БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ • BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, **69**
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, **69**

София • 2018 • Sofia

Control of Robotic Arm Manipulator Using ROS

D. Chikurtev[1, 2], I. Rangelov[2], N. Chivarov[2], E. Markov[2], K. Yovchev[2]

[1]Institute of Information and Communication Technologies – BAS
[2]European Polytechnical University - Pernik

Email: denis@iinf.bas.bg, rangelov.ivaylo@yahoo.com, nshivarov@code.bg, emospy@gmail.com, kyovchev@gmail.com

Abstract: *In this paper, we represent how ROS is using to Control Robotic Arm Manipulator. Our goal is to achive control of simulation model of robotic arm manipulator in the RVIZ environment. Described software in this paper is based on ROS, RVIZ and MoveIt!. ROS is open-source operating system for robots, which provides different services for robots. RVIZ is a 3D visualizer for displaying sensor data and state information from ROS. MoveIt! is software for mobile manipulation, motion planning, kinematics, control and navigation. Conducted experiments test on the reach of the robot arm and researches various control methods for manipulation of the robotic arm in the execution of specified trajectories.*

Keywords: *robotic arm, ROS, RVIZ, MoveIt!, kinematic model, control, motion planing*

1. Introduction

Service robots can became a huge part of the human life because the rapidly ageing population is placing increasing strain on healthcare services [1]. Mobile robots have been proposed as a way to assist people to stay healthy and safe in their own homes [2]. They can be used to help elder and disabled people with daily works, serve drugs, foods and drinks [3]. To provide this type of services, we are using Robot Operating System (ROS). ROS [4] is one of most popular operating system for controlling robots. ROS provides standard operating system services such as hardware abstraction, low-level device control, implementation of commonly used functionality, message passing between processes, and package management.

Robotic arm is visualize and displaying by RVIZ. In RVIZ, you can visualize current configuration on a virtual model of the robot. You can also display live representations of sensor values coming over ROS Topics including camera data, infrared distance measurements, sonar data, and more.

For control robotic arm manipulator in ROS, we are using MoveIt [5]. It provides an easy-to-use platform for developing advanced robotics applications, evaluating new robot designs and building integrated robotics products for industrial, commercial and other domains. The robotic arm in this paper is Mover4. The Mover4 robot arm allow replaying automation scenarios close to reality, it can be used as a motion platform and connects physics. Those types of robotic arm usually are mounted on top of mobile robot platforms.

The anthropomorphic arm should be able to perform complex movements on a given trajectory and reach places. Various virtual experiments will be conducted to show the reach of the hand and test joystick and automate control of robotic arm.

2. Methods

2.1. ROS

Robot Operating System (ROS) is a collection of software frameworks for robot software development, providing operating system-like functionality on a heterogeneous computer cluster. ROS provides standard operating system services such as hardware abstraction, low-level device control, implementation of commonly used functionality, message passing between processes, and package management. Running sets of ROS-based processes are represented in a graph architecture where processing takes place in nodes that may receive, post and multiplex sensor, control, state, planning, actuator and other messages. Despite the importance of reactivity and low latency in robot control, ROS, itself, is not a real-time OS (RTOS), though it is possible to integrate ROS with real-time code.

In order to understand how the ROS System works, we need to have a high-level understanding of the ROS Framework. ROS is based upon distributed computing with the “node” as the basic computational entity. Each ROS system is composed of one or more nodes. Nodes are useful because they communicate with other nodes. Services provide node with a client/server mechanism to communicate with another node.

Topics were used in nodes we wrote for this project. Topics provide a communication mechanism to send a message from one node to many other nodes. The node that sends the message is called the Publisher. There is only one Publisher node for a given topic. The nodes that receive the message are Subscribers to the Topic.

