

# A Robotics Simulator Platform for RADOE

Longjiang Zhou, Renjun Li, Kam Pheng Ng, Aditya Narayanamoorthy, Zhiyong Huang

Robotics Department,  
Institute for Infocomm Research  
Singapore 138632  
e-mail: zhoul@i2r.a-star.edu.sg

**Abstract**—This paper presents a new simulation platform for the industrial project of Robot Application Development and Operating Environment (RADOE). The simulator displays the robot models in Gazebo and Rviz interfaces concurrently starting from the bare bone framework. A Rviz interface plugin is developed for users to easily load the launch files or parameter files to the simulation platform. Experiments are designed to show some typical industrial robots integrated with their end effectors.

*Keywords*-RADOE; robotics; simulator; plugin

## I. INTRODUCTION

A robotics simulator is software to produce embedded systems for a robot without physically relying on the real robot, thus reducing development cost and improving efficiency by simplifying the development work. By way of modern techniques, the behaviours of a robot can be simulated with high accuracy in comparison to the actual robot. Most robotics simulators can be applied to the actual robots without need of major modifications. Moreover, robotic systems are becoming more complex nowadays, and requirements of the performance are higher, so robotics simulation is vitally important in the research and development of the robotic systems. In particular modern applications emphasize that the robots can operate autonomously and interact closely with humans in medical robots [1], assistive robots [2], rescue robots [3] and so on. Furthermore, simulators applied in robotics research are highly efficient in examining the safety, reliability and robustness of algorithms under development [4]. In the education area, researchers have also shown that students can learn programming languages faster by way of robotics simulation [5]. There are a lot of robotics simulations systems nowadays, which can be mainly divided into two categories: commercial and open-source.

Webots [6] is an example of a commercial simulator for fast prototyping of mobile robots. The simulated models have customized properties such as mass and friction provided by the Open Dynamics Engine (ODE) Library [7] which is used for the dynamics simulation of the rigid body. Another kind of commercial software, V-REP (Virtual Robot Experimentation Platform) [8], is a 3D simulator which is used to simulate any robot, especially in distributed control system, and is versatile application compatible with Linux, MacOS, and Windows. V-REP is also used for fast development of algorithm, fast prototyping, remote monitoring, and so forth. Easy-Rob [9] is simulation

software for robotic platforms in manufacturing plants that permits the user to visualize the operation processes like handling and coating. Easy-Rob also has functionality of generating AVI movies and providing libraries and the import / export of CAD files. RobotStudio [10] is a simulator that uses the RAPID language in Windows as software platform for the industrial robots developed by ABB. This software platform allows users to develop realistic simulation environments based on virtual robots which are exact copies of the software run in production. MRDS (Microsoft Robotics Developer Studio) [11] is a Windows-based 3D software developed by Microsoft to support a variety of sensors and many other platforms. This simulator is compatible with all versions of Windows OS and it is also able to simulate service robots.

The commercial robotic simulation softwares usually have high quality and advanced technologies, but at the same time they greatly increase the development cost for users. In addition, many commercial suppliers only provide simulators for their own products, which may lack versatility and flexibility in robot development. On the other hand, open-source softwares can avoid the repetition research work so as to improve the efficiency of research and shorten the product development cycle as possibly by enabling the research community to exchange their ideas, and they are also becoming more powerful and reliable in robotics simulation applications, so open-source robotics simulators are attracting more and more interest nowadays.

MORSE (Modular Open Robots Simulation Engine) [12] is a generic open-source simulator on the basis of Blender [13] and it is mainly applied in academic fields to simulate autonomous robots in indoor or outdoor environments. The programs are mostly written as simple Python scripts. ARS [14] [15] is another example of generic open-source simulator for academic robotics written in Python. iCUB [16] [17] is a cognitive simulation platform for a complete humanoid robot developed with kinematics complexity. ROS (Robot Operating System) [18] is an open-source distributed software framework for robotics that allows executables to be flabbily coupled at run time, which enables robotics researchers to reuse the source codes from ROS community. ROS is based on Linux Ubuntu operating system and supports a variety of languages across C++, Python, Octave, and LISP by way of a simple Interface Definition Language (IDL). ROS reuses source codes from

