Simulation of a Tele-operated Task under Human-Robot Shared Control

Longjiang Zhou, Keng Peng Tee and Zhiyong Huang
A*Star Institute for Infocomm Research
Singapore
{zhoul, kptee, zyhuang}@i2r.a-star.edu.sg

ABSTRACT

This poster presents simulation of a tele-operated shared controlled robot task that is integrated with a generic simulator of RADOE (Robot Application Development and Operating Environment). A customized and extendable Rviz interface plugin is designed and applied to import models, do s imulation, enable real robot operation, and communicate with other projects by clicking related buttons. In the simulation process, the robot model in the simulator is controlled and visualized by human operator using an Omega 7 haptic device and automatic method, i.e., the shared control. After the simulation is conducted and satisfied, the system will send back a signal to the real robot system to execute the operation task; otherwise, simulation of the shared control process will be continued until satisfaction. We provide a simulation of a drawing task on the surface of a sphere.

Author Keywords

Robotics simulation; end-effectors; teleoperation motion planning.

ACM Classification Keywords

I.6.m. Simulation and Modeling: Miscellaneous.

INTRODUCTION

In application tasks with uncertainties in the pre-programmed settings (e.g. work piece location, size, orientation), it is difficult to achieve reliable robot operation under full autonomy, due to technical challenges in robotic perception, motion planning or control in the presence of uncertainties. On the other hand, humans are more adept in perception and cognition than robots, and can easily spot errors during operation. Human-robot shared control systems allow for more complex operations to be performed reliably due to complementary proficiencies between human and robot [1][2][3]. This approach is particularly useful for tele-operation tasks in which impoverished human depth perception of the remote environment can be compensated by the robot's local sensing and automatic surface tracking behavior [4]. For example, in a task of drawing on a curved surface, the robot automatically controls the distance and orientation with respect to the surface, while the human performs the drawing motion easily as if it is on a flat plane.

HAI '16 , October 04-07, 2016, Biopolis, Singapore
ACM 978-1-4503-4508-8/16/10.
http://dx.doi.org/10.1145/2974804.2983316

To reduce the development time and cost, robotics simulators are used widely to simulate the behaviours of components and control programs so as to improve operation performance and robustness of the robots, especially for robots applied in special areas [5][6]. Of these software, commercial simulators [7][8] generally have high quality and reliability, but they increase the economic burdens to ordinary users. In comparison, open-source simulators [9] [10] decrease the development time and cost by providing an open and free communication platform for researchers. ROS (Robot Operating System) [11] is an open-source simulator which applied Rviz as its visualization tool and it also integrated Gazebo [12] for modelling and physical simulation of rigid bodies. However, the Gazebo was separated from ROS from Gazebo 1.9. So our team has developed the RADOE [13], a generic industrial robotics simulation platform with a user-friendly interface, to makeup this gap.

With respect to simulating tele-operated shared control tasks, the challenge is to incorporate another platform with both human and automatic control strategies, running on a different operating system. In our case, the tele-operated shared control drawing task is running on Microsoft Windows, and the RADOE simulation is running on Linux Ubuntu. We solved it by designing and implementing a wrapper, a software API based on TCP/IP protocol [15]. Through the wrapper, the trajectory of robot by tele-operated shared control is transmitted to RADOE simulation platform.

This poster introduces RADOE software and its integration with a tele-operated, human-robot shared controlled robotic drawing task [4] to show how the RADOE is applied in the development of industrial robotics simulation.

OVERALL STRUCTURE OF SIMULATION SYSTEM

The RADOE simulator has an overall structure as shown in Figure 1. The RADOE software is installed in a central PC, which may have multiple robots attached to it. The RADOE PC performs task definition and monitoring, which may require inputs from the environment sensors connected to it. It then performs the operation task by sending the required robot motion commands to the robots through the corresponding robot motion controller.

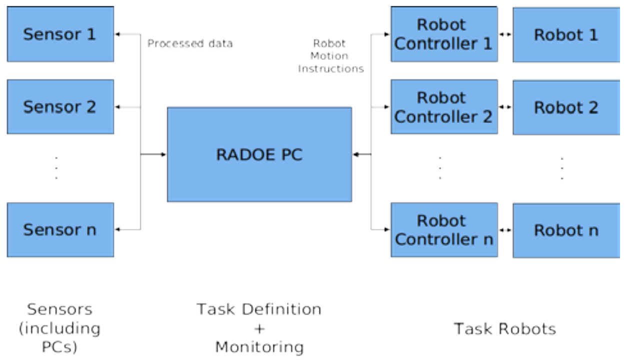


Figure 1. System structure of the RADOE simulator.

The framework of the shared controlled robot drawing system under study in this poster is shown in Figure 2. The human operator uses an Omega 7 haptic device to tele-operate the writing task of the Kuka robot on the surface of a flat plate. The writing procedure and results are displayed on the screen with Windows system to guide the robot motion control through the haptic device.

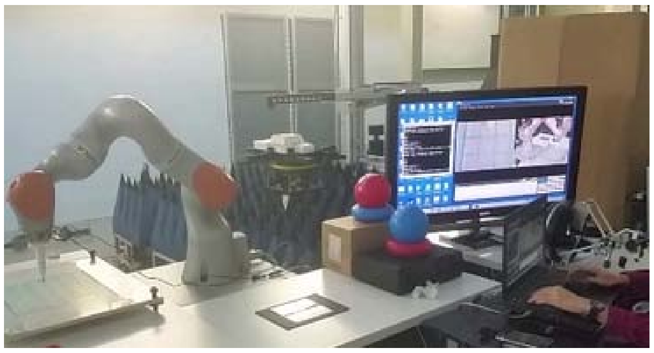
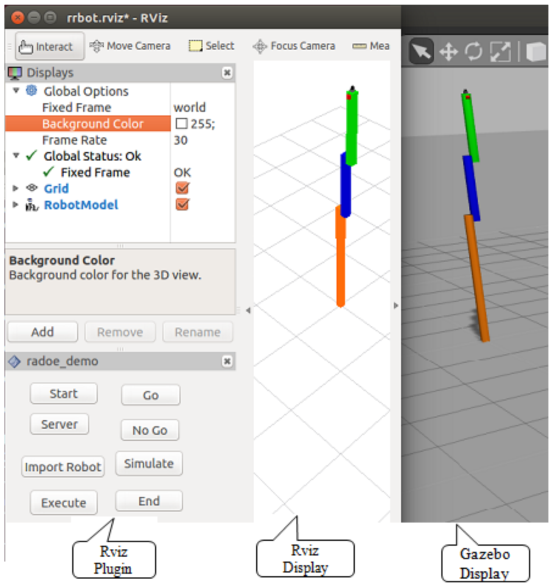


Figure 2. Structure of physical robot drawing system.

DISPLAY AND SIMULATION PROCEDURE OF RADOE

The framework of the RADOE simulator is shown in Figure 3. There are three regions of the simulator. The right hand side is the Gazebo display, the middle side is the Rviz display, and the left hand side is the "Displays" panel of the Rviz. For this panel, the upper part is the original settings provided by the Rviz software, and the lower part is a customized Rviz plugin interface developed by ourselves for the purpose of the specific project. If the "Import Robot" button is clicked, the user can freely select and import the robot into the simulation platform. The example in this figure is "rrbot", a simple 3-linked robot. The user can also click the "Import Environment" button to import some proper environment objects to the simulator. When the "Simulate" button is clicked, the user can simulate the motion trajectory control by running a bash script file which stores the predefined joint trajectories.

Note that the Rviz and Gazebo displays can be operated separately while at the same time they are integrated to make use of advantages of both softwares.



Global Options	
Fixed Frame	world
Background Color	255;
Frame Rate	30
Global Status: Ok	
Fixed Frame	OK
Grid	✓
RobotModel	✓

Background Color
Background color for the 3D view.

Figure 3: Displays of Rviz and Gazebo with Rviz Plugin interface.

SIMULATION OF THE SHARED CONTROLLED ROBOT TASK

The RADOE simulator has tried to incorporate lots of industrial robot models into the simulation platform [13]. In this poster, another application of the RADOE simulator is addressed. The KUKA LBRiiwa industrial robot with 7 axis is used to write some characters on the surface of a sphere under the control of a haptic device through tele-operation [4].

