International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 9, Issue 5, September - October 2020 ISSN 2278-6856

Robust Control Platform for Robotic Arm Using ROS

Dr. Ignatius Antony Herman[1], Mr. Nikhil Amala Jerrin J[2], Mr. Kishore S [3], Ms. Cecil Ignatius Hermina [4] and Ms. Sneha N [5]

[1]Professor and Head Of Computer Science Department ,
DMI – St. Eugene University, Lusaka, Zambia

[2]Department of Mechatronics, Bannari Amman Institute of Technology
Sathyamangalam, Erode, Taminadu

[3]Department of Bio Medical Engineering, Bannari Amman Institute of Technology
Sathyamangalam, Erode, Taminadu

[4]Department of Electronics and communication Engineering, Bannari Amman Institute of Technology
Sathyamangalam, Erode, Taminadu

[5]Department of Electrical and Electronics Engineering, Bannari Amman Institute of Technology
Sathyamangalam, Erode, Taminadu

Abstract: ***Over the horizon of automation and robotics, the designing of an economically efficient robot with multiple features via an open source tool becomes immensely important and greatly valid. Though substantially demandable Computer Numerical Control (CNC) programming constituting g code and m code for controlling , starting, stopping of robots is used as a procedural instruction for 3D printing it lacks in the domain of feedback control. An open source moveo which is an open design is being stimulated . Here , It is made to perform as a powerful equipment reconnoitering multiplex actions. Initially beginning with the construction of the robotic arm, the proposed idea deals with stimulating, controlling the real robot and the interface between these two. Simulation of the robot involves the creation of URDF file of the moveo using solidworks. Controlling deals with the coupling of Arduino Mega 2560 along with a RAMPS 1.4 and the stepper motors. And finally, the main objective that is the interface between these two modules such that the real robot echoing the movements or changes made in the stimulation part is achieved with the creation of rosnode that converts the rotation into steps necessary to move the robotic arm as desired. Systamizing of objects in automation industries could be consummated by this trajectory.***

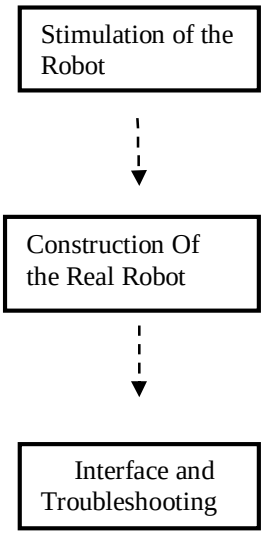
Keywords: Robotics, Computer Numerical Control Programming , 3D printing , ROS .

1. INTRODUCTION

Into the great beyond of robotization and advanced mechanics, the planning of a monetarily proficient robot with different highlights through an open source device turns out to be colossally significant and incredibly legitimate. Despite the fact that substantially demandable Computer Numerical Control (CNC) programming comprising g code and m code for controlling , beginning, halting of robots is utilized as a procedural guidance for 3D printing it needs the space of feedback control. An open source moveo which is an open plan is being animated. Here , It is made to proceed as an amazing hardware surveying multiplex activities. At first start with the development of the robotic arm, the proposed thought manages invigorating, controlling the real robot and the interface between these two. Reproduction of the robot includes the production of URDF document of the moveo utilizing solidworks. Controlling arrangements with the coupling of Aurdino Mega 2560 alongside a RAMPS 1.4 and the stepper motors. Lastly, the fundamental target that is the interface between these two modules with the end goal that the real robot repeating the developments or changes made in the incitement part is accomplished with the formation of rosnode that changed over the rotations into steps important to move the robotic arm as wanted. Systamizing of articles in robotization ventures could be culminated by this direction.

2. SYSTEM ARCHITECHTURE

The framework utilizes Arduino Mega 2560 combined with Ramps 1.4 which fills in as the standard equipment in an open source 3D printer control and which is configuration indicated for moveo and all the joints are impelled with 6 stepper motors. While the equal jaw gripper has gotten under way with servo. The control part of the robotic arm includes the composed Arduino firmware . An outside library called accelstepper is utilized for the movement control of the stepper motors. the constant interface between the incitement and the real robot is finished utilizing ROS . The rostopics hence created host the data to be communicated/gotten, and both the recreation and Arduino firmware can distribute or buy in to those rostopics continuously.



