TOOL MANAGEMENT USING ROBOT OPERATING SYSTEM*

BY

[1]Tasneem Khan♣, [2]Sheldon Remedios, [3]Kunj Parsania,

[4]Santosh Govindraj, [5]Ipsita Swain

[1,2,3,4]Ajeenkya DY Patil University, Pune, Maharashtra, India

Email: [1]tasneem.khan@adypu.edu.in, [2]sheldon.remedios@adypu.edu.in,

[3]kunj.parsania@adypu.edu.in,

[4]santosh.govindraj@adypu.edu.in, [5]ipsita.swain@adypu.edu.in

ABSTRACT

Manufacturing industries produce goods on a large-scale using machines, humans, and tools. Tools, like humans and machines, play an equally important role. It aids in the setup of a production line and the maintenance of machines. It is a time-sensitive asset. Mismanagement of tools could impede the production line and cost time. To address this issue, a pick and place 6-DOF robotic manipulator is proposed that uses vision to keep track of all the tools in its workspace and provide visual assistance in the form of messages using ROS. This is a common complaint in MSME industries that can be solved with the help of an intelligent automation system, saving the industry money and time.

KEYWORDS

Tool Management, ROS, Robotics, MoveIt.

I.INTRODUCTION

Robotics is a multidisciplinary branch of science and engineering concerned with using robots to replicate human actions in order to make life easier by solving complex problems and performing tedious tasks. It is a modern technology that has gained popularity in solving complex problems and is widely used in a variety of fields such as construction, healthcare, industrial manufacturing, and agriculture. Robotics is the branch, and its application is creating robots. The word 'robot' was coined by a famous playwright Karl Capek in his paradoxical drama Rossum's Universal Robots, which portrays robots replacing humans to perform monotonous work to pursue more creative interests [1]. Robots are becoming a more common part of people's daily lives.

* Received 22 September 2021, Accepted 09 October 2021, Published 24 October 2021

♣ Corresponding Author

There are various categories of robots available and can be classified into three aspects based on their application.

Robotic manipulators, which are most commonly used in industrial settings to manipulate objects of interest.

Drones, used for search and rescue, delivery and videography.

Gathering data, processing data, and taking action based on the data.

In the three categories defined above, the primary focus will be on the robotic manipulator. Robot Institute of America defines a robotic manipulator as "A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of task" [2].

A manufacturing industry involves the production of products for consumer consumption using labour, machines and tools. Machines, tools, and labour are critical in ensuring that products are manufactured on time and of high quality. The labour manages the entire production process and ensures that the machines run smoothly. Machines are used to efficiently manufacture products in less time while wasting as little material as possible. Repairing faulty machines requires the use of tools. Machines play an essential role in ensuring the smooth manufacturing of products.

If a machine fails, it disrupts the entire production line; to assess the situation, labour will follow the assigned protocols and repair the machine with tools. The tools assist the labour in performing a specific task to fix the machine and restart the entire production line. As a result, it is critical to recognise that tools, like machines and labourers, play an equally important role in the manufacturing process. According to research, tools account for 14% of total production costs [3]. Screwdrivers, spanners, hammers, and other tools are required to repair machines and check their working condition regularly. When a specific tool is not available when it is demanded, it can impede the production line. This is a common issue in industries where tools are frequently misplaced and are not found in their appropriate location when needed. A standard solution to this problem has been discovered by appointing a supervisor to regulate tool usage while taking into account the time and the person who has taken the tool and other methods involving automation. The following paragraph talks about it.

Sharan et al. [4] worked on a vision system that can automatically classify tools work pieces according to their shape and colour and also determines the work piece’s location for manipulation. For vision purpose an overhead Charge-Coupled Device (CCD) camera was used. A single image is obtained by the computer which uses image processing to determine the number of corners which helps to determine the shape of the object. Whereas hue, saturation and value of colours are used to determine the colour of the work piece. Eversheim et al. [3] explained the importance of tool management in the near future. The need of the right tool at the right place and the right moment is necessary for uninterrupted production. Tools also make up 14% of the production cost. Akturk et al. [5] proposed an algorithm that helps minimize total production costs. The study focuses to solve tool management and scheduling problems. The algorithm helps identify optimum machining conditions and specific tool allocations for all operations that in turn minimizes the total manufacturing costs. Kwon et al.

[6] focused on the development of a tool wear index (TWI), a surface roughness control model, a tool life model for an extended use of the worn-up tool and the development of an optimal control strategy. The production costs can be minimized by using an optimal control strategy by adjusting machining parameters and extending the tools usage within the constraints for certain machining conditions. Rather than discarding the tool after the tool is worn out, the proposed method allows continued use of the tool by monitoring the TWI.

The solutions provided for tool management are done through image processing, where the tools are detected and there are scheduling tasks that are followed by the industries; the problem with that is human involvement. There are many issues with tool management when humans are involved in scheduling and returning tools to their original location, and while there are systems that detect the tool, a complete solution has not been provided. Using the simulation software ROS and the tools it provides, a more appropriate and cost-effective method has been proposed. This study proposes a 6 degree of freedom (DOF) articulated arm configuration robotic manipulator with a mechanical gripper that detects objects of interest and reports their location to workers as feedback so that they can understand the tool's location, reducing search time and machine idle time. Since building a robotic manipulator is an expensive task with a high level of complexity, Robot Operating System (ROS) is the best option [7].

II.DEVELOPMENT OF COMPUTER AIDED DESIGN MODEL USING SOLIDWORKS

Mechanical design necessitates knowledge from various disciplines, including machine design, structural design, mechanical engineering, and electrical engineering. For the development, assembly, and analysis of the manipulator, Solidworks 2018 vSP5 is used, which is published by Dassault systems that runs primarily on windows. In Solidworks, model creation typically begins with creating a 2D sketch comprised of geometry such as lines, arcs, rectangles, and splines. Attributes such as parallelism, tangency, and concentricity are defined using constraints. Dimensions are added to the geometry to define its location and size. Because it is parametric software, the geometry is driven by relationships and dimensions. A 2D sketch is given a third dimension using extrusion, revolve, and sweep for detailing such as drills, holes, and ribs tools such as cut-extrude hole- wizard and rib. Assembly and subassemblies are created after the 3D models have been completed. Individual components relationships to assemblies are given using mates such as concentricity, coincidence, parallelism, perpendicular, distance-offset, angle- offset, tangency, and fix. Figure 1 shows the robot manipulator designed in Solidworks.

Figure 1: Robot Manipulator

III.GENERATION OF UNIFIED DESCRIPTION ROBOT FORMAT

Unified Robot Description Format (URDF) is the Domain Specific Modelling Language (DSML) used by ROS to read Computer Aided Design (CAD) files of manipulators. It stores information in .xml format about the robot's layout, appearance, and extra information such as links and joints of manipulators [8]. For the time being, CAD models can be converted to URDF using the software Solidworks and Fusion

360. Solidworks was used for the same purpose. The URDF model is shown in Figure 2.

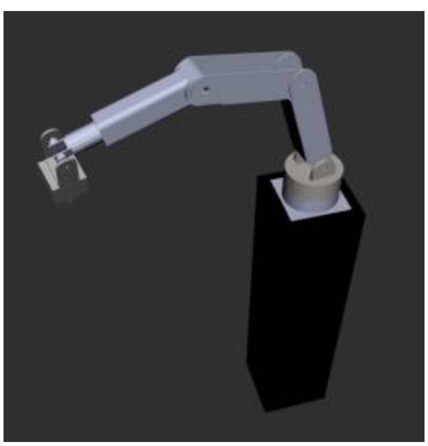

Figure 2: URDF model of manipulator

IV.GENERATING A PACKAGE WITH THE HELP OF MOVE IT SETUP ASSISTANT

MoveIt is the motion planning framework that provides essential capabilities for a robotic manipulator to function in an unstructured environment or to perform other desired applications in an industrial or commercial setting. MoveIt works by leveraging the capabilities of the ROS framework and utilising ROS core tools such as the ROS Visualizer, also known as RViz, and the ROS robot format URDF. It evolved from the Arm Navigation framework, which was initially developed for the PR2 robot. It was later integrated with ROS to broaden the scope of functionality available to other people who wanted to use it for their robotic application as a manipulation framework. MoveIt is a library that contains tools and features that allow a robotic manipulator to perform functions such as kinematics, motion planning, manipulation, 3D perception, trajectory processing, control, and navigation. All these functions mentioned can be easily added as a feature to the robotic manipulator using MoveIt Setup Assistant [9, 10].

