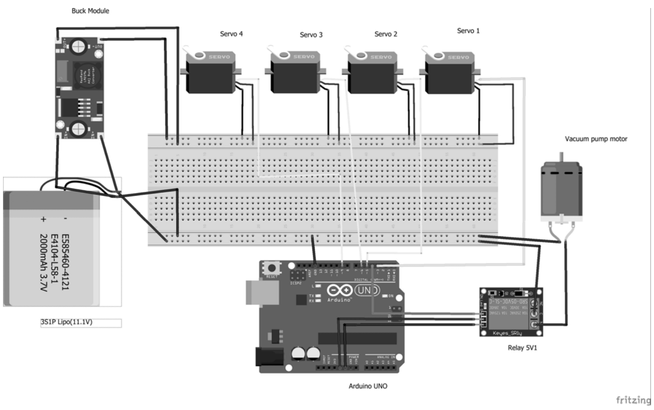

Fig.08: Schematic of the robot control circuit

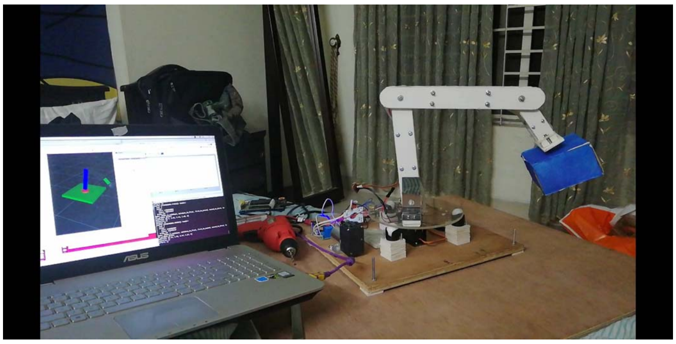

Fig. 09: Robotic arm picking up a blue color box

VII. CONCLUSION

In this research, a robotic arm was designed and implemented for autonomous pick and place of object and sort the objects' according to their color. The whole system was implemented in ROS to simulate the robot and maintain precise control of the robot. Simulation of the robot in ROS prior to the implementation made it possible to reduce the errors. In future more advanced sensor like Kinect can be used to detect the position of the objects along with their color and other properties like shape and size for more advanced level of autonomy and control.

REFERENCES

[1] Roland Szabó and Aurel Gontean, "Robotic Arm Control with Stereo Vision Made in LabWindows/CVI," 38th International Conference on Telecommunications and Signal Processing (TSP), 2015, pp. 1-5.

[2] Hwi-Su Kim, Jae-Kyung Min, and Jae-Bok Song, "Multiple-Degree-of- Freedom Counterbalance Robot Arm Based on Slider-Crank Mechanism and Bevel Gear Units," IEEE Transactions on Robotics, vol. 32 , issue. 1 , pp. 230-235, February 2016.

[3] Kadirimangalam Jahnavi and Sivraj P, "Teaching and Learning Robotic Arm Model," 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), July 2017.

[4] Elham Saraee, Ajjen Joshi and Margrit Betke, "A Therapeutic Robotic System for the Upper Body Based on the Proficio Robotic Arm," 2017 International Conference on Virtual Rehabilitation (ICVR), pp. 1-2, June 2017.

[5] Hao Gu and Marco Ceccarelli, "Simulation of Combined Motions for a I-DOF Clutched Robotic Arm," 2009 IEEE International Conference on Mechatronics and Automation, August 2009.

[6] S. Rosa, A. Russo, A. Saglinbeni, G. Toscana, "Vocal Interaction with a 7-DOF Robotic Arm for Object Detection, Learning and Grasping," 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), March 2016.

[7] Pedro Tavares and Armando Sousa, "Flexible Pick and Place Architecture using ROS Framework," 2015 10th Iberian Conference on Information Systems and Technologies (CISTI), June 2015.

[8] M.-A. Lacasse, G. Lachance, J. Boisclair, J. Ouellet, and C. Gosselin, "On the Design of a Statically Balanced Serial Robot Using, Remote Counterweights," 2013 IEEE International Conference on Robotics and Automation, pp. 4174–4179, May 2013.

[9] Hwi-Su Kim and Jae-Bok Song, "Multi-DOF Counterbalance Mechanism for a Service Robot Arm," IEEE/ASME Transactions for Mechatronics, vol. 19, issue. 6, pp. 1756–1763, December 2014.

[10] Robot Operating System NODE operation, https://rosindustrial.org/ (20/1/2019)