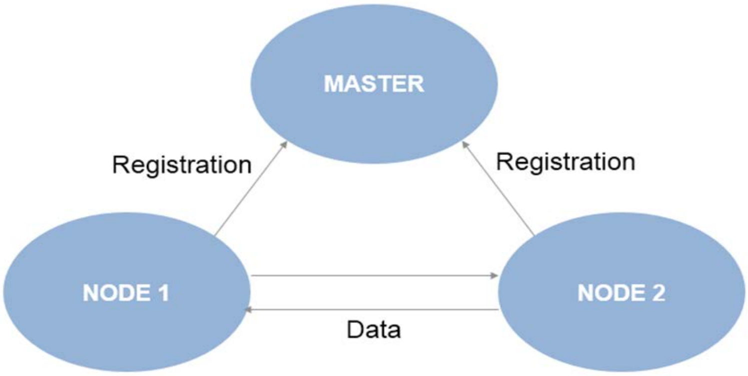

Figure 1. ROS communication.

2.2. RVIZ

Visualization tools have always been some of the most popular features of robot operating systems (ROS). The widely used Robot Visualizer (RVIZ) [6] is capable of rendering three-dimensional (3-D) data ranging from actuated robot models to sensed images and point clouds. This way, it provides much needed insight into an application's state and view of the world.

Built-in Display Types in RVIZ

Name	**Description**	**Messages Used**
Axes	Displays a set of Axes	
Effort	Shows the effort being put into each revolute joint of a robot.	sensor_msgs/JointStates
Camera	Creates a new rendering window from the perspective of a camera, and overlays the image on top of it.	sensor_msgs/Image, sensor_msgs/CameraInfo
Grid	Displays a 2D or 3D grid along a plane	
Grid Cells	Draws cells from a grid, usually obstacles from a costmap from the navigation stack.	nav_msgs/GridCells
Laser Scan	Shows data from a laser scan, with different options for rendering modes, accumulation, etc.	sensor_msgs/LaserScan
Map	Displays a map on the ground plane.	nav_msgs/OccupancyGrid

Markers	Allows programmers to display arbitrary primitive shapes through a topic	visualization_msgs/Marker, visualization_msgs/MarkerArray
Path	Shows a path from the navigation stack.	nav_msgs/Path
Pose	Draws a pose as either an arrow or axes.	geometry_msgs/PoseStamped
Pose Array	Draws a "cloud" of arrows, one for each pose in a pose array	geometry_msgs/PoseArray
Range	Displays cones representing range measurements from sonar or IR range sensors.	sensor_msgs/Range
RobotModel	Shows a visual representation of a robot in the correct pose (as defined by the current TF transforms).	

2.3. The MoveIt!

MoveIt! [7] is a set of software packages integrated with the Robot Operating System (ROS) and designed specifically to provide such capabilities, especially for mobile manipulation. MoveIt! will allow robots to build up a representation of their environment using data fused from three-dimensional(3-D)and other sensors, generate motion plans that effectively and safely move the robot around in the environment, and execute the motion plans while constantly monitoring the environment for changes.

MoveIt! comes with a plugin for the ROS Visualizer (RVIZ). The plugin allows you to setup scenes in which the robot will work, generate plans, visualize the output and interact directly with a visualized robot. The robot package (cpr_mover) and the RVIZ plugin (cpr_rviz_plugin) allow to include the Mover4 robot into an ROS [8] environment, also provide control for Mover4.

The MoveIt! Setup Assistant is a graphical user interface for configuring any robot for use with MoveIt!. Its primary function is generating a Semantic Robot Description Format (SRDF) file for your robot. Additionally, it generates other necessary configuration files for use with the MoveIt! pipeline.

2.4. Kinematics

The study and understanding of Kinematics is a tool in both industrial and mobile Robotics. A robot, to perform most applications needs to process positional data and transform data from one frame of reference to another. Robots have sensors, links and actuators each with its own frame of reference, so transformations between reference frames can be quite tedious.