The MoveIt setup assistant is a graphical user interface (GUI) as shown in Figure 3 that allows the user to import a URDF or Collada model and generate a package used in application. With the help of the setup assistant, features can be easily added to the URDF model. Collision checking, robot control, joints, and 3D perception are among the new features added to the URDF model. This package now can be easily used for simulation in Gazebo, visualisation and interaction purposes in RViz. It is effortless to use the GUI, click buttons, and get the result in the form of a package.

Figure 3: MoveIt setup assistant

Some of the essential features that can be added to the robot using MoveIt setup assistant is explained below. These are not the only available feature there are many more, the one's essential to the application have been explained over here.

i.Collision Checking

Collision Checking is part of any applications in robotics. It is an essential element that enables to navigate the environment without interacting with the robotic manipulator itself and the surrounding objects. The default Collision Checking plugin used by MoveIt is the FCL (Flexible Collision Library). FCL allows us to do collision checking and proximity computation using different algorithms [9, 11].

ii.Kinematics

Kinematics is the branch of mechanics that deals with the body's motion and does not consider the body's forces. Kinematics concerning robotics can be divided into two categories, Forward Kinematics and Inverse Kinematics. Given the robotic manipulator joint values, forward kinematics is used to determine the end-effector position in 3D space. MoveIt has a wide variety of solvers and a native implementation of forward kinematics [9, 12].

Given the end-effector position in 3D space, inverse kinematics is used to determine the joint position of the robotic manipulator. For this functionality, there are numerous solvers available. The TRAC-IK Kinematic solver is implemented in the application. TRAC-IK Kinematic Solver is an inverse kinematics solver developed by TRAC Labs that combines two IK implementations via threading to achieve more reliable solutions than standard available open-source IK solvers. From their documentation, "TRAC-IK concurrently runs two IK implementations. One is a simple extension to KDL's Newton-based convergence algorithm that detects and mitigates local minima due to joint limits by random jumps. The second is an SQP (Sequential Quadratic Programming) nonlinear optimisation approach that uses quasi-Newton methods to handle joint limits better [13].

iii.Motion Planning

The ability of a robot to move in an environment without colliding with the objects in its surroundings is referred to as motion planning. Motion Planning is a fundamental concept in robotics that ROS can solve with its default solver, the Open Motion Planning Library (OMPL). OMPL contains many sampling- based practical algorithms to be integrated with any software, not only ROS. OMPL's planners are abstract; that is, OMPL has no concept of a robot. MoveIt, on the other hand, configures OMPL and provides the back-end for OMPL to work with Robotics problems [9, 14].

iv.Trajectory Processing

Motion Planners are used to determine the best kinematic chain to achieve a specific goal. Because they lack timing information, a timing solver must be used in conjunction with the motion planning plugin to perform trajectory processing. The optimal solution of the motion planner can be found in a few seconds; typically, this message is sent when a successful path is found; trajectory processors perform the timing computation. The commonly used trajectory processing algorithms are Iterative Parabolic Time Parametrization and Time Optimal Trajectory Generation. [9, 15]. The package generated is shown in Figure 4.

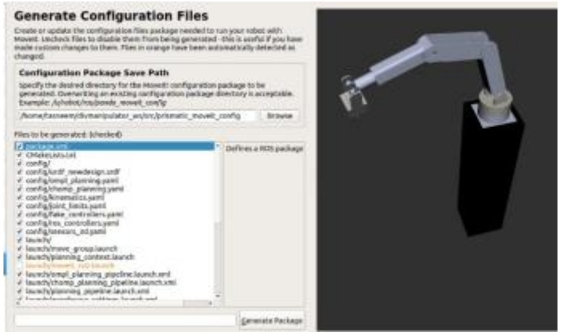

Figure 4: The package generated using MoveIt setup assistant

V.VISUALIZATION IN ROS

The MoveIt setup assistant was used to generate the package, which was then used to visualise the model in Robot Visualizer (RViz). Robot Visualizer is a 3D visualiser tool built on top of ROS that allows the user to conceptualise various sensor data related to the robot model. It also lets the user interact with the robot model and see how it reacts to its surroundings [9]. For example, if the robot model includes a camera, RViz can be used to visualise the camera data. If an IMU sensor is present, it can visualise the sensor's data, and different data from different sensors can be easily visualised using RViz. It is easier to visualise and interact with the robot model when using the MoveIt motion planning plugin in conjunction with RViz, as shown in Figure 5. RViz is used to access Move It features such as motion planning, trajectory planning, collision detection, and perception.

