Controlling the Bionic ARM to Certain Gestures using ROS Serial Communications

Kavya. P. S
PG Scholar, Robotics and Automation
College of Engineering
Trivandrum, Kerala, India

Jaimon Cletus
Professor, Mechanical engineering
College of Engineering
Trivandrum, Kerala, India

Abstract—The project aims to provide an aid which will help the physically disabled ones to enhance and upgrade their human capabilities. Those people with disabilities face discrimination and barriers that restrict them from participating in society on an equal basis with others every day. The work is to develop a robotic arm that provides sufficient mechanical support with better accuracy within affordable cost. The system has sought to simulate the natural movements of the arm and hand. Since this project is beneficial to the paralyzed patients, it is socially important in many cases. Researches are done on automatic prosthetic using different methods. The main disadvantages of these methods were its structural complexity and high manufacturing cost.

Keywords—Robotics and automation; bionic arm; Robotic Operating System ; ROS serial communication

I. INTRODUCTION

The project aims to develop an exoskeleton robotic suit. HAL (Hybrid assistive limb)[1] will help people with disabilities to do their daily needs without any external help. Persons with disabilities face discrimination and barriers that restrict them from participating in society on an equal basis with others every day. They are denied their rights to be included in the general school system, to be employed, to live independently in the community, to move freely, to participate in sport and cultural activities, to enjoy social protection etc. A person with physical disability is not able to execute activities associated with movement, both the persons himself and objects. Such immobility resulting from disorder of the muscular or skeletal system, may include disability resulting from injury or disease. Technology is a valuable tool that can be used to improve the lives of people with disabilities. In recent years, the number of amputation patients continues to rise due to disease, traffic accidents etc. Cosmetic devices developed for amputated persons have no active functionality; they provide only passive support, which will restrict the free movement of human body.

II. METHODOLOGY

A. Design a human like bionic arm

The conceptual design is brought into virtual platform with the aid of Solid Works. The required joints and parts are assembled and finger mechanism has been done according to the desired hypothesis.

B. Safety Analysis, Prototyping and Testing

Safety analysis and optimization is performed using simulation platforms such as ROS (Robotic Operating Systems). Technical drawings are prepared. Assembling and prototyping is performed in FAB lab. User testing of prototype is performed and necessary modifications are done in iterative manner.

III. CAD MODELLING AND HARWARE FABRICATION

The CAD system is build from the GUI interface (Graphical User Interface), which is interacting with the B-rep data (Boundary representation) or geometry using the geometric modeling kernel. It may also be employed to manage the associative relationships between the geometry in sketch or a component in an assembly.
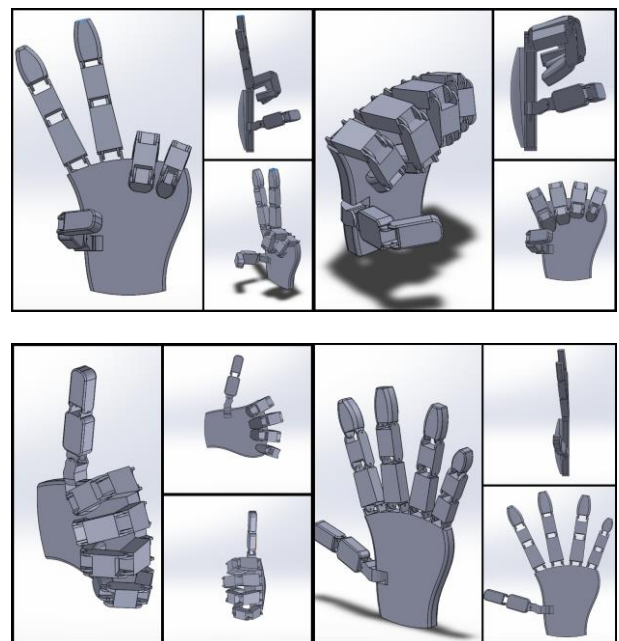
A. MODELS

Figure1: Model gestures (Solid Works)

B. 3D Printing the Hardware

- The Arm is 3D printed as per the required CAD model.
- Printer Details: FDM-Fused Deposition Method
- Material Used : PLA
- Tensile Yield Strength : D638 psi
- Tensile Modulus: D638 psi
- Melting Point : D3418°C
- Glass Transition Temperature : D3418°C

