An Analytical Approach to Safe Collaborative Assembly Station with Cobots

by

Hamed Montazer Zohour

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT OF A MASTER'S DEGREE
WITH THESIS IN MECHANICAL ENGINEERING
M.A.Sc.

MONTREAL, DECEMBER 21, 2022

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

M. David St-Onge, Thesis supervisor
Department of mechanical engineering, École de technologie supérieure

M. Vincent Lévesque, Chair, Board of Examiners
Department of Software Engineering, École de technologie supérieure

M. Jean-Philippe Roberge, Member of the Jury
Department of Systems Engineering, École de technologie supérieure

M. Maarouf Saad, Member of the Jury
Department of Electrical Engineering, École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON "DECEMBER 14, 2022"

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I would like to thank all the people who have contributed directly or indirectly to undertaking this research project and without whom I would not have been able to complete this research. First of all, I would like to express my deepest appreciation to my research supervisor, Professor David St-Onge, for his consistent support, guidance, and encouragement during the running of this project. I would also like to extend my gratitude to all the members of the INIT Robots laboratory. They supported me in different stages of the development of the solution, conducting the experiments, and in the daily work throughout my master's degree.

In addition, I would like to acknowledge the CoRoM program funded by NSERC and partner organizations for their financial support throughout my master's degree. Furthermore, special thanks to the Kinova company which was the industrial partner of my research project and gave me a chance to complete my internship. The support of these two organizations enabled me to carry out my research and write this dissertation.

To conclude, I cannot forget to thank my family and friends for all the unconditional support in these very intense academic years.

Une approche analytique d’un poste d’assemblage collaboratif sécurisé avec des cobots

Hamed Montazer Zohour

RÉSUMÉ

La robotique est une science interdisciplinaire englobant de vastes domaines de recherche : vision, planification du mouvement, contrôle, locomotion, design, etc. L’objectif de la création des robots au début des années soixante était de soulager l’homme de certains travaux pénibles tels que : la manipulation d’un objet lourd, et des tâches répétitives souvent fatigantes voire parfois irréalisables manuellement. Suite à cette situation, plusieurs sortes de manipulateurs ont été créés. Historiquement, les bras manipulateurs ont été les premiers fabriqués et sont encore largement utilisés dans l’industrie. Ces systèmes robotisés ont la capacité d’agir sur l’environnement par la réalisation de tâches de manipulation telles que la préhension d’objets, l’assemblage de pièces, etc. Ils sont néanmoins très limités dans leur espace de travail opérationnel et dans le type de travail qu’ils peuvent réaliser.

De nouvelles applications où les robots sont employés à proximité des humains se développent rapidement. Contrairement aux robots industriels, qui sont utilisés dans des cages pour garantir une grande précision et sécurité, les robots collaboratifs, également appelés Cobots, sont conçus avec un haut degré de conformité pour assurer la sécurité en présence de l’homme.

Plusieurs applications industrielles de pick-and-place, telles que les chaînes de montage collaboratives, reