Development of a Virtual Teaching Pendant System for Serial Robots based on ROS-I

Luping Chen[1], Zhongqi Wei[2], Fei Zhao[2*], Tao Tao[2]
China Software Testing Center
Shanxi Key Laboratory of Intelligence Robot, Xi'an Jiaotong University, China
E-mail: ztzhao@mail.xjtu.edu.cn

Abstract: The aim of this paper was to present a virtual teaching pendant system for serial industrial robots. Based on Robot Operating System Industrial (ROS-I), the virtual teaching pendant system we designed has the capacity for robot model establishment, direct and inverse kinematics manipulation, as well as motion planning. This virtual teaching pendant system not only contains the main teaching pendant functions but can drive any serial robots in both simulation and real-world. This paper will also highlight the many advantages of this virtual teaching pendant system which can be used as a good teaching tool to help students in understanding robotics. Several demonstrations are made to show the universality and functionality of the virtual teaching pendant system.

Keywords: Virtual Teaching Pendant System, ROS-I, Moveit!, GUI teaching pendant

I. INTRODUCTIONS

A teaching pendant is a portable device that is employed to manipulate the robot arm. It can perform basic functions such as manual control, teaching and play back, programming robot and processing robot emergency. Apart from controlling the motion of robot manually, the teaching pendant can be used to program and provide interpolating algorithm to perform individual or a sequence of combined robot movements.

Many industrial robot corporations have produced various on-chip teaching pendants. For the studies of teaching pendants, they focus largely on new functions such as new ways to teach, playback and manipulate robot arm safely [1], [2]. The various functions are accessed by the available buttons on the on-chip teaching pendant. The buttons have different functions such as menu keys, function keys, motion control keys etc. However, those buttons are packed in a limited area and the functions of traditional on-chip teaching pendant are hard to extend. The on-chip teaching pendants are generally pretty heavy which greatly reduce the degrees of user experience. Furthermore, different corporations have produced various teaching pendants with different communication protocols and robot programming languages, which causes irreplaceable to the teaching pendants between different types of industrial robots.

In this paper, we present a virtual teaching pendant system with the common functions of pendants based on Robot Operational System Industrial (ROS-I). Unlike the traditional on-chip teaching pendants, function keys on GUI teaching pendant need not to limited to one area and teaching function can be extended easily. GUI teaching pendant was developed on a cross-platform SDK (Qt creator) which means the GUI pendant codes could be implemented on the portable devices such as Ipads, hand-phones, or Laptops. By ROS-I packages, the virtual teaching pendant system can not only manipulate serial robot models in simulation but also is capable of communicating the multi-axis controller to control the real robot. This industrial control system supports many common manipulations such as ABB, Fanuc, Universal Robots etc [3]. The user can also use their multi-axis controller to connect the virtual pendant system which needs to support the TCP/IP protocol.

II. TECHNICAL BACKGROUND

A.ROS-I

Robot Operating System Industrial (ROS-I) is an open source robotic operation that extends the advanced capabilities of ROS to new industrial applications. Based on ROS framework, we can access the unified communication protocol and programming language such as C++, Python. ROS-I provides the user with an overall framework of industrial robot control system. The framework of industrial robot control system was composed of GUI layer, application layer, interface layer, simple message layer and motion controller layer [4],[5]. The functions of every layer and adopted solution are shown in Tab.1. Each layer needs to be developed to meet the required functions of industrial robot control system.

B. Moveit!

Moveit! is a state-of-the-art software for kinematics based manipulations [6], [7]. It provides an easy-to-use platform for developing advanced robotics applications. Based on the OMPL (The Open Motion Planning Library) from Moveit!, we can apply many state-of-the-art sampling-based motion planning algorithms to our system. Building a URDF (The Unified Robot Description Format), MoveIt! can generate a kinematic model for any serial-chain manipulator on which many kinematic model based algorithms could be implemented. The user also can establish the workspace by adding a set of collision objects or the workpiece in the "world" part of the planning scene module. When the motion planning starts, all the plans can be visualized by the

978-1-5386-3135-5/17/$31.00 ©2017 IEEE

Rviz(a graphic user interface of Moveit!).

C. The Multi-axis Motion Controller

The virtual teaching pendant system is designed to work with the multi-axis motion controller to execute the joint trajectories. The proposed controller should be able to carry out the tasks of point-point motions and trajectory motions in the current coordinate system. It is worth to notice that ROS-I packages provide the communication interfaces for many industrial robots controllers. The user can also design a multi-axis motion controller with TCP/IP communication interface to connect the virtual teaching pendant system.

Based on the overall framework of ROS-I, the software architecture of the whole virtual teaching pendant system is shown in Fig 1. The GUI pendant can generate a set of robot motion instruction and send commands to the Moveit!. By using MoveIt!, we can apply the inverse kinematics and motion planning algorithms to generate joint trajectories for the given tasks. With ROS-I helper (industrial_robot_client), the Moveit! interface can send the joint trajectories command and receive the feedback from the multi-axis motion controller through the TCP/IP protocol [8]. With this designer, many different types of serial robots can be modeled by Moveit! and manipulated by GUI teaching pendant.

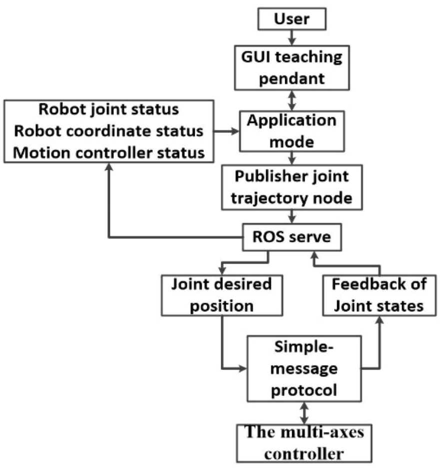

Fig.1 Software Structure of the Virtual Teaching Pendant System

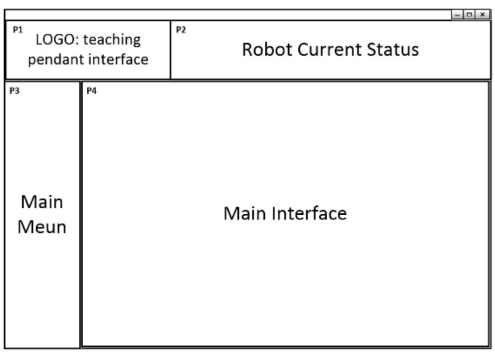

Fig.2 The Outline of virtual GUI teaching pendant

III. DESIGN AND IMPLEMENTATION OF THE VIRTUAL TEACHING PENDANT SYSTEM

Qt-creator was used to design the GUI teaching pendants on Linux using C++. Qt-creator contains many widgets which can be used in designing the GUI, such as Buttons, Contains, Labels, and Item Widgets etc. Fig. 2 shows the outline of the GUI teaching pendant. The Background in Fig 2 is divided into four parts P1, P2, P3, and P4. P1 was the logo part. The second part P2 was the robot status columns, which show the real-time display of robot current status. The third part P3 was the main menu which was divided to several buttons for the change of different modes. The last part P4 was the main interface of the pendant to show the manipulator status, the joint speed, and the current coordinate systems etc.

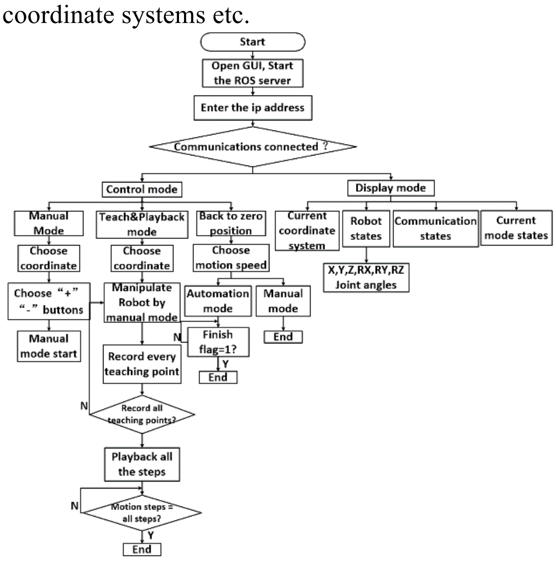

Fig.3 Software Architecture of GUI pendants

A. The Software System of GUI teaching pendant

The software of GUI pendant focuses on the functionality and user-friendliness. It is required to monitor and response the user input, deal with communication with ROS server and display the GUI to meet user needs. The overall software architecture of GUI pendant was shown in Fig. 3. It was designed to three types of control modes and four types of display modes. The control modes include the manual mode, teach and playback mode and back to zero mode. The display modes include the current coordinate systems, the robot states, communication states and the current control modes. Each mode has different threads in case GUI pendant gets stuck to influence the user experience.

1) Access Authentication Interface: Fig.4(a) show the access authentication interface of GUI teaching pendant. The Access authentication interface starts immediately after the application started, prompting users to enter ROS server's IP address and establishing the connection between ROS server and GUI pendant. Once the connection success, the manipulator status changes to "green"

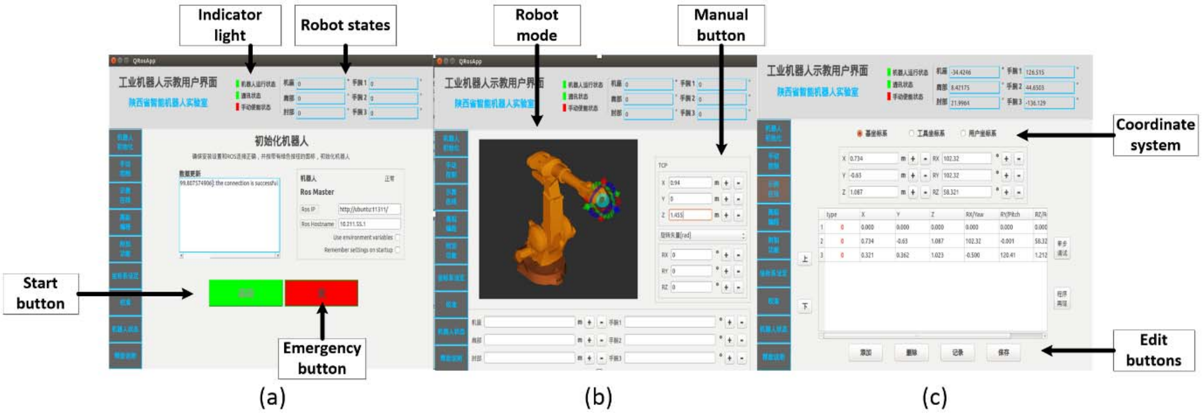

Fig.4. User Graphic Interface

Layers	Functions	Adopted Solution
1.GUI Layer	GUI	GUI Teaching Pendant
2.Application Layer	Motion Planning/Kinematics Providing Advanced Algorithm	Moveit! OMPL/URDF
3.Interface Layer	Communication Interface	Moveit!
4.Simple Message Layer	Communication Protocol	Industrial_robot_client
5.Motion Controller Layer	Manipulate the real robot	Motion Controller

Tab.1 Layer Functions and Adopted Solution

and the robot current status column display the six joint angles in degrees. Failing to connect ROS server, the pendant will remain unchanged: the manipulator status textbox will display as “connection failed” and all other function keys remain disabled.

2) Manual Control: After a successful connection, users can select one of the following operation-modes: Manual-Mode, Teach and Playback Mode and Back to zero Mode. Each mode is activated by clicking on the relevant Tab on the main menu panel. Fig.4(b) show the Manual-Mode panel of GUI pendant. The user can move specific joints in the manipulator with the corresponding control button. The panel includes six buttons for the six different joints, namely: Base, Lower-Arm, Upper-Arm, Wrist-Roll, Wrist-Pitch, and Wrist-Twist, which are named to match the relevant axis. The Manual-Mode panel also provides a set of predefined coordinate systems including the joint coordinate system, base coordinate system, and tool coordinate system. The user can choose one of the coordinate systems or established their own coordinate system for the robot end-effector execution. Textboxes are attached to each button to display the angles in degrees or radians, which could be incremented/decremented through the “+” and “-” buttons. Once entered, the corresponding joint of the arm will move by that angle amount. Each joint has the limited range regions that are defined by the manufacturer. Wrong entries will result in an “Out of Range Error Message”.

3) Teach & Playback: The Teach and Playback Mode is the main function of GUI pendant which is shown in Fig 4(c). In GUI teaching pendant, teach and playback function can be carried out with four steps: manual control, record, edit and playback. First, the user needs to manipulator the robot to teaching point by the manual control mode. After recording the teaching points by the “save” button, the user can edit all the teaching points by adding a new teaching point, insert a new teaching point and delete a teaching point. Playback program is created when the editing is finished. The user can choose the single step playback or all teaching points playback in playback program.

B. Communication with ROS server

Every command from GUI needs to send to ROS server by the specified communication protocol in ROS. “Node” is the ROS term for an executable that is connected to the ROS network. Every node is required to connect the corresponding topic to transmit information. By the publisher and subscriber nodes, a set of robot commands can be broadcasted. Meanwhile, corresponding information can be received from ROS server. The software architecture of the communication between ROS server and GUI pendant is shown in Fig.5.