C. Electrical Design And Control Parameters.

- Arduino Uno comes with USB interface. The USB port is added on the board to develop serial communication with the computer.
- It is an open source platform where anyone can modify and optimize the board based on the number of instructions and task they want to achieve.
- This board comes with a built-in regulation feature which keeps the voltage under control when the device is connected to the external device.
- Reset pin is added in the board that reset the whole board and takes the running program in the initial stage.
- This pin is useful when board hangs up in the middle of the running program; pushing this pin will clear everything up in the program and starts the program right from the beginning.
- Operating Voltage: 5 Volts
- Input Voltage: 7 to 20 Volts
- A servomotor[27] is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration.
- It consists of a suitable motor coupled to a sensor for position feedback.
- It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.
- They are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.
- Servomotors are generally used as a high-performance alternative to the stepper motor.
- Using standard logic level signal to control , will be able to drive a two-phase stepper motor or four-phase stepper motor, and two-phase DC motors.
- Adopt a high-capacity filter capacitor and a freewheeling diode that protects devices in the circuit from being damaged by the reverse current of an inductive load, enhancing reliability.

- The module can utilize the built-in volt tube 78M05 to obtain 5v from the power supply.
- But to protect the chip of the 78M05 from damage, when the drive voltage is greater than 12v, an external 5v logic supply should be used.
- Drive voltage: 5-35V; logic voltage: 5V and PCB size: 4.2 x 4.2 cm

IV. SIMULATION USING 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. ROS was built from the ground up to encourage collaborative robotics software development. For example, one laboratory might have experts in mapping indoor environments, and could contribute a world-class system for producing maps. Another group might have experts at using maps to navigate, and yet another group might have discovered a computer vision approach that works well for recognizing small objects in clutter.

ROS was designed specifically for groups like these to collaborate and build upon each other's work, as is described throughout this site.

A. ROS-Robotic Operating System

Developing a robot with a computer brain requires a bunch of software tools on the computer side—like software drivers, third party tools for computer vision and simulation tools. ROS framework gathers all these tools and manages to develop a code for robot. It's a special framework initially developed by the Stanford AI Laboratory in 2007 for developing robots. The Open Source Robotics Foundation now maintains ROS.

The concept of ROS goes far beyond just a framework. ROS is an OS in concept because it provides all the services that any other OS does—like hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It is possible to integrate ROS with real time code. Even though ROS is still a framework that isn't a stand alone OS or an RTOS and isn't the only framework for robots.

B. Steps for acheiving robot position control using ROS

The URDF(Unified Robotic Description Format) should be generated on solid works named as arm (preferred).

- First open a new terminal on ROS either by searching terminal on search bar or by pressing ctrl+alt+T at same time ,and one or more tab can be opened on same terminal by pressing ctrl+shift+T.
- Robot URDF is the basic key element for simulation on ROS and which is to be moved on to a workspace named catkin ws, using the code 'cd catkin ws' on terminal or by normal copy paste method in src on catkin ws. Then should generate a moveit package. Intialising the move it package by 'roslaunch moveit setup assistant setup assistant.launch'.
- A new moveit package window is opened, click on create New Moveit! Configuration Package , Browse then from the desired location the package robot 1 is selected from which URDF is loaded on the moveit package. If the robot model is successfully loaded, robot model appears on window.
- Next Self-Collision matrix needs to be generated. It checks for each link pair and categorizes the links as always in collision, never in collision , default in collision, adjacent links disabled, and sometimes in collision and it disables the pair of links which makes any kind of collision.
- Virtual Joint need not to specify on here hence it is for mobile robots.
- Planning group is a group of joints/links in a robotic arm which plans together in order to achieve a goal position of a link or the end effector.
- Two planning groups are to be created, one for the arm and one for the gripper,since end-effector is not used, only one planning group is created.
- Planning Group, planning group window appears, in this window Group Name is given. Next is the

selection of the Kinematic Solver. From the drop down box select Kinematics Plugin.

- The arm group Kinematic Chain is added. For adding different poses for robot, click on the robot poses and same in case for adding robot pose, where fixed poses like the zero position and home position in the robot configuration are added. That's where we include our desired hand gestures.

- The last step is to generate configuration files. Click on the Browse button to locate a folder where configuration file that is generated by the Setup Assistant tool has to be saved . After browsing the required folder, click on the Generate Package button, this will save the files into the chosen folder.

- Then Exit Setup Assistant, which will exit us from the tool.