numerous other open source projects such as Player [19] and OpenRAVE [20]. ROS is a modular and tool-based platform that makes it easy for others to build their own robotic software systems. ROS also uses MoveIt [21] packages to provide mobile manipulation and path planning functionalities for robots. To solve the possible singularity issues, an improved version called “Cartesian Path Planner Plug-in” was created for mobile robots in MoveIt. ROS also utilizes Rviz as its visualization plugin, which has a Graphical User Interface (GUI) for users to configure robots. The simulator integrated in ROS was Gazebo [22] [23], a general purpose, high fidelity and open-source modelling and physical simulation tool for rigid robots which is able to do kinematics as well as dynamics simulation for complex systems and sensor components. From Gazebo 1.9 onwards, however, it became a stand-alone package running in Ubuntu and separating from ROS. Fortunately, a package called gazebo\_ros\_pkgs [24] was developed to combine Gazebo with ROS, which made use of the powerful motion planning and enriched resources of ROS and strengthened rigid body simulation functionality of Gazebo.

For industrial applications, the software should ideally be user - friendly and as easy as possible for programmers to make programs and for operators to manoeuvre the practical machines. At the same time, cost efficiency is vitally important for the competitiveness of products. So we have developed an open-source software called RADOE (Robot Application Development and Operating Environment). RADOE works on the basis of ROS Industrial packages and it is expected to improve manoeuvrability and product efficiency in the manufacturing industry. This paper puts forward a new simulation platform for RADOE. Section II describes the overall system architecture of the simulator. Section III elaborates the methodologies of this simulator. Section IV shows the experimental procedure and results. Section V draws the conclusions.

## II. SYSTEM ARCHITECTURE

The simulation platform uses ROS Rviz and Gazebo for visualization as show in Fig. 1. The trajectory generator node reads the input trajectories from external trajectory files, parses data and ensures that the supplied data is valid. If data is not valid, simulation stops and error messages is displayed. If data is valid, then it is sent to the joint trajectory controller to control the motion of the robots. The robot states are then published and sent to the visualization tool Rviz for display. At any point of time, simulation can be paused, resumed or stopped using the trajectory generator node.

To support different robots, the trajectory files have a flexible file structure where the number of robot’s joints are not fixed and are left to the user to define. This allows easy creation and management of the trajectory files as well as controlling just a subset of the joints instead of the whole robot. For example, a trajectory file can contain only the gripper - related trajectories to perform a gripping task.

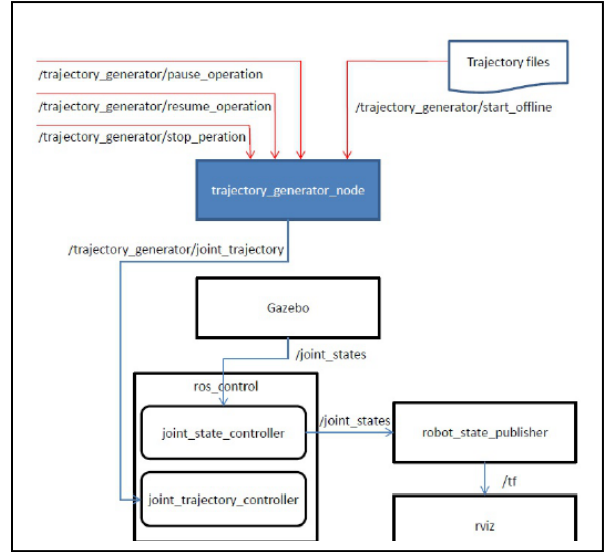


Figure 1. The Flow Chart of Rviz and Gazebo.

## III. METHODOLOGIES

This project aims to develop a user-friendly simulation platform for industrial robots. A bare-bone system file is launched to display both Gazebo and Rviz concurrently on the same screen without any robots or their components. Then a robot model is imported, as seen in Fig. 2. The right-hand side is the display of Gazebo and left-hand side is Rviz.