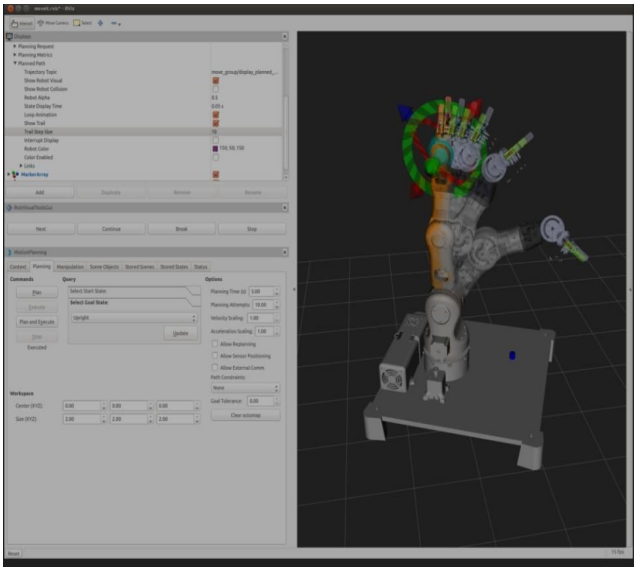
3. METHODOLOGY PROPOSED

3.1 Stimulation Of The Robot

The different investigations till now includes just the beginning, halting , pick and spot of robots The Urdf enumerating the designs, for example, the joints,links, surfaces ,physical constraits and those that important to fabricate a 3D printing in reenactment programming projects, for example, RVIZ and Gazebo is made utilizing solidworks. This urdf arranges the moveo with the animated moviet . Moveit brings movement arranging capacity to the robot in reproduction. Utilizing moveit, it is conceivable to design and execute directions utilizing an assortment of movement arranging calculations. . Ensuring that ROS presented precisely with a working workspace-Ros dynamic on Ubuntu 16.4 including the librarires 'moveo_ros' in the 'src' organizer is utilized.To plan and execute headings for the Moveo in amusement (RVIZ with Moveit module), the going with terminal request is utilized .

roslaunch moveo_moveit_config demo.launch

While the window loads, in the base left corner checking "Grant Approximate IK Solutions." Then snaping on the "Organizing" tab in the Motion Planning leading body of RVIZ and by Selecting another target state by either pulling the natural marker (light blue ball on the end effector) or under "Select Goal State." The Once target state could be invigorated, "Plan and Execute" will structure and execute the heading from the earliest starting point state to the revived target state.



3.2 List Of The Directories

- The moveo_urdf
- moveo_moveit_config
- moveo_moveit

moveit_convert.cpp:

move_group_interface_coor_1.cpp:

3.2.1 About The Directories

Place moveo_urdf
Contains the URDF (Unified Robot Description File) for the BCN3D Moveo. Important for reenactment in RVIZ and moveit design.

moveo_moveit_config

Setup for moveit, a movement arranging structure that has a module in RVIZ, which is the thing that we are utilizing here.

moveo_moveit

moveit_convert.cpp:

Converts reproduction 'joint_state' pivots (from the 'move_group/fake_controller_joint_states' subject) to steps and distributes on the/joint_steps point.
Joint_steps is a variety of 6 Int16 values (however we just have 5 joints for this situation) that speak to the gathered advances executed by each joint since the moveit_convert hub has begun running.

move_group_interface_coor_1.cpp:

Can hardcode a posture/position for the end effector in the content and plan/execute a direction there.
Additionally peruses/yields the current posture/position of the end effector.

3.2.2 Construction And Interface

The real robot which is an open source tool is controlled by Arduino Mega 2560. The actual interface that creates the real time communication between the stimulation and the real robot is done by

Ensuring the download of (AccelStepper Library Download) and ros_lib (rosserial-arduino instructional exercise) libraries into Arduino environment , the event that ros_lib as of now exists in Arduino libraries (/libraries), following the last investigating tip or by getting a command saying "ArmJointState.h: no such document". ROS eliminate ros_lib and recover it each time to present another custom message. Changing the pin design between robot and the RAMPS 1.4 in 'moveo_moveit_arduino.ino' and transferring the record to Arduino was the next step consequently , Ensuring the robot and the recreation are similarly situated (to set the reenactment upstanding at first - select "Upstanding" from "Select Goal States" in RVIZ. In 'moveit_convert.cpp' supplant the stepsPerRevolution cluster with the means/insurgency (or microsteps/upheaval) of every one of the motors.

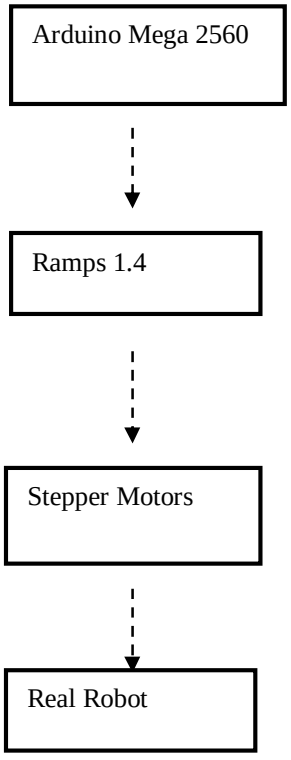
With the reproduction previously running, execution of every one of the accompanying orders in it's own, independent terminal:

rosrun rosserial_python serial_node.py/dev/ttyUSB0(establishes rosserial hub that speaks with Arduino)

rosrun moveo_moveit moveit_convert (changes over recreation joint_state revolutions to steps and distributes on the/joint_steps theme, which the Arduino content buys in to)

rostopic bar gripper_angle std_msgs/UInt16 <angle 0-180> (distributes gripper_angle)

Presently, whatever directions are arranged and executed in reenactment are repeated on the real robot.