C. Motion Planning Of Robotic Arm in Rviz

Rviz allows to create new planning scenes where robot works, generate motion plans, add new objects, visualize the planning output and can directly interact with the visualized robot.

Moveit! package includes configuration files and launch files to start motion planning in Rviz. Next step is visualization of the robot for that demo launch file is launched the command is 'roslaunch rpp moveit demo.launch'.

Rviz window is opened which consists of several tabs, on context tab from OMPL drop down list select RRT. Next is Planning tab where, the start state, goal state plan a path, and execute the path of the robot can be assigned. Plan button, is used to plan the path from the start to the goal state, and if the planning is successful, path can be executed by using Plan and Execute button.
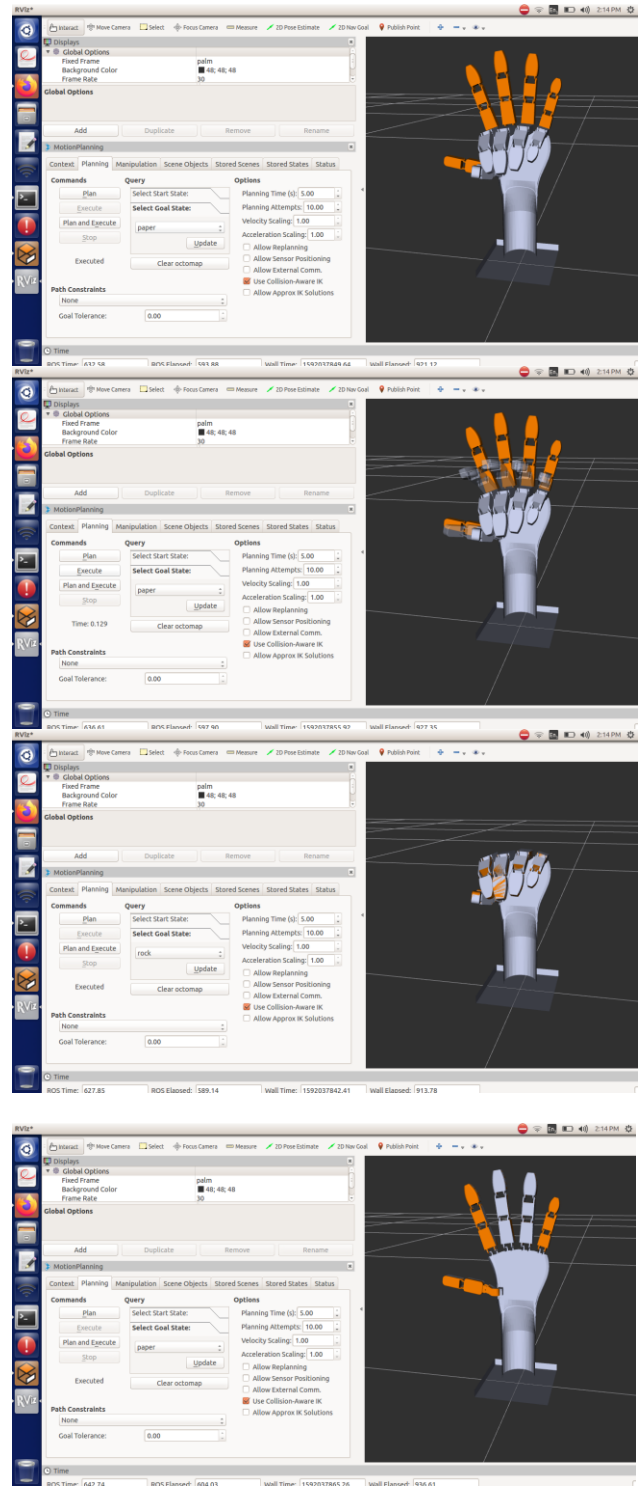

Figure 2: Motion Planning Using Rviz

D. Robotic arm simulation using Gazebo

Gazebo is a multi robot simulator for complex indoor and outdoor robotic simulation. We can simulate complex robots, robot sensors, and a variety of 3D objects.

The simulation model for a robotic arm by updating the existing robot description can be created by adding simulation parameters. To launch the existing simulation model, add world.launch and bringup.launch files needs to be

added to moveit! Package. Now open a new terminal to launch Gazebo using the code 'roslaunch rpp moveit arm arm' world.

Rviz is used to control the robot on Gazebo, which creates a robotic environment similar to the real world. The controller can be called using the command roslaunch rpp moveit arm arm bringup. After running this command both the Gazebo and Rviz windows are opened together and the motion path planned on the Rviz is achieved on the gazebo.

