Manipulation and Path Planning for KUKA (LWR/LBR 4+) Robot in a Simulated and Real Environment

Submitted: 30th November 2016; accepted: 8th August 2017

Tarwadi Parag Jitendra, Arockia Selvakumar Arockia Doss, Juan Antonio Corrales Ramon

DOI: 10.14313/JAMRIS_3-2017/24

Abstract:
Robotics has accomplished its greatest triumph to date in the world of industrial manufacturing and academia. This work aims to perform path planning using a KUKA (LWR/ LBR 4+) robot platform as well as a simulator to grasp the object. This whole implementation will be carried out in a ROS environment with Ubuntu (Linux) as an operating platform. The KUKA (LWR/ LBR 4+) has 7 degrees of freedom with several joints and linkages. It uses KR C2 LR as the main hardware controller. The robot gets visual information of an object by Microsoft Kinnect RGB-D camera and carries out necessary actions to clasp the object using a shadow hand and Barrett hand. The simulation and manipulation of robot gantry is performed by using C++ and python as a programming language. The bilateral robot platform and main PC hub are linked together by using Ethernet cable. The obtained results from the current research are found to be satisfactory and can be proven beneficial for researcher as a reference.

Keywords: *ROS (Indigo), motion planning, moveIt simulation, OMPL, robotics, KUKA LWR robotic arm*

1. Introduction

Robots have the potential to improve efficiency of proposed work, safety and convenience of human endeavors. To realize this hidden potential, research in robotics necessarily involves investigation into a broad range of various disciplines [3]. A fundamentally vital discipline is that of motion planning and manipulation. It is important because motion is a primary requirement, which is common to all robots [6], [12].

While the fact remains that all robots necessarily move, the actual motion execution is highly dependent on the robot under consideration, the task it fulfils and the constraints it operates under. Therefore, numerous algorithms, concepts and theories have been developed for solving specific classes of motion problems [2].

The bilateral research collaboration between Institute of Robotics and Mechatronics and KUKA Roboter GmbH at the German Aerospace Center (DLR) leads to the development of KUKA LWR [7]. Due to its remarkable features like programmable active compliance, high payload ratio and torque sensor feedback system, it enables automation engineers and researchers to develop new industrial and service robotic applications. Figure 1 represents the KUKA LWR/ LBR 4+ [25].

Fig. 1. Snapshot of KUKA LWR/ LBR 4+ [25]

Nowadays, due to rapid increase in production rate, automation plays a vital role in industrial environment [22]. Generally, the robots are often required to pick some object and place it at an exact or different place in working environment [16] where the robots are programmed to follow a path with knowledge of motion planning and manipulation [24]. This entire task is crucial and requires speedy operation with accuracy.