To realize this simulation task of robotic drawing system, we need to use the haptic device as the input of the simulator instead of the real robotic drawing system. Another issue we should solve is that the robot drawing system works in Windows system, while the RADOE simulator works in Linux Ubuntu system. Moreover, the control input of the real robot drawing system is the position of the end-effector, while the ROS Rviz cannot provide inverse kinematics solver for the joint trajectory control. So we will replace the Rviz with MoveIt [14], which provides motion planning resources for the operation task.

The overall structure of the kuka tele-operated drawing system is shown in Figure 4. The real robot drawing system with its Windows software (server) is linked to the RADOE simulator with Linux software (client) through the TCP/IP protocol, as shown in Figure 5. In the "radoe_demo" panel of Figure 3, the "Start" button is used to start the Client terminal that is linked to the RADOE simulator, and the "Server" button is used to start the server terminal that is linked to the real robot operation system. The server and client terminals communicate with each other automatically through the TCP/TP protocol [15]. When the server sends request of simulation to client, the operator on the client site will control the motion of robot end-effector by the Omega

7 haptic device as shown in Figure 4. When the pen tip on the end-effector touches the surface of the sphere, markers will be published to symbolize the writing of the pen, as shown in Figure 6. If the end-effector can complete the writing task successfully, the operator will click "Go" button to send command to the server site so that the real robot is allowed to execute the drawing task. Otherwise, if the end-effector fails to finish the writing task for certain reasons such as collision with obstacles or the unreachable spot of the task, the operator will click the "No Go" button to tell the server not to do operation task control of the real robot.

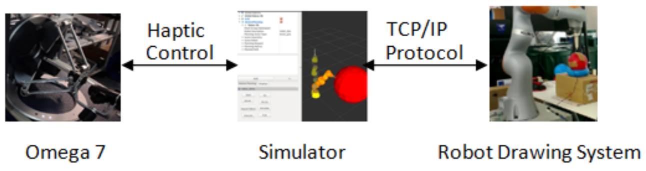


Figure 4: Overall structure of Kuka tele-operated drawing system with simulation.

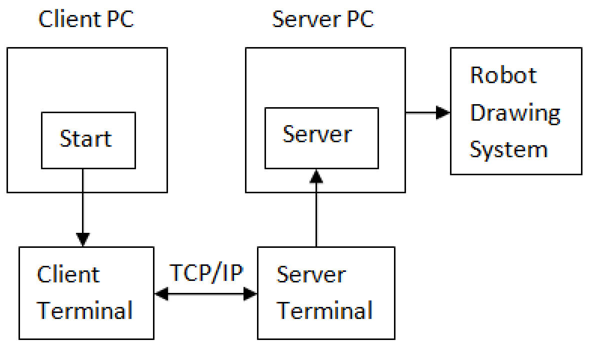


Figure 5: TCP/IP Protocol for the Robot Drawing System.

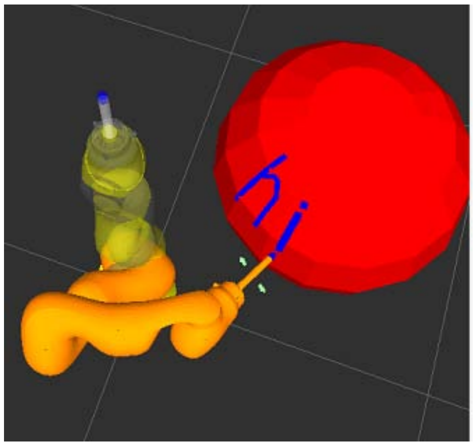


Figure 6: Simulation of Kuka drawing task on the surface of a sphere.

CONCLUSION

This poster put forward an approach to integrate the RADOE simulator with a tele-operated, human-robot shared controlled robot drawing system. A TCP/IP protocol has been established for communication between the RADOE simulator (as a cl ient) and the real robot drawing system (as a s erver). The simulator can accept command from the server to implement the simulation task under the control of a haptic device and send back signals to the server to tell if the execution of robot task is feasible or not.