A Graphical User Interface (GUI) has been developed in the Rviz framework. The GUI is developed by following the Rviz plugin method. It could be easily plugged into the Rviz window, as shown on the bottom left corner in Fig. 2.

There are 4 buttons in this plugin: “Import robot”, “Import environment”, “Simulate” and “Execute”. When the “Import Robot” button is clicked, an “Open File” window will pop-up so that users can choose and import target models into the simulator both in Gazebo and in Rviz.

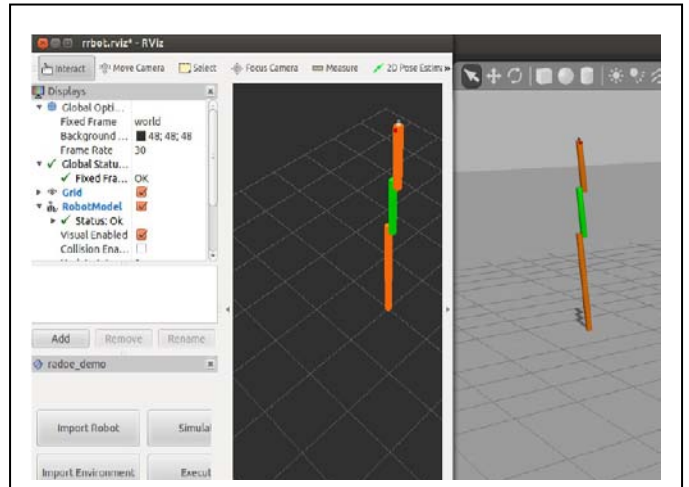


Figure 2. The Displays of Rviz and Gazebo with a robot model.

The example robot of “rrbot” (“Revolute - Revolute Manipulator Robot”, a simple structure of 3-link and 2-joint robot) is used in this paper to demonstrate the operation procedure of the RADOE simulation platform. The “rrbot” is chosen through the “Import Robot” button and displayed in both Gazebo and Rviz concurrently. Note that URDF files of robots and their components should be adapted to the requirements of Gazebo for version 1.9 and above. The URDF transformation guideline can be found in the Gazebo website [25]. Now the display of the imported robot model can be operated respectively in Gazebo and Rviz.

The configuration file of the motion controller and the Gazebo launch file are also required of modification. For the “rrbot\_control.yaml” configuration file in the package of “rrbot\_control”, the original position control is substituted by the trajectory control in the newly developed simulation platform. Therefore, the configuration file may be modified by replacing the joint position controller with the joint trajectory controller. For the “rrbot\_world.launch” file in the “rrbot\_gazebo” package, the basic settings of gazebo, launch of “empty\_world.launch”, and the trajectory generation node have already been fulfilled in the “bare\_bone” package previously. Thus, the modified “rrbot\_world.launch” file only needs to import the rrbot URDF file, generate the robot model in the Gazebo display, launch the trajectory control file, and update the robot state in the Rviz display.

The ambient objects can also be imported into the simulation platform via the button “Import Environment”. For example, we can import some table into the Gazebo and Rviz displays and put some specific objects on the table for the robot to pick up. The ambient objects are chosen from a model library.

The “Simulate” button enables operators to simulate the robot’s motion control by following some predefined joint trajectories, which are stored in some bash script files (with the extension name “.sh”). The bash script files can be written and edited offline according to the requirements of the robot’s trajectory. We have written a bash script file “rrbot\_senddata.sh” as an example. In this file, the angular trajectory of each movable joint is published sequentially through a “rostopic” command.

When the “Simulate” button is clicked and the “rrbot\_senddata.sh” file is chosen through the “Open File” pop-up window, the commands in this bash script file will be executed step by step. The robot state in Gazebo and Rviz displays will be updated step by step accordingly. Fig. 3 displays the ultimate states of rrbot links after the link 1 rotates for 45 degrees and the link 2 rotates for 90 degrees.