Recently, a very luxuriant literature related to motion planning and manipulation of robots is available. Chettibi *et al*. have developed a minimum cost trajectory planning path for robot manipulators. By considering dynamic equations of motion as well as bounds on joint positions, jerks, velocities and torques, it contains linking of two points in the operational space while minimizing a cost function [1]. Lars Blackmore *et al.* have presented an algorithm for robot path planning with obstacles. By using a Disjunctive Program, they are able to use existing constrained optimization methods to generate minimum trajectories. The proposed method plans entirely in the manipulator working area by minimizing the cost of the process of mapping the obstacles from the workspace into the configuration space [2]. John *et al.* have introduced a general motion planning approach based on evolutionary computation that is suitable for both real-time and offline adaptive motion planning for robots under multiple optimization criteria and manipulator constraints in environments with obstacles having changes not known as a priority. The outcome of testing results demonstrates the effectiveness and efficiency of the approach [3]. Dmitry has described the design of efficient algorithms for special instances of robot motion planning problems and prediction of motion of fluids. The intricate nature of these problems may manifest itself in enhanced computational complexity. The main contribution of this work is a brilliant approach for motion planning problem for multiple tasks distributed in an open space. This algorithm computes the minimum-time frame of robotic motion considering orientation constraints of the robot end-effector during task execution and for a given velocity [4]. Solteiro Pires *et al.* have presented a multi-aim genetic algorithm based technique to address the problem of a non-trivial optimization in generating manipulator trajectories while considering obstacle avoidance and multiple objectives. Multiple criteria are optimized by considering up to five different objectives. The outcome of simulation is incorporated for robots with two and three degrees of freedom, considering two and five optimizations. A subsequently correct analysis of the spread and solution distribution along with the converged non-dominated Pareto front is carried out, in terms of achieved diversity [5]. Günter Schreiber *et al.* have described the capabilities of interface, the practical realization within the KUKA LWR robotic manipulator control architecture and first application of the real-time interface. In this, the development of the Fast Research Interface for the KUKA Lightweight Robot was carried out by the researchers to provide such an interface [7]. Yanyu Su *et al.* have introduced a most significant approach to develop a kinematic controller for any serial type of robots using MoveIt! Simulator and PMAC, which is a commercially available product for motion control and also an embedded system to manipulate the motion, amplify signals and acquire sensory information. MoveIt! is a simulation software for kinematics based manipulators [8]. Asad Yousuf *et al*. have examined ROS based introduction of kinematics for robotic manipulator. ROS implements tool for kinematic transforms, as a unique part for the ROS Core Libraries [9]. Stephen Hart *et al.* have described a ROS package for quickly programming, adjusting and executing robot applications in RViz environment. This developed package extends the capabilities of RViz interactive markers by allowing an operator to specify grasp poses in object-centric coordinate frames and multiple end-effector waypoint locations and to adjust these waypoints in order to meet the run-time demand of the task [10]. Mathew et al have explained their software architecture and hardware choices, which ensure human loop control of a 28 degree-of-freedom ATLAS humanoid robot over a limited bandwidth link [11]. Philipp *et al.* have described a Robot Operating System based middleware software framework for the NimbRo-OP. This software provides functionality for visual perception, hardware abstraction and behavior generation [12]. Ioan *et al.* have developed many state-of-the-art motion planning algorithms. These all algorithms included in an Open Motion Planning Library for sampling based motion planning. The library is designed with minimum input for solving a complex motion planning problems [12], [14]. Coleman *et al*. have presented a case study related to the lowering of the barrier of an entry in the MoveIt! framework, which is an open platform tool for a mobile manipulation in ROS. ROS allows user to quickly get basic motion planning functionality with minimal initial setup, automate its configuration and customized components. In short, the developed best practices are summarized into a set of the barrier to entry design principles applicable to the other major robot software packages [15]. Dirk *et al.* have represented a generalized system for mobile manipulators based automated kitting with focus on de-palletizing task. In order to allow low cycle time, they prepare particularly efficient solution for motion planning and execution as well as object perception [16]. Jorg Stuckler *et al.* have proposed a system for de-palletizing and localizing objects as well as analysis and verification for found object, which does not differ from the known object modes. Their approach is based on the multi-resolution surfel model [17]. Stephan *et al.* have developed a system, which includes RGB-D camera for surveillance of the common working place, an optical distance sensor to control shadowing effects of the RGB-D camera and a laser range finder to detect a co-worker. This ROS software is based on the behavior of the co-worker and predicts collision and path planning [18]. Marsette *et al.* have developed an open hardware mobile manipulator for teaching robotic software and algorithm that uniquely combines a 3DOF arm and gripper, low and high level processors, portability, a mast-mounted camera and the possibility to be incrementally assembled in different configurations [19]. Rhama Dwiputra *et al*. have presented the development of a distinct Modelica model for the youBot manipulator. Apart from other robotic simulators Modelica allows the modeling of the manipulator controller and motors [20]. Dereck *et al*. have implemented a navigation based planner on a Pioneer 3-AT mobile robot using the ROS software library. They updated the existing ROS software to enable functionality and compatibility with Michigan Tech's Pioneer robot [21].