When the actual robot is fabricated the Gazebo is relaced by the robot and the robot can be controlled by using Rviz. After completing all the above steps the robot can achieve the desired poses.
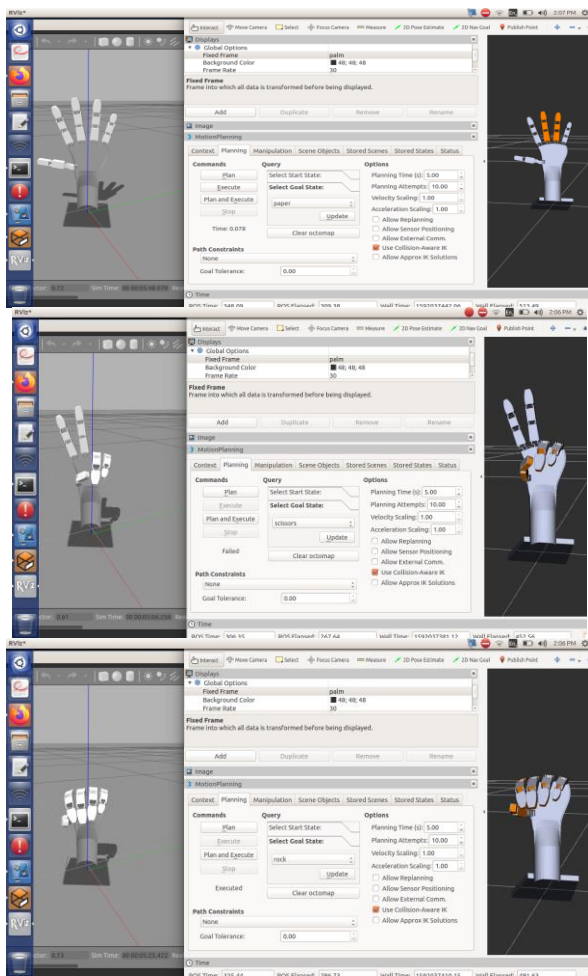

Figure 3: Execution Using Gazebo

V. HARDWARE INTERFACE

The finger mechanism is controlled using servo motors. There are 5 servo motors. A servo shield is used with the Arduino-UNO to perform the finger actions accordingly. Data is output from a sensor via a controller and interface pair. Just as a physical model may use a controller and interface to receive control commands, a sensor can use the controller and interface to process sensor data and publish it to lib-gazebo. Sensors do not have a physical representation.

Instead they define how data is captured from the Gazebo environment.

Figure   4: Hardware showing desired gestures

RESULT AND CONCLUSION

The main objective of this thesis is to develop a robotic arm that assists the arm amputees to perform certain hand gestures. The goal to design and develop an artificial limb having sufficient grasping strength, response speed and a visual appeal in an open source platform is achieved. The components required for the robot has manufactured by using laser cutting and 3D printing technologies. Finally a 5 DOF robot has manufactured to achieve the objectives.

A control system has to be developed for the motion of the robot.It includes obtaining the coordinates and joint angles of the target position. The trajectory planning can be done using softwares by interfacing the robot with ROS. Hardware interface using ROS and Arduino is done using ROS serial communication.

REFERENCES

[1] EunyoungBaek; Seok-kiSong; SehoonOh; SamerMohammed; Doyoung-Jeon; Kyoungchul Kong, IEEE 13th International Workshop on Advanced Motion Control (AMC), 2014

[2] R. Achintha M. Abayasiri; D.G. Kanishka Madusanka;A.T.S. Silva," MoBio: A 5 DOF Trans-humeral Robotic Prosthesis",International Conference on Rehabilitation Robotics (ICORR) QEII Centre, 2017.

[3] Atsushi Saito; Alexsandr Ianov; Yoshiyuki Sankai, "Measurement of brain activity using optical and electrical method", IEEE International Conference on Robotics and Biomimetics (ROBIO), 2009.

[4] F. Ichihashi, Y. Sankai, S. Kuno, Development of Secure Data Management Server for eHealth Promotion System, International Journal of Sport and Health Science,Vol.4, 2006

[5] N. Kurokawa; N. Yamamoto; Y. Tagawa; T. Yamamoto; H. Kuno, "Development of hybrid FES walking assistive system – Feasibility study-".The International Conference on Advanced MechatronicSystems,2012.