Forward Kinematics determines the position of the Robot given the joint rotations or distance for Prismatic sliding joints. Inverse Kinematics is more difficult than Forward Kinematics since we need to find one or more ways to move a Robot to a given point in space. Where there is usually a solution in Forward Kinematic problems, there may be multiple or no solutions in an Inverse Kinematic problem [9]. Therefore, we can calculate the Kinematics by using ROS [10].

3. Robot manipulator Mover4

3.1. Robot description

The Commonplace Robotics Mover4 is a four-axis robot for the use in education, entertainment and research environments. The robot can lift a payload up to 500 g with a reach of 500 mm plus gripper.

Figure 2. Mover4 and the necessary components

The modular set up of the robot connects four joint modules with aluminum profiles. Each joint module contains a servomotor.

The front end of the robot, the flange, allows mounting a gripper or another tool. The robot base has to be fastened on a stand or a table. At the robot base one plug connect with power and the PC, another provides digital IO access.

The cable loom is connected with a 12V/5A DC power supply. The USB adapter connects the control PC with the robots internal communication bus. The CPRog software or cpr_mover allows to control and program the robot.

3.2. Robot specifications

Nr of Joints	4 servo joints
Power Supply	12V DC at max 5 A
Power Consumption	Paused: 0,5 A / in motion: < 2.5 A. Fuse with 2.5 A in base

Communication	CAN at 500 kBit/s
Reach	455 mm plus gripper
Payload	500 g
Inputs / Outputs	At the base of the standard version: 3 digital outputs (5V / 25 mA) and 4 digital Inputs (5V) via D-Sub 9 poles female. or, with 24V-DIO Extension: 3 relay out (24V/1A) and 4 digital in (24V) via D-Sub 9 poles male. At the flange: 2 digital Outputs (5V/ 25 mA) und 12V/0,5A supply via Harting SEK 6 poles.
Communication	Definition of position setpoints for all four joints with 20 Hz cycle. Reading of the current position and the motor current.

3.3. Robot geometry

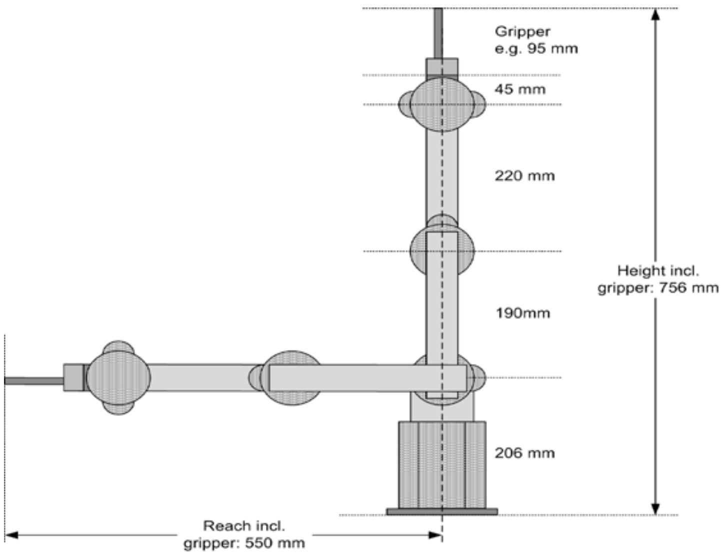

Figure 3. Mover4 simplified side view

The picture shows the joints length of the Mover4 robot arm. Including the gripper the arm reaches 550 mm.

4. Control of Mover4

4.1. Button control

Teleop comes with the cpr_mover package and allows you to command the robotic arm through the button menu (Fig. 4). The buttons allow control of the robotic arm in different directions. Each button is responsible for a different axis and two separate gripper buttons.