The research in this project would lead to more efficient and generalized solution to automate the pick and place operation of KUKA robotic arm by using OMPL.

The working hypothesis throughout the paper is that, it is possible to have a generic software framework for robot motion planning and manipulation, which can be configured to solve specific robot motion tasks [5]. The contribution of this paper is the validation of the hypothesis via realization and real-world testing of such a framework. Additionally, the research work also resulted in the development of a software library which can be used by software programs to interact with the framework [8], [15].

The main objective of this research work is to manipulate the KUKA robot by feedback of visual information for assisting a shadow hand and Barrett hand to grasp the object. This research work can be further used in the fields of robotics, manufacturing line, material handling systems; food industry and automotive industry for pick and place operation [9].

2. Methodology

Figure 2 represents the complete methodology for whole research work. Manual robots teaching using tech pendant is performance wise disadvantageous over automatic robot teaching [3], [10]. To overcome this issue visual feedback with programming automated control is useful. For this, the simulation version of real robot was made using a URDF architectural file and RViz software package in ROS. Desired movement of robot is achieved by using C++ code, while testing is performed using interactive markers. At the final stage integration of developed software package and the real robot was made.

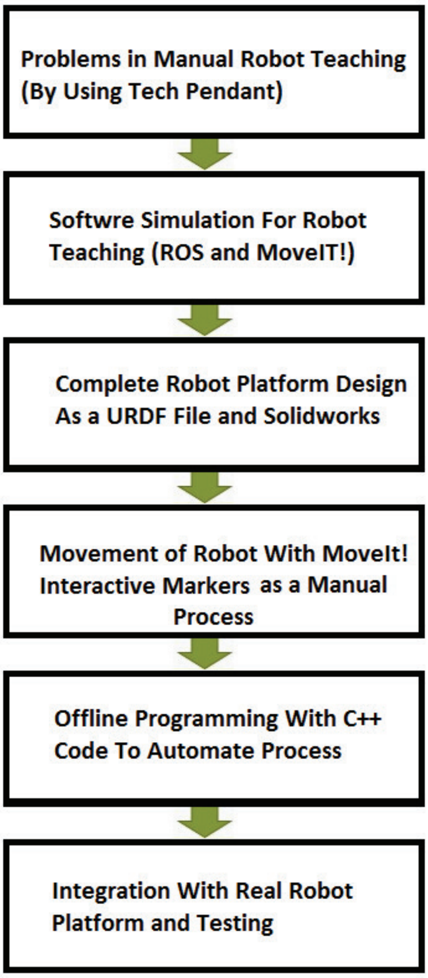

Fig. 2. Methodology of the proposed work

2.1. Hardware Overview

The KUKA (LWR/ LBR 4+) has 7 degrees of freedom with several links and joints. It uses KR C2 lr as a main controller [19] [25]. Figure 3 represents the complete research work. In which the robot gets visual information of object by Microsoft Kinnect RGB-D camera [18] and carries out necessary actions to clasp the object using a shadow hand or Barrett hand. The simulation and manipulation of robot gantry is performed by using C++ and python programming languages. The robot platform and main hub are linked together by using Ethernet [20].

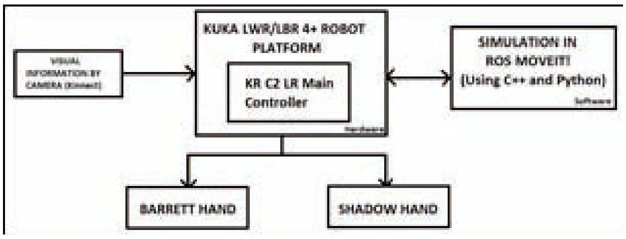

Fig. 3. Block Diagram of the whole task

Fig. 4. Experimental setup for the proposed work

Figure 4 shows the experimental setup of proposed work. In which a bilateral platform of KUKA LWR/ LBR 4+ robot with its controller and software arrangements are shown.