Future work will continue to implement and complete the overall simulation system and improve the quality of user-friendly interface through formal user study.

ACKNOWLEDGMENTS

This work is supported by the Agency for Science, Technology and Research (A*STAR) Industrial Robotics Programme Grant 12251 0 0001 and Grant 12251 00008. The authors would like to give thanks to Dr Wu Yan, Mr Chua Yuanwei, Mr Aditya Narayanamoorthy and Mr Ng Kam Pheng for their supportive work.

REFERENCES

1. Robert Riener, Alexander Duschau-Wicke, Alexander Konig, Marc Bolliger, Martin Wieser, and Heike Vallery. 2009. Automation in rehabilitation: How to include the human into the loop. In *Proceedings of the World Congress on Medical Physics and Biomedical Engineering*, 180–183.
2. Yanan Li, Keng Peng Tee, Shuzhi S. Ge, and Haizhou Li. 2013. Building companionship through human-robot collaboration. In *International Conference on Social Robotics*, 1-7.
3. Yanan Li, Keng Peng Tee, Wei Liang Chan, Rui Yan, Yuanwei Chua, and Dilip K. Limbu. 2015. Continuous Role Adaptation for Human–Robot Shared Control. *IEEE Transactions on Robotics* 31, 3:672-681.
4. Yan Wu, Wei Liang Chan, Yanan Li, Keng Peng Tee, Rui Yan and Dilip K. Limbu. 2015. Improving Human-Robot Interactivity for Tele-operated Industrial and Service Robot Applications. In *Proceedings of the IEEE 7th International Conference on Cybernetics and Intelligent Systems and IEEE Conference on Robotics, Automation and Mechatronics* (CIS/RAM' 15), 153 – 158.
5. Bing L. Luk, David S. Cooke, Stuart Galt, Arthur A. Collie, and Sheng Chen. 2005. Intelligent legged climbing service robot for remote maintenance applications in hazardous environments. *Robotics and Autonomous Systems* 53, 2: 142-152.
6. Louis Whitcomb, Dana Yoerger, Hanumant Singh, and Jonathan Howland. 1999. Advances in Underwater Robot Vehicles for Deep Ocean Exploration: Navigation, Control, and Survey Operations. In *Proceedings of the Ninth International Symposium on Robotics Research*, 346-353.
7. Peter I. Corke. 1995. A computer tool for simulation and analysis: the robotics toolbox for MATLAB. In *Proceedings of the 1995 National Conference of the Australian Robot Association*, 319–330.

8. Logic Design Inc. Programming with RoboLogix. Retrieved from: http://www.robologix.com/programming_robologix.php
9. Tim Laue, Kai Spiess, and Thomas Rofer. 2005. SimRobot - A General Physical Robot Simulator and its Application in RoboCup. In *Proceedings of the Robot Soccer World Cup Symposium,* 173-183.
10. Benjamin Balaguer, Stephen Balakirsky, Stefano Carpin, Mike Lewis, and Christopher Scrapper. 2008. USARSim: a validated simulator for research in robotics and automation. In *IEEE/RSJ Workshop on "Robot Simulators: Available Software, Scientific Applications, and Future Trends".*
11. Steve Cousins, Brian Gerkey, Ken Conley, and Willow Garage. 2010. Sharing Software with ROS [ROS Topics]. *IEEE Robotics & Automation Magazine* 17, 2: 12-14.
12. Nathan Koenig and Andrew Howard. 2004. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Proceedings of the IEEE International Conference of Intelligent Robots and Systems* (IROS' 04), 2149-2154.
13. Longjiang Zhou, Renjun Li, Kam Pheng Ng, Aditya Narayanamoorthy, and Zhiyong Huang. 2016. A robotics simulator platform for RADOE. In *Proceedings of the IEEE International Conference on Control, Automation and Robotics (ICCAR'16)*, 44–48.
14. Ioan A. Sucan and Sachin Chitta. 2013. Moveit!. Retrieved in 2013 from http://moveit.ros.org.
15. Silver Moon. 2012. Server and client examples with C sockets on Linux. Retrieved on July 30, 2012 from http://www.binarytides.com/server-client-example-c-sockets-linux/.