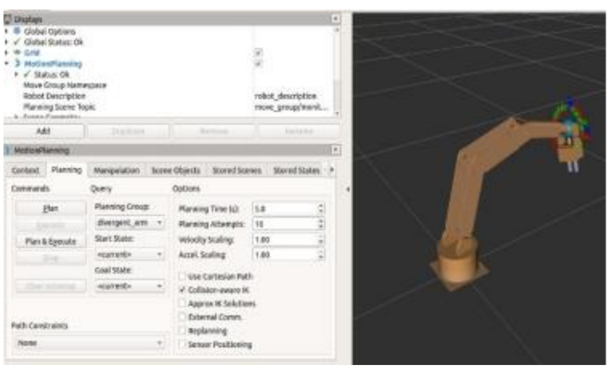

Figure 5: Visualization in RViz

RViz provides the user with a set of interactive markers, as shown in Figure. With the help of these markers, it can help to move the robot model to different orientations in the 3D space and plan out the path for the robot according to its desired location. It helps to visualise if the robot can follow the desired trajectory or not. RViz also helps to visualise the robot's start position, green in colour and the goal position, orange in colour, as shown in Figure 6.

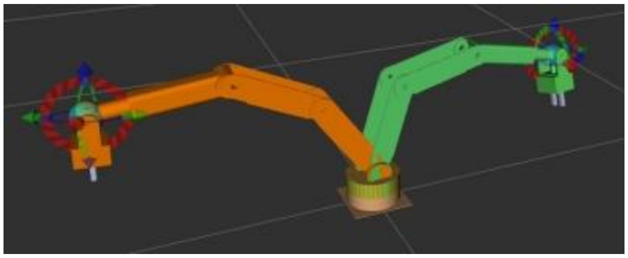

Figure 6: Start goal (green) and end goal (orange) visualization in ROS

VI.THE CONNECTION BETWEEN GAZEBO AND MOVEIT

Gazebo is a simulation programme. A roboticist's ability to simulate a robot in a virtual environment is a must-have tool. This is where Gazebo enters the picture. It allows to quickly test algorithms, design robots, and conduct testing. Gazebo enables robot simulation in complex indoor and outdoor environments to be accurate and efficient. Gazebo includes features such as dynamic simulations, advanced graphics, and the ability to add sensor plugins to the robot. Because Gazebo is a simulation environment, it is can be linked to MoveIt, which provides manipulation control, collision detection, kinematics, control, and navigation to the application. The following steps must be taken in order to establish a connection between Gazebo and MoveIt [16].

- Using the ros control package provided by ROS. It allows the user to provides PID values for each joint for the robot for smooth motion of the robot path.
- Adding hardware transmission to the URDF file.
- Adding controller structure to the MoveIt config file.
- Adding controller structure to the Gazebo config file.

MoveIt is used to control the model in Gazebo once everything is set up.

VII.ENVIRONMENT SETUP IN GAZEBO

The Figure 7 shows the environment set up of the application. The table is the place where worker will place the tools when they have finished working with it and the robot will place it back in its original position which is the bin. The robot is used for tool organization in this environment and the tools scattered on the table will be organized and kept inside the tool bin. But often due to mismanagement the tools are misplaced and the workers cannot find them. Therefore, using the robotic manipulator in such a way that it aids the worker in finding the tool as soon as possible using logical camera. The logical camera is also used to recognise objects in the environment. A logical camera is a type of simulation concept. A standard camera produces images, whereas a logical camera provides poses and model names. The logical camera detects and acknowledges objects at the intersection of its frustum and uses the frustum geometry to compute the object's pose in 3D space. It works by determining whether its frustum intersects a model's axis aligned bounding box. The logical camera has been set up above the table and the container for dropping the object shown in Figure 8. The cubes in the table represent the tools that have to be picked up and placed on the opposite side in the place bin behind the robotic manipulator.