2.2. Software Simulation

A. Simulation of KUKA LWR in Rviz Environment with Interactive Marker

In MoveIt! the set up assists in decrease of motion planning process time, a default self-collision matrix generator algorithm was used to prevent collision of link pairs and helps to disable them safely [22]. These pairs of links are disabled when the links are adjacent

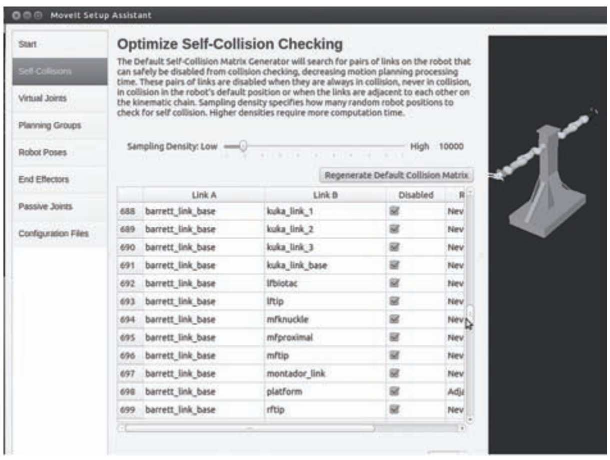

Fig. 5. Generation of Self Collision Matrix

to each other on the kinematic chain or when they are never in collision, always in collision, in collision in the KUKA LWR/ LBR 4+ robot's primary position. The calculation of how many random the KUKA LWR/ LBR 4+ robot positions to check for self collision specifies the sampling density. The lower densities have a higher possibility of disabling pairs that should not be disabled while higher densities require more computational time. The standard value is 10,000 collision checks. Collision checking is done in parallel to decrease processing time [23]. The generation of self collision matrix is shown in Figure 5.