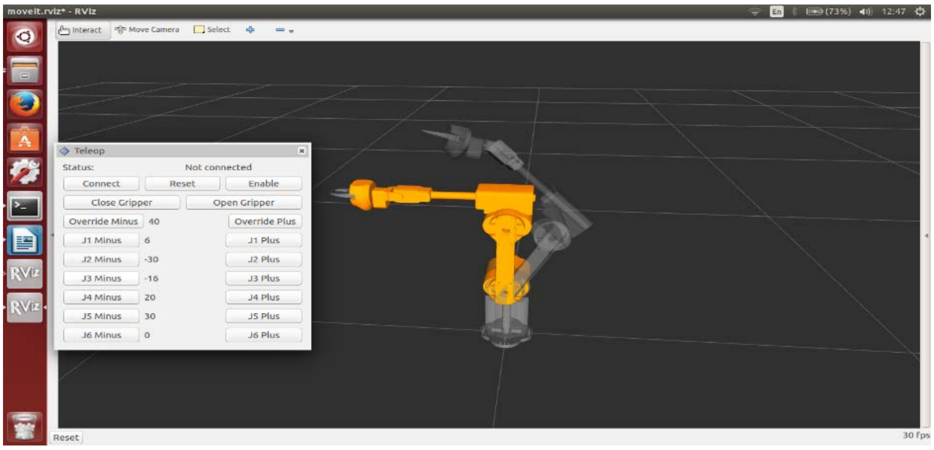

Figure 4. Teleop Control.

4.2. Control via joystick

With cpr_moveit package gives option to control robotic arm with Joystick.

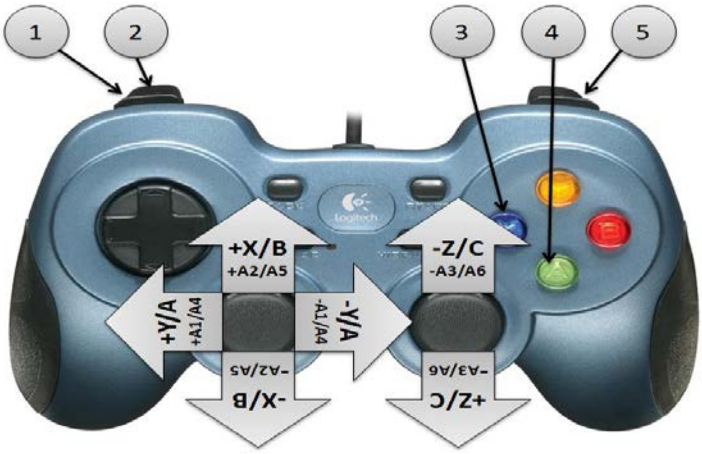

Figure 5. Joystick Button numbering.

Joystick Command Mappings

Command	PS3 Controller	Xbox Controller
+-x/y	left analog stick	left analog stick
+-z	L2/R2	LT/RT
+-yaw	L1/R1	LB/RB
+-roll	left/right	left/right
+-pitch	up/down	up/down
change planning group	select/start	Y/A
change end effector	triangle/cross	back/start
plan	square	X
execute	circle	B

4.3. Trajectory control

The joint trajectory action is a node that provides an action interface for tracking trajectory execution. It passes trajectory goals to the controller, and reports success when they have finished executing. The joint trajectory action can also enforce constraints on the trajectory, and abort trajectory execution when the constraints are violated. The robot follows the planned trajectory in simulation.

5. Experiments and results

For the first experiment we use example image with two people and the task for the vision system is to detect the faces and face detection from video stream. By default, in the opencv libraries the detects faces are marked with blue square and the eyes are marked with geen squares.

6. Conclusion

In this paper we describe the properties, functions and application if RVIZ and MoveIt in ROS. MoveIt and RVIZ - provides an easy-to-use platform for developing advanced robotic applications, evaluating new robot projects, and building integrated robotic products. The results show that various methods – which have been described in this paper can be successfully used to control a robotic arm using ROS.

Acknowledgments

This research was carry out as part of the project “Telecontrolled Service Robots for Increasing the Quality of Life of Elderly and Disabled, № DN 07/23 – 15.12.2016”, financed by the Bulgarian National Science Fund.

References

1. Chivarov, N.; Shivarov, S.; Yovchev, K.; Chikurtev, D.; Shivarov, N. - Intelligent Modular Service Mobile Robot ROBCO 12 for Elderly and Disabled Persons Care; Robotics in Alpe-Adria-Danube Region (RAAD), 2014 23rd International Conference on, Smolenice, Slovakia, 3-5 Sept. 2014, Print ISBN: 978-1-4799-6797-1. p. 343-348.
2. Nayden Chivarov, Denis Chikurtev, Kaloyan Yovchev, Stefan Shivarov - Cost-Oriented Mobile Robot Assistant for Disabled Care; TECIS 2015, 16th IFAC Conference on Technology, Culture and International Stability, Sozopol, Bulgaria, 24-27 September 2015
3. Ulrich Reiser, Christian Connette, Jan Fischer, Jens Kubacki, Alexander Bubeck, Florian Weisshardt, Theo Jacobs, Christopher Parlitz, Martin H¨agele, Alexander Verl, “Care-O-bot 3 - Creating a product vision for service robot applications by integrating design and technology”, The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, 2009 St. Louis, USA
4. Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Ng, ” ROS: an open-source Robot Operating System”
5. MoveIt website: http://moveit.ros.org/ Accessed 20 046 2017
6. Thomas Linner, Alaguraj Shrikathiresan, Maxim Vetrenko, Bernhard Ellmann, and Thomas Bock “MODELING AND OPERATING ROBOTIC ENVIRONMENTS USING GAZEBO/ROS”
7. Sachin Chitta, Ioan Sucan, and Steve Cousins, MoveIt!, IEEE ROBOTICS & AUTOMATION MAGAZINE, March 2012,Pages 18-19
8. Rosen Diankov, James Kuffner, OpenRAVE: A Planning Architecture for Autonomous Robotics, CMU-RI-TR-08-34, July 2008
9. Patrick Beeson and Barrett Ames, TRAC-IK: An Open-Source Library for Improved Solving of Generic Inverse Kinematics. Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on, November 2015, Seoul, South Korea, ISBN: 978-1-4799-6885-5
10. Asad Yousuf, William Lehman, Mohamad A. Mustafa, Mir M Hayder, Introducing Kinematics with Robot Operating System (ROS). ASEE Annual Conference & Exposition, June 2015, Seattle, WA, USA

Управление роботизированным манипулятором с помощью ROS

Денис Чикуртев[1, 2], Ивайло Рангелов[2], Найден Шиваров[2], Емануил Марков[2], Калоян Йовчев[2].

[1]Institute of Information and Communication Technologies – BAS
[2]European Polytechnical University - Pernik

Email: denis@iinf.bas.bg, rangelov.ivaylo@yahoo.com, nshivarov@code.bg, emospy@gmail.com, kyovchev@gmail.com

Аннотация: *В этой статье мы представляем, как ROS использует для управления роботизированным манипулятором. Наша цель - добиться контроля над имитационной моделью манипулятора роботизированного манипулятора в среде RVIZ. Описанное программное обеспечение в этой статье основано на ROS, RVIZ и MoveIt !. ROS - это операционная система с открытым исходным кодом для роботов, которая предоставляет различные услуги для роботов. RVIZ - это 3D-визуализатор для отображения данных датчиков и информации о состоянии от ROS. MoveIt! это программное обеспечение для мобильных манипуляций, планирования движения, кинематики, управления и навигации. Проведенные эксперименты проводят проверку на расстоянии от руки робота и исследуют различные методы управления манипуляцией роботизированной рукой при выполнении заданных траекторий.*

Ключевые слова: *роботизированная рука, ROS, RVIZ, MoveIt !, кинематическая модель, управление, планирование движения*