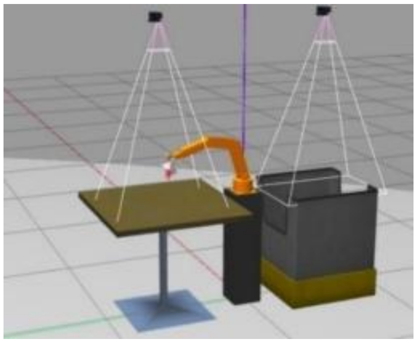

Figure 7: Environment Setup in Gazebo

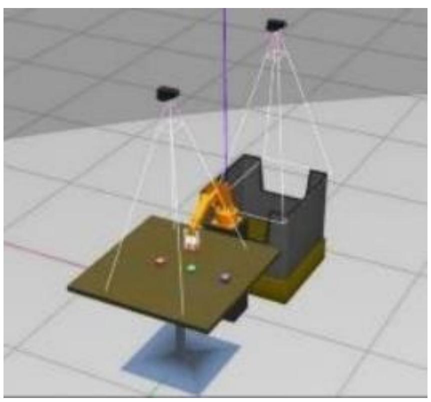

Figure 8: Setting up of objects in Gazebo

VIII.SIMULATION AND RESULTS

There are several approaches to solving the problem of tool management. This paper suggests a solution that is believed to be suitable. Using the robot manipulator, arrange the worker's tools that are lying on the table and place them in an organized way in the bin. To avoid wasting workers' time, a visual aid in the form of a message display is provided. ROS has been used to set up a node that is subscribed to the logical cameras 1 and 2. Various scenarios for tool management with ROS have been considered. The scenarios considered while resolving the tool management problem are depicted in the flowchart in Figure 9.

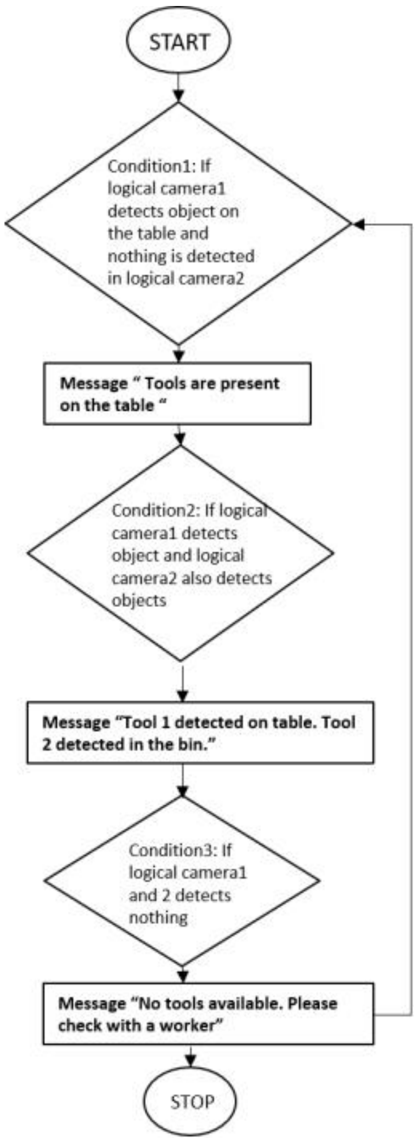

Figure 9: Flowchart of the code.

A simple pick and place pipeline was created for each of the conditions. The application MoveIt is used to plan the next steps. Using the poses previously configured in the MoveIt setup assistant, Gazebo can see how the robot plans its motion in the real-world simulation. It enables to determine whether the robot is performing as expected. A series of steps must be completed before the robotic manipulator can pick and place.

- Initially, the robot is in its home position.
- A safety check is performed after the robot to go to upright position to ensure that everything is connection between MoveIt and Gazebo is successful.
- Open the gripper.

- Because the pose and orientation of the object to be picked are already known via the logical camera, place the robot in the pick position using inverse kinematics.
- The pick position is always a few centimeters above the object to be picked up.
- This is regarded as acceptable practice.
- A message in the terminal confirms this. Once everything is in place, secure the object with the gripper.
- This is confirmed by a message in the terminal.
- The object has been selected, and the robot manipulator can now proceed to the location and place the tool in the bin.

The steps outlined above can be implemented in the tool management application, allowing the user to easily achieve the goal. The object can be picked and placed using this. The result of one of the conditions is shown in the following Figure 10.

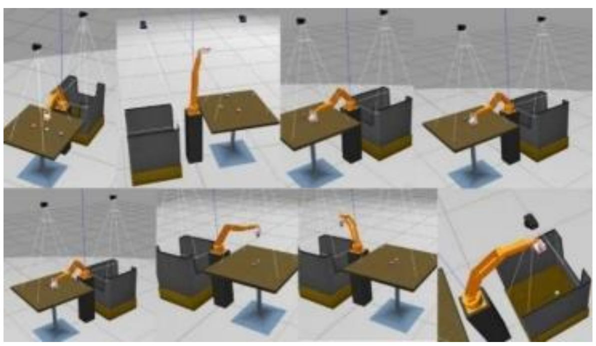

Figure 10: The simple pick and place pipeline

A message through the logical camera1 has been displayed as shown in Figure 11.

Figure 11: Output of logical camera1

CONCLUSION

Tool management may appear to be a minor issue, but ignoring it can have serious consequences. A variety of tools, as well as labour and machinery, are required. Robotic technology is used to solve the problem because everything in the manufacturing industry is automated or semi-automated. A simulation appears to be required before building a real-life model. Simulation is important because it allows to see how our robot performs in the real world; by seeing how the model works in the real world, hardware errors can be reduced. It allows to quickly test our algorithms while minimising model damage. It enables the users to see how our model interacts with different sensors. During the coronavirus outbreak, with everyone subject to quarantine rules, a real-world alternative was required. Simulation software, such as ROS, is critical in the development of robots in this scenario. If someone has a new idea for a robot, they can quickly create a model, add important features to it using MoveIt, and then deploy it in the real-world simulator known as Gazebo to see how the model works. With minimal effort, ideas can be generated, tested, deployed, and the results observed.

REFERENCES

[1]D. B. Camarillo, T. M. Krummel, and J. K. Salisbury, Jr., "Robotic technology in surgery: past, present, and future," The American Journal of Surgery, vol. 188, no. 4A Suppl, pp. 2S- 15S, Oct 2004.

[2]D. M. Considine and G. D. Considine, Standard Handbook of Industrial Automation. 1986, p. 472.

[3]W. Eversheim et al., "Tool Management: The Present and the Future," CIRP Annals, vol. 40, no. 2, pp. 631-639, 1991.

[4]R. Sharan, G. J. I. J. o. I. Onwubolu, Graphics, and S. Processing, "Automating the process of work-piece recognition and location for a pick-and-place robot in a SFMS," vol. 6, no. 4, p. 9, 2014.

[5]M. S. Akturk and S. Ozkan, "Integrated scheduling and tool management in flexible manufacturing systems," International Journal of Production Research, vol. 39, no. 12, pp. 2697- 2722, 2001.

[6]Y. Kwon, G. W. J. I. J. o. M. T. Fischer, and Manufacture, "A novel approach to quantifying tool wear and tool life measurements for optimal tool management," vol. 43, no. 4, pp. 359-368, 2003.

[7]M. Q. B. Gerkey, "ROS: an open-source Robot Operating System."

[8]Y. Kang, D. Kim, and K. Kim, "URDF Generator for Manipulator Robot," presented at the 2019 Third IEEE International Conference on Robotic Computing (IRC), 2019.

[9]S. Chitta, "MoveIt!: An Introduction," in Robot Operating System (ROS)(Studies in Computational Intelligence, 2016, pp. 3-27.

[10]S. Chitta, I. Sucan, and S. Cousins, "MoveIt! [ROS Topics]," IEEE Robotics & Automation Magazine, vol. 19, no. 1, pp. 18- 19, 2012.

[11]J. Pan, S. Chitta, and D. Manocha, "FCL: A general purpose library for collision and proximity queries," in 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 3859-3866: IEEE.

[12]A. Yousuf, W. Lehman, M. A. Mustafa, and M. M. Hayder, "Introducing kinematics with robot operating system (ROS)," in 122nd ASEE Annual Conference and Exposition, 2015, pp. 1-18.

[13]P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), 2015, pp. 928-935: IEEE.

[14]I. A. Sucan, M. Moll, L. E. J. I. R. Kavraki, and A. Magazine, "The open motion planning library," vol. 19, no. 4, pp. 72-82, 2012.

[15]W. Khalil and E. Dombre, "Trajectory generation," in Modeling, Identification and Control of Robots, 2002, pp. 313- 345.

[16]S. Chitta et al., "ros_control: A generic and simple control framework for ROS," The Journal of Open Source Software, vol. 2, no. 20, 2017