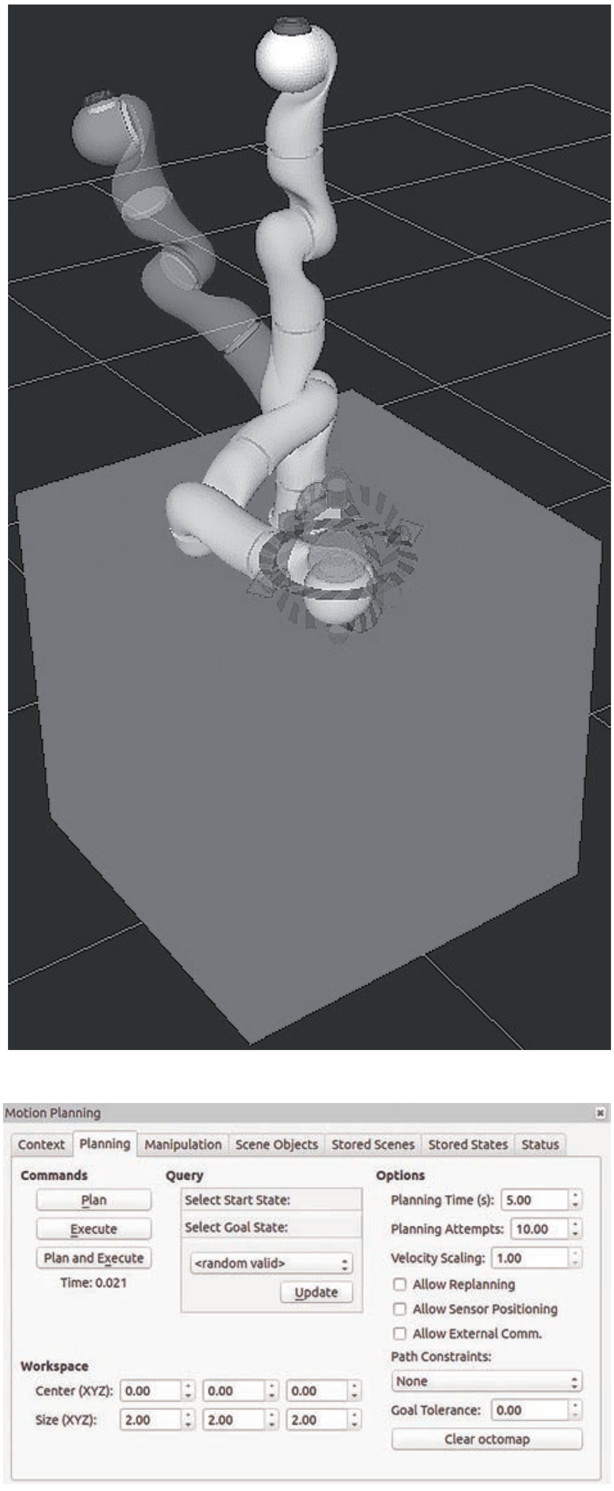

Fig. 6. Motion planning for KUKA LWR robot in MoveIt! using interactive marker

The motion planning for KUKA LWR robot in MoveIt! [14], [23] is done by using interactive marker. Start state was the home state of KUKA LWR robot, while final goal state was selected by interactive marker as shown in Figure 6. Then plan and execution command will be initialized.

B. Simulation of Bilateral KUKA LWR Platform in Rviz Environment with Interactive Markers

Figure 7 represents the motion planning for bilateral KUKA LWR robot platform in MoveIt! In which selection of final goal state is defined by using interactive marker, the command plan and execution was used for movement of arm. For minimum trajectory path selection for arm OMPL (Open motion Planning Library) was used [13].

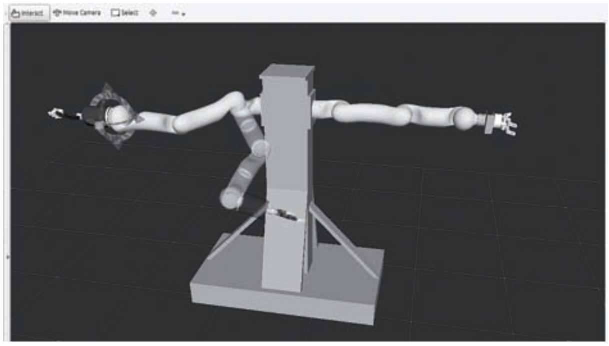

Fig. 7. Motion planning for Bilateral KUKA LWR Robot Platform in Moveit! using interactive marker

C. Simulation of Bilateral Kuka LWR Platform in Rviz and Real Environment Using C++ Code

Figure 8 shows the motion planning and manipulation of bilateral KUKA LWR robot platform in MoveIt! using C++ programming code. For this case there is no need of any RViz command was required i.e. Plan and

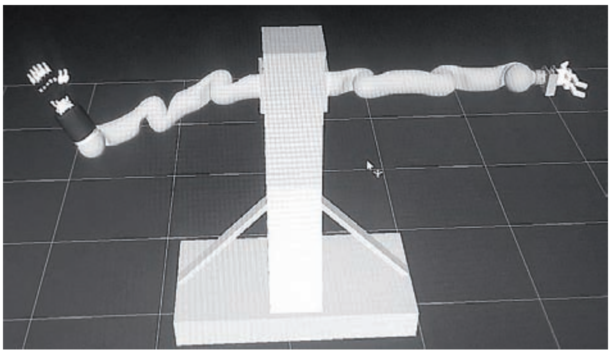

Fig. 8. Motion planning for bilateral Kuka LWR robot platform in MoveIt! using C++ Code

Execution etc. in which .cpp file with programming code was generated and it was executed by creating launch file in catkin workspace with required dependencies. For minimum trajectory planning for arm, OMPL was used. For this particular application, OMPL uses inverse kinematic solver to design required path.