The “Execute” button is used to activate the control of robot’s behaviours in a real world environment. This button is not in use yet in the simulation stage, but it will be used to guide the real world robotics motion control in future.

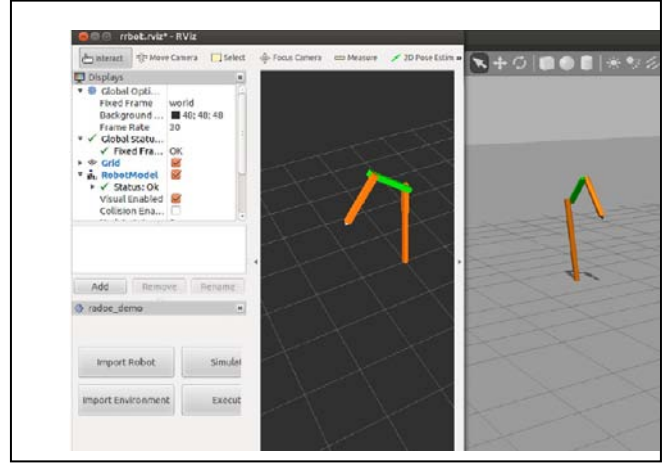


Figure 3. The states of rrbot links after running the joint trajectory file.

#### IV. EXPERIMENTAL RESULTS

A series of experiments has been designed to incorporate some typical industrial robots and their end effectors into the hereby developed RADOE simulation platform.

The first model incorporated is a modified version of ABB IRB6600 industrial robot [26]. The result of importing the robot into the simulation platform by clicking the button of “Import Robot” on the Rviz plugin is shown in Fig. 4. On clicking the “Simulate” button and choosing the ABB joint trajectory control file (“abb\_senddata.sh”, a bash script file that is written and edited offline), all the movable links can move according to predefined trajectories generated by the joint trajectory control file, and the displays of Gazebo and Rviz can also be updated concurrently.

It is shown that the RADOE simulation platform integrates the display of Gazebo with Rviz, so it can take advantage of the two well – known frameworks that are familiar to the task programmers and operators. For example, Rviz is a nice 3D visualization tool which can incorporate interactive markers and point cloud images for laser scan data, while Gazebo is famous for its functionality of dynamics simulation for physics of rigid bodies.

At the same time, the simulation platform that we are developing is an open architecture that allows a plug-in to be integrated in the similar style of GUI. Thus, it is extendable and the same style of GUI allows users to start using new functions easily and rapidly.

Another model that we have incorporated into our RADOE simulation platform is KUKA LBRiiwa industrial robot, of which the ROS packages of robot description (in the format of URDF), Gazebo and control are found in the Github website [27]. The packages will be modified following rules similar to those of the example robot “rrbot” of this paper. The KUKA robot is displayed concurrently in Gazebo and Rviz.

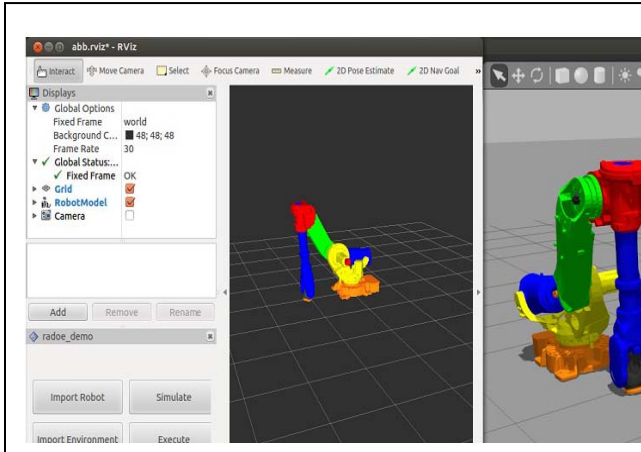


Figure 4. Simulation of the ABB robot in displays of Rviz and Gazebo.

We also wrote a bash script file “kuka\_senddata.sh” as a joint trajectory control file, where the angular trajectory of each movable joint is published sequentially. Fig. 5 shows the simulation result of robot motion when the joint trajectory control file is executed by clicking the “Simulate” button and choosing the “kuka\_senddata.sh” file.

Another model incorporated into the RADOE simulation platform is a UR10 robot [28] combined with a Meka arm [29], as shown in Fig. 6. The joint trajectories of the arm and the robot can be controlled concurrently through one single “gazebo\_ros\_controller” by clicking the “Simulate” button and choosing the bash script file “ur10\_senddata.sh”.

One more simulation of an automatic robot taping system [30] operated by a UR10 robot has also been incorporated into the simulator and is shown in Fig. 7. The Meka arm shown in Fig. 6 is replaced by another gripper as the end-effector. The taping system is shown in Gazebo and Rviz displays. Similarly, the joint trajectories of the gripper and the UR10 robot can be controlled concurrently through one single controller by clicking the “Simulate” button.

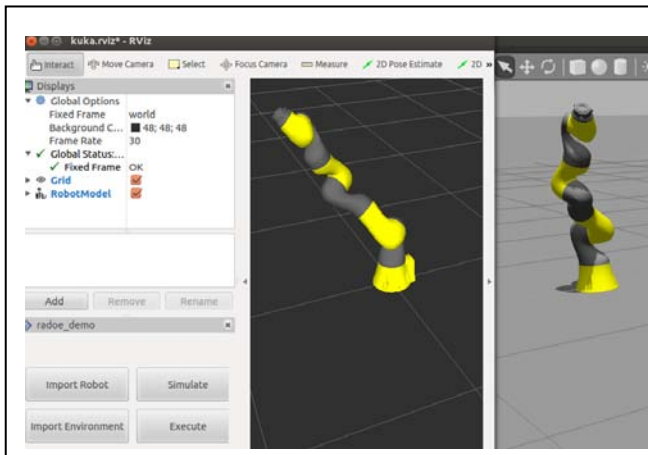


Figure 5. Simulation results of KUKA joint trajectory control

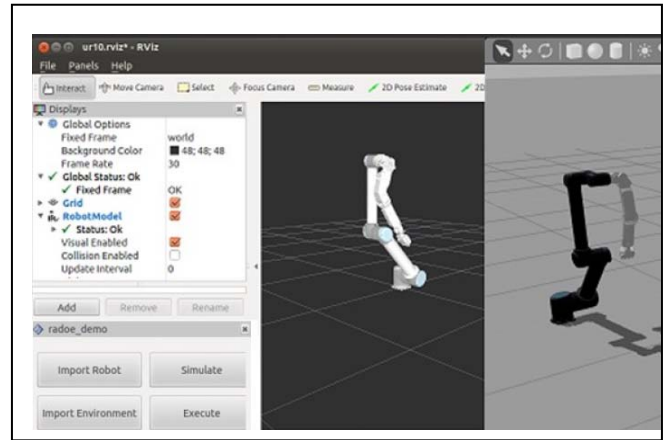


Figure 6. Simulation of a Meka arm integrated with the UR10 robot.

## V. CONCLUSIONS

This paper put forward a new user friendly simulation platform for the RADOE software framework for industrial robotics to improve product efficiency and bring convenience and manoeuvrability to task programmers and operators. A Rviz plugin interface was developed to ease the Task Programmers’ workload of loading robot models, environment objects, controllers and other files into the simulation system. A joint trajectory controller was applied to control the joint trajectories of different robots and end-effectors. The simulation platform can show the individual robot and update its state in both the Gazebo and the Rviz displays concurrently after execution of the joint trajectory motion control file. The open structure of this simulation platform allows the Rviz plugin interface to be extendable and so enables users to develop new functions rapidly and easily.

Experimental results by incorporating some typical industrial robots and their end effectors showed that the simulation platform is general can make use of advantages of both Rviz and Gazebo in display and joint trajectory motion control. This newly-developed simulation platform is user friendly in the preliminary user study for simulating the taping task. With integrating the Gazebo and Rviz seamlessly, the task programmers and operators are able to make use of the rich resources of the two well - known frameworks efficiently.

Future work will be continued with a more systematic study for more simulation tasks. We will also incorporate other typical models of industrial robots and their end effectors into this simulation platform. The interface performance may be improved by incorporating more functional buttons in the Rviz plugin. We also intend to integrate other industrial robotics projects into this simulation platform to make a generic purpose and standard simulator in applications to the real world robotics environment of manufacturing sectors.

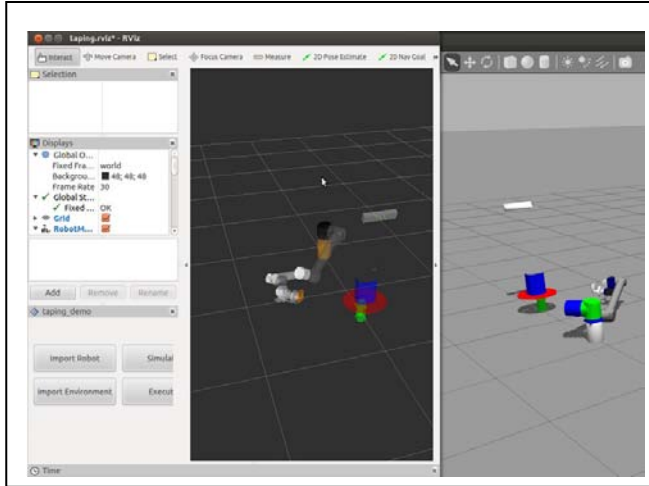


Figure 7. Simulation of a taping system operated by UR10.

#### ACKNOWLEDGMENT

The work of this paper is supported by A\*STAR (Agency for Science, Technology and Research) SERC (Science and Engineering Research Council) Industry Robotics Programme Grant 12251 00008.

#### REFERENCES

- [1] R. H. Taylor and D. Stoianovici, "Medicalrobotics in computer-integrated surgery," *IEEE Trans. Robotics and Automation*, vol. 19, pp. 765 – 781, Oct. 2003.
- [2] D. Kim, R. Lovelett, and A. Behal, "An imperical study with simulated adl tasks using a vision-guided assistive robot arm," *Proc. IEEE Int. Conf. Rehabil. Robot.*, June 2009, pp. 504-509.
- [3] R. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Trans. Systems, Man and Cybernetics, Part C*, vol. 34, pp. 138-153, May 2004.
- [4] K. Kimoto and S. Yuta, "Autonomous mobile robot simulator – a programming tool for sensor-based behaviour," *Autonomous Robots*, vol. 1, Jun. 1995, pp. 131-148.
- [5] A. Liu, J. Newsom, C. Schunn, and R. Shoop, "Students learn programming faster through robotic simulation," *Tech Directions*, vol. 72, Mar. 2013, pp. 16-19.
- [6] O. Michel, "Cyberbotics ltd webots professional mobile robot simulation," *Int. J. Advanced Robot. Sys.*, vol. 1, 2004, pp. 39-42
- [7] Open Dynamics Engine. [Online]. Available: <http://www.ode.org/>.
- [8] E. Rohmer, S. P. N. Singh, and M. Freeze, "V-REP: a versatile and scalable robot simualtion framework," *Proc. Int. Conf. Intelligent Robots and Systems (IROS 13)*, Mar. 2013, pp. 1321-1326.
- [9] Easy Rob 3D robot simulation tool. [Online]. Available: [http://www.easy-rob.com/easy\\_rob.html](http://www.easy-rob.com/easy_rob.html).
- [10] L. Xu, "Robotstudio: A modern IDE-based approach to reality computing," *Proc. The 38<sup>th</sup> SIGCSE Technical Symposium on Computer Science Education (SIGCSE 07)*, Covington, Kentucky, Mar. 2007, pp. 440-444.
- [11] J. Jackson, "Microsoft robotics studio: Atechnical introduction," *IEEE Robotics and Automation Magazine*, Dec. 2007, [Online]. Available: <http://msdn.microsoft.com/en-us/robotics>.
- [12] G. Echeverria, N. Lassabe, A. Degroote, and S. Lemaignan, "Modular open robots simulation engine: MORSE," *Proc. Int. Conf. Robot. and Automation*, 2011, pp. 46-51.
- [13] Blender 3D. [Online]. Available: <http://www.blender.org>
- [14] G. Larrain-Munoz, A robotics simulator for Python, Master Thesis, Pontificia Universidad Catolica De Chile, Mar. 2014.
- [15] ARS: Python robotics simulator. [Online]. Available: <https://ars-project.readthedocs.org/en/latest>.
- [16] N. G. Tsagarakis, B. Vanderborght, M. Laffranchi, and D. G. Caldwell, "The Mechanical Design of the New Lower Body for the Child Humanoid Robot 'iCub'," *Proc. Int. Conf. Intelligent Robots and Systems (IROS 09)*, Oct. 2009, pp. 4962-4968.
- [17] V. Tikhandoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale, and F. Nori, "An open-source simulator for cognitive robotics research: the prototype of the iCub humanoid robot simulator," *Proc. 8<sup>th</sup> Workshop on Performance Metrics for Intelligent Systems (PerMIS' 08)*, 2008, PP. 57-61.
- [18] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Weeler, and A. Y. Ng, "ROS: An open-source robot operating system," *Proc. ICRA Workshop Open Source Software*, 2009, pp. 1-6.
- [19] R. T. Vaughan and B. P. Gerkey, "Really reusable robot code and the Player/Stage Project," *Software Engineering for Experimental Robotics*, Springer Tracts on Advanced Robotics, D. Brugali, Ed. Springer, pp. 267-289, 2007.
- [20] S. Srinivasa, D. Ferguson, J. M. Vandeweghe, R. Diankov, D. Berenson, C. Helfrich, and H. Strasdat, "The robotic busboy: steps towards developing a mobile robotics home assistant," *Proc. Int. Conf. on Intelligent Autonomous Systems*, vol. 10, July 2008, pp. 129-138.
- [21] S. Chitta, I. Sukan, and S. Cousins, "MoveIt! [ROS Topics]," *IEEE Robotics and Automation Magazine*, vol. 19, no. 1, pp. 18-19, 2012.
- [22] N. Koenig, and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," *Proc. IEEE Int. Conf. Intelligent Robots and Systems (IROS 04)*, Sep. – Oct. 2004, pp. 2149-2154.
- [23] Gazebo: Robot simulation made easy. [Online]. Available: <http://gazebosim.org/>.
- [24] Gazebo\_ros\_pkgs. [Online]. Available: [http://wiki.ros.org/gazebo\\_ros\\_pkgs](http://wiki.ros.org/gazebo_ros_pkgs).
- [25] Tutorial: Using a URDF in Gazebo. [Online]. Available: [http://gazebosim.org/tutorials/?tut=ros\\_urdf](http://gazebosim.org/tutorials/?tut=ros_urdf).
- [26] Abb\_irb6600\_support. [Online]. Available: [http://wiki.ros.org/abb\\_irb6600\\_support?distro=hydro](http://wiki.ros.org/abb_irb6600_support?distro=hydro).
- [27] [Online]. Available: [https://github.com/rtkg/lbr\\_iiwa](https://github.com/rtkg/lbr_iiwa).
- [28] Ur\_description. [Online]. Available: [http://wiki.ros.org/ur\\_description](http://wiki.ros.org/ur_description).
- [29] Meka-ros-package. [Online]. Available: [https://github.com/ahorau/meka-ros-pkg/blob/master/meka\\_description/robots/m3.urdf.xacro](https://github.com/ahorau/meka-ros-pkg/blob/master/meka_description/robots/m3.urdf.xacro).
- [30] T. S. Lembono, Q. Yuan, Y. Zou, and I. M. Chen, "Automatic robot taping: system integration," *Proc. Int. Conf. Advanced Intelligent Mechatronics (AIM 15)*, July 2015, pp. 784-789.