4. Troubleshooting

Troubleshooting is a type of critical thinking, frequently applied to fix bombed items or cycles on a machine or a framework. It is an intelligent, precise quest for the wellspring of an issue so as to unravel it, and make the item or cycle operational once more. Troubleshooting is expected to distinguish the indications.

The Troubleshooting is done by following the steps below

/joint_steps: steps important to move each motion to wanted position

/joint_steps_feedback: same as/joint_steps, aside from distributed back by arduino to watch that data is being gotten by Arduino accurately

/gripper_angle: current edge of the gripper

To move Moveo from the order line:

rostopic bar joint_steps moveo_moveit/ArmJointState <Joint1 Joint2 Joint3 Joint4 Joint5 0>

Change "Joint1, Joint2, and so on." to the quantity of steps you need each joint to move.

Using rostopic rundown and quest for these subjects to check in the event that they are presently running could be known.

Using rostopic reverberation/<topic> the information on <topic> is seen

In the event that get the accompanying "blunder: moveo_moveit/ArmJointState.h: No such record or catalog", play out the accompanying strides in terminal:

compact disc <Arduino sketchbook>/libraries

rm - rf ros_lib

rosrun rosserial_arduino make_libraries.py

5. Results And Discussions

Utilizing constant article acknowledgment from a monocular picture (normal webcam) related to this stage, we can perform 'pick and spot' directions that are explicit to the perceived item. With no guarantees, this kind of use could be valuable for arranging objects in modern mechanization. A stage for powerful criticism control would permit to do things like use observation and an IMU to control developments, or use power sensors to control handles – there's an unbelievable measure of potential here. Through this undertaking, By this the moveo is made to be an incredible asset for investigating complex control ideas at the front line of mechanical technology. Automated arm built here could be utilized in the various fields like a family unit, work environment, and working station.

6. Conclusion

Effectively built up the mechanical structure of the automated arm and interfaced with the Aurdino

processor. Created G-codes for different figures and effectively meant regulator recognizable language. In this manner by consolidating the exceptionally adaptable automated arm structure with 3D printing innovation it can expand the printing territory also, can print little bends without any problem. Following up is preparing extra sensors for the mechanical arm, for example, settling on movement control choices dependent on information from monocular pictures, IMU, RGBD sensors, and power sensors.

References

[1] "a robotic arm with lables ," http://i-heart-robots.blogspot.in/, accessed: 2015-08-20.

[2] A. Shirkhodaie, S. Taban, and A. Soni, "Ai assisted multi-arm robotics," in Robotics and Automation. Proceedings. 1987 IEEE International Conference on, vol. 4, Mar 1987, pp. 1672–1676.

[3] Y. Huang, L. Dong, X. Wang, F. Gao, Y. Liu, M. Minami, and T. Asakura, "Development of a new type of machining robot-a new type of driving mechanism," in Intelligent Processing Systems, 1997. ICIPS'97. 1997 IEEE International Conference on, vol. 2. IEEE, 1997, pp. 1256–1259.

[4] Z. Kuijing, C. Pei, and M. Haixia, "Basic pose control algorithm of 5-dof hybrid robotic arm suitable for table tennis robot," in Control Conference (CCC), 2010 29th Chinese. IEEE, 2010, pp. 3728–3733.

[5] A. Bejo, W. Pora, and H. Kunieda, "Development of a 6-axis robotic arm controller implemented on a low-cost microcontroller," in Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on, vol. 1. IEEE, 2009, pp. 328–331.

[6] M. Nakashima, K. Yano, Y. Maruyama, and H. Yakabe, "The hot line work robot system phase ii and its human-robot interface mos," in Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on, vol. 2. IEEE, 1995, pp. 116–123.

[7] V. Potkonjak, S. Tzafestas, D. Kostic, and G. Djordjevic, "Human-like behavior of robot arms: general considerations and the handwriting taskpart i: mathematical description of human-like motion: distributed positioning and virtual fatigue," Robotics and Computer-Integrated Manufacturing, vol. 17, no. 4, pp. 305–315, 2001.

[8] N.E. Chávez Rodríguez and J. Esquivel Villar. (2006). Remote Manipulation of a Robotic Arm by Using Infrared Sensors. Nanotechnology Conference and Trade Show-Nanotech 2006-9thAnnual. Boston, Massachussetts.