Figure 9 shows the real bilateral KUKA LWR robot movement by following the path of simulated robot in RViz environment [21] using C++ code.

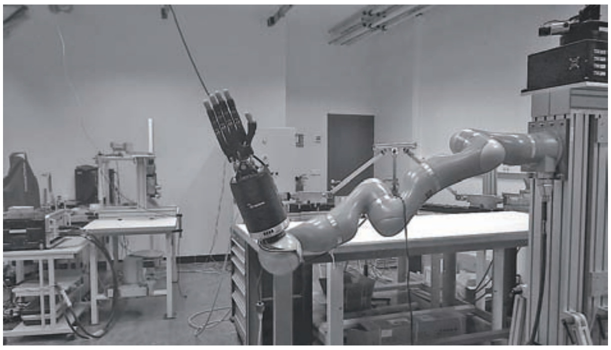

Fig. 9. Motion planning for real bilateral KUKA LWR robot platform in MoveIt! software using C++ Code

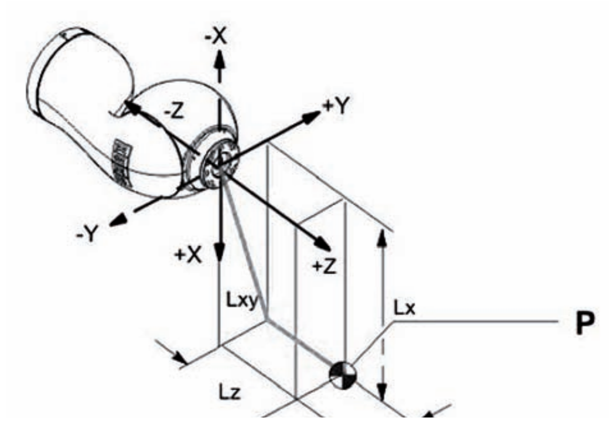

Fig. 10 Representation for axis of rotation [25]

Figure 10 shows the representation of axis of rotation for the KUKA robot platform.

Figure 11 shows the comparison for Simulated and Real Robot in terms of degrees of rotation of each joint.

3. Results and Discussion

Table 1 shows the particular translational and rotational axis of bilateral KUKA LWR robot with respect to particular time. It also gives RPY value i.e. rotation of joint along x- axis, y- axis and z- axis, respectively.

Table 2 shows the maximum torque for each axis of the robot.

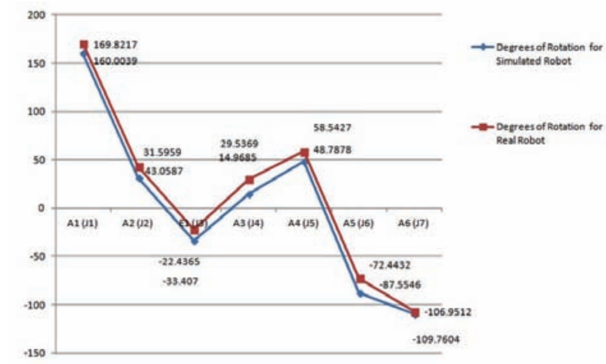

Fig. 11. Comparison for simulated and real robot

Table. 1. Representation of particular translational and rotational axis

Run Time (msec.)	Axis of rotation		
	Translational	Rotational	
		In Quaternion	In RPY
3431.476	[0.000, 0.001, 1.039]	[0.000, 0.000, 0.000, 1.000]	[0.000, -0.000, 0.000]
3432.576	[-0.454, -0.220, -0.724]	[0.000, -0.140, 0.954, -0.265]	[-0.271, 0.074, -0.2610]
3433.575	[-0.454, -0.220, -0.724]	[0.000, -0.140, 0.954, -0.265]	[-0.271, 0.074, -0.2610]
3434.476	[-0.454, -0.220, -0.724]	[0.000, -0.140, 0.954, -0.265]	[-0.271, 0.074, -0.2610]

Table. 2. Maximum Torques

Axis	Maximum torque
A1 (J1)	200 Nm
A2 (J2)	200 Nm
E1 (J3)	100 Nm
A3 (J4)	100 Nm
A4 (J5)	100 Nm
A5 (J6)	30 Nm
A6 (J7)	30 Nm

4. Conclusion

This research work has presented a software package for manipulation and motion planning for real and simulated KUKA LWR/LBR4+ robotic arm. Based on the OMPL suit of MoveIT, KUKA arm gains some advanced features like an effortless way to define couple degrees of freedom and modify the planning accordingly, smooth parameterization for planners, the dynamic simulation and integration with task planner. The validation of the hypothesis via realization and real-world testing of such a framework was carried out.

Although the software framework works satisfactorily and is a sufficient proof for the concept, it has not yet been fully developed. As described in this paper earlier, to perform object detection and localization using visual feedback, we want to integrate the developed libraries of bilateral KUKA LWR platform with libraries of Microsoft Kinnect camera, Barrett hand and shadow hand. Further, we will plan variety of motions for bilateral KUKA LWR platform like palletizing application in industrial environment [11] [16]. Finally but not limited to, we will use Python programming language to develop graphical user interface for each joint of bilateral KUKA LWR platform [17].

AUTHORS

Tarwadi Parag Jitendra – School of Mechanical and Building Sciences, VIT University, Chennai Campus Chennai, India.

Arockia Selvakumar Arockia Doss*– Mechatronics Research Group, School of Mechanical and Building Sciences, VIT University, Chennai Campus, Chennai, Tamil Nadu-600127, India.
E-mail:arockia.selvakumar@vit.ac.in

Juan Antonio Corrales Ramon – Image, Systems of Perception, Robotics (ISPR), Institute Pascal, Clermont-Ferrand, France.

*Corresponding author

REFERENCES

1. T. Chettibi, H.E. Lehtihet, M. Haddad, S. Hanchi, "Minimum cost trajectory planning for industrial robots", *European Journal of Mechanics, Solids*, 23, 2004, 703–715.
http://dx.doi.org/10.1016/j.euromechsol.2004.02.006
2. L. Blackmore, B. Williams, "Optimal Manipulator Path Planning With Obstacles Using Disjunctive Programming", IEEE, 1–3, 2012.
http://dx.doi.org/10.1109/ACC.2006.1657210
3. John Vannoy and Jing Xiao, "Real-Time Adaptive And Trajectory-Optimized Manipulator Motion Planning", IEEE, 1–6, 2013.
http://dx.doi.org/10.1109/IROS.2004.1389401
4. Dmitry Chibisov, "Design of Algorithms for Motion Planning and Motion Prediction", *IEEE, Technical University of Munchen*, 2009, 1–73.
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.665.2817.
5. E.J. Solteiro Pires, P.B. de Moura Oliveira, J.A. Tenreiro Machado, "Manipulator Trajectory Planning Using A MOEA", *Applied Soft* Computing, 2006, 659–667.
http://dx.doi.org/10.1016/j.asoc.2005.06.009
6. Charles C. Kemp, Aaron Edsinger, Eduardo Torres, "Challenges For Robot Manipulation In Human Environments", *IEEE Robotics & Automation Magazine*, 2007, 20–29.
DOI: 10.1109/MRA.2007.339604.
7. Günter Schreiber, Andreas Stemmer and Rainer Bischoff, "The Fast Research Interface for the KUKA Lightweight Robot", ICRA, 2011, 15–21.
http://citeseerx.ist.psu.edu/viewdoc/download?DOI=10.1.1.471.3844&rep=rep1&type=pdf.
8. Yanyu Su, Yongzhuo Gao, Yan Wu, Wei Dong, Weidong Wang And Zhijiang Du, "YARC - A Universal Kinematic Controller For Serial Robots Based On PMAC And Moveit!", IEEE, 2014, 1–6.
http://dx.doi.org/10.1109/APSIPA.2014.7041708.
9. Asad Yousuf, William Lehman, Mir M. Hayder, "Introducing Kinematics with Robot Operating System (ROS)". In: *122nd ASEE Annual Conference and Exposition*, 2015, 1–18.
http://dx.doi.org/10.18260/p.24361
10. Stephen Hart, Paul Dinh, Kimberly Hambuchen, "The Affordance Template ROS Package For Robot Task Programming". In: *NASA Human Robotics Systems Project*, 2014, 1– 8.
http://dx.doi.org/10.1109/ICRA.2015.7140073
11. Mathew DeDonato, Velin Dimitrov, Ruixiang Du And Ryan Giovacchini, "Human-in-the-loop Control of a Humanoid Robot for Disaster Response: A Report from the DARPA Robotics Challenge", *Journal of Field Robotics*, 2015, 1–18.
http://dx.doi.org/10.1002/rob.21567.
12. Philipp Allgeuer, Max Schwarz, Julio Pastrana, "A ROS-based Software Framework for the NimbRo-OP Humanoid Open Platform". In: *13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, 1–6.
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.645.3441.
13. Dmitry Berenson, Pieter Abbeel, Ken Goldberg, "A Robot Path Planning Framework that Learns from Experience". In: *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, 3671–3678.
http://dx.doi.org/10.1109/ICRA.2012.6224742
14. Ioan A. Sucan, Lydia E. Kavraki, "The Open Motion Planning Library", *IEEE Robotics & Automation Magazine*, 2012, 1–10.
http://dx.doi.org/10.1109/MRA.2012.2205651
15. D. Coleman, I. A. Sucan, Sachin Chitta, N. Correll, "Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study", *Journal of Software Engineering for Robotics*, 2014, 1–14.
https://arxiv.org/abs/1404.3785.
16. Dirk Holz, Francesco Rovida, Sven Behnke, "A Skill-Based System for Object Perception and Manipulation for Automating Kitting Tasks". In: *20th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2015, 1–9. http://dx.doi.org/10.1109/ETFA.2015.7301453.
17. Jorg Stuckler, Angeliki Topalidou-Kyniazopoulou, Dirk Holz, "Real-Time Object Detection, Localization and Verification for Fast Robotic Depalletizing". In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2015, 1–8.
http://dx.doi.org/10.1109/IROS.2015.7353560
18. Stephan Kallweit, Robert Walenta, Michael Gottschalk, "ROS Based Safety Concept for Collaborative Robots in Industrial Applications", HAL Archives, 2016, 1–9.
http://dx.doi.org/10.1007/978-3-319-21290-6_3.

19. Marsette Vona and Shekar NH, "Teaching Robotics Software with the Open Hardware Mobile Manipulator", *IEEE Transactions on Education*, 2015, 1–6. http://dx.doi.org/10.1109/TE.2012.2218657.
20. Rhama Dwiputra, Alexey Zakharov Roustiam and Chakirov, "Modelica Model for the youBot Manipulator", *10th International Modelica Conference*, 1205–1212, 2014. http://dx.doi.org/10.3384/ecp140961205.
21. Dereck Wonnacott, Matias Karhumaa and James Walker, "Autonomous Navigation Planning with ROS", Michigan Technological University (Report), 2012, 2–12. http://cs.mtu.edu/~jwwalker/files/cs5881.pdf.
22. http://robohub.org/ros-101-intro-to-the-robot-operating-system/
23. http://moveit.ros.org/documentation/concepts.
24. http://robotnor.no/expertise/fields-of-competence/ motion-planning/
25. Kuka Lightweight Robot 4+ Operating Manual, 2012, 1–77.