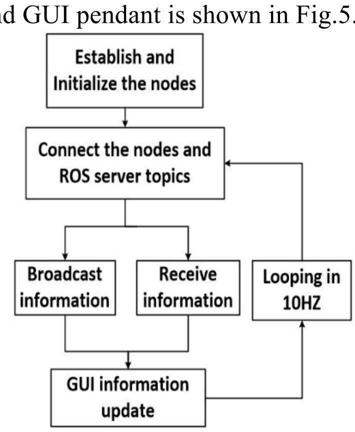

Fig.5.Software Architecture of the Communication

C. Coordinate System:

GUI pendant provides a set of coordinate systems including Joint Base, and Tool to define the coordinate system. Except for the defined coordinate systems, the user can also establish their own coordinate systems when user need to set up a new reference frame to simplify computation. We adopted TF library to implement this function. The TF library is a package from ROS which can let user transform points or vectors between any two coordinate frames at any desired coordinate in time [9]. By TF library, we are able to choose the desired coordinates to define current the coordinate system.

D. Motion Planning

GUI teaching pendant was designed base on Moveit!. Running on ROS server, Moveit! is responsible for receiving the command from GUI teaching pendants and supporting the motion planning based on the kinematics and OMPL library. As shown in Fig.6, the user can manipulate robot model and observe the process of motion planning from the user-interface Rviz. OMPL is a powerful collection of state-of-the-art sampling-based motion planning algorithms. The user can choose the proposed motion planning algorithm from OMPL module to deal with motion planning. After motion planning by Moveit!, the result of motion planning will be displayed in the terminal when motion planning is finished. The computational results include each joint angles, joint angular velocity, joint angular acceleration and duration times which are shown in Fig. 7. The computation time for simulation depends on the path length of the end-effort and the complexity of path. Every single computational result will be updated in every 10 milliseconds.

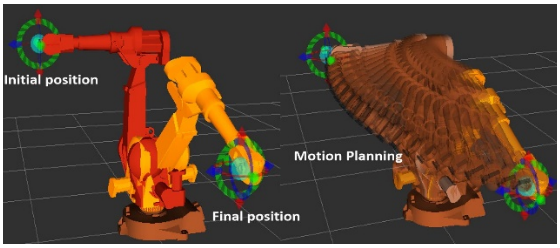

Fig.6 The User Interface of Moveit!

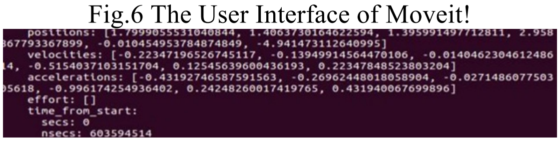

Fig.7 The Computational Results of Motion Planning

E. Robot Multi-Axis Controller

The computational results of motion planning need to be uploaded to the real industrial robot for execution. The first step is to establish the communication between ROS server and multi-axis controller. In ROS server part, the communication carried out by the industrial_robot_client, a package from ROS-I for implementing the interface to the multi-axis controller. In the multi-axis controller part, the multi-axis need to open the standard TCP/IP interface for receiving the results of motion planning from ROS server.

The common industrial robot companies like ABB, Fanuc, Motoman, and Universal Robots provide the robot motion controllers which have the corresponding packages for communication with ROS [10]. There are available on the website (http://wiki.ros.org/Industrial). For user multi-axis controller, the user multi-axes controller needs not only have TCP/IP interface for communication but also meet real robot point to point motion. Based on the multi-axis controller, the virtual teaching pendant is able to manipulate serial robot in real-world.

F. Configuration Steps for a New Robot

Before using the virtual teaching pendant system to control a new robot, the user needs to configure serial robot control system first. The following four steps should be carried out to configure the robot control system for a new robot.

1) Establish the virtual robot model using URDF.
2) Generate MoveIt! scripts using MoveIt! Setup Assistant.
3) Configure multi-axes controller to connect to the Moveit!. Modify corresponding settings in the interface.
4) Build the communication links between GUI teaching pendant and Moveit!.

IV. DEMONSTRATIONS

The virtual teaching pendant system is tested on several robots in simulation to show the universality and functional superiority of that.

It is really useful that the teaching pendant can control the virtual robot models because manipulating the real industrial robot is a complex and time-consuming task that requires a considerable amount of work-related skills, robotics knowledge, and real robot. Especially for the robot new learner, this function is a safe and effective way to learn basic robotics and manipulate the industrial robots.

Based on the ROS-I framework, the virtual teaching pendant system is capable of manipulating the virtual serial robot models which can be displayed and controlled simultaneously in user-interface Rviz from Moveit!. ROS-I provide the source codes of various virtual industrial robots models which make it easy to prepare the tests of the virtual teaching pendant system using for different types of serial robots. As shown in Fig.8, it achieved to control the serial robot model of a UR5 robot model, an ABB IRB2400 robot model in simulation.

Through the planning scene module from Moveit!, our system supports the offline programming method to program robot. We can establish a demo workspace by adding several objects and the workpiece, as shown in Fig.9. In the demo workspace, we can acquire exactly the position of the workpiece which is shown in Moveit! interface. Based on the workpiece position, GUI teaching pendant was manipulating Kawasaki RA10N industrial robot to grab the workpiece and move it to the desired position. The results of motion planning are then uploaded to the robot controller for real robot execution.

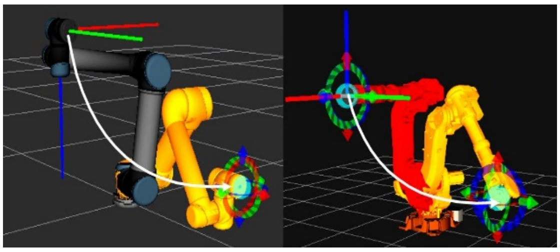

Fig.8 UR5 Robot and ABB IRB246700 Robot

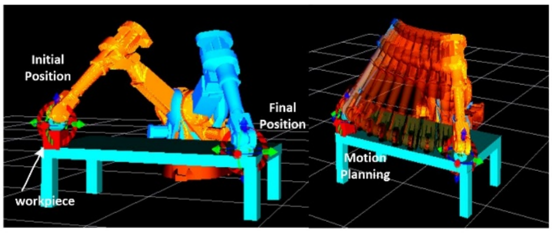

Fig.9 Kawasaki RA10N Robot

V.CONCLUSION

In this paper, we presented a virtual teaching pendant system to manipulate serial robot in both simulation and real-world based on ROS-I. The design goal is to integrate ROS-I advantages to design GUI pendant which not only stipulates the whole industrial robot control system framework but also provide a set of useful industrial robot applications such as Moveit!. Three contributions are made in the pendants. 1) We designed a software GUI teaching pendant which implemented the essential functions of teaching pendants such as manual control, teach & playback. 2) Based on Moveit!, GUI teaching pendant is capable of manipulating many virtual robot models in the simulation which has proven to be a good teaching tool to help students in understanding robotics. 3) Through the communication between ROS server and multi-axis controller, the virtual teaching pendant system is able to control many different types of real serial robots.

Look forward, we envision several improvements that would increase the utility of GUI teaching pendants. ROS is an opening robot operating system which provides many applications in robot design [11]. Based on ROS, we can further expand the functions of the virtual teaching pendant system in many aspects such as programming module, algorithm module, and robotic vision module.

REFERENCES

[1] Kaluarachchi, Malaka Miyuranga, and F. Y. Annaz. "GUI Teaching Pendant Development for a 6 Axis Articulated Robot." 330(2012):111-118.

[2] Annaz, Fawaz, and M. Kaluarachchi. "Development of a Soft Teaching Pendant for a Six-Axis Manipulator." Intl Tax & Bus.law 10.2(2014):3295-3300.

[3] Michieletto, Stefano, et al. "ROS-I Interface for COMAU Robots." (2014).

[4] Gao, Yongzhuo, et al. "U-Pendant: A universal teach pendant for serial robots based on ROS." IEEE International Conference on Robotics and Biomimetics IEEE, 2014:2529-2534.

[5] ROS-I, Available online https://github.com/ros-industrial/.

[6] S. Chitta, I. Sucan, and S. Cousins, “MoveIt!,” IEEE Robotics & Automation Magazine, vol. 19, no. 1, pp. 18–19, 2012.

[7] Moveit!, Available online http://moveit.ros.org.

[8] Su Y, Gao Y, Wu Y, et al. “YARC — A universal kinematic controller for serial robots based on PMAC and MoveIt!”Asia-Pacific Signal and Information Processing Association, 2014 Summit and Conference. IEEE, 2015:1-6.

[9] TF, Available online “http://wiki.ros.org/tf/”.

[10] Michieletto, Stefano, et al. "ROS-I Interface for COMAU Robots." (2014).

[11] Cousins, Steve, et al. "Sharing Software with ROS." IEEE Robotics & Amp Amp Automation Magazine 17.2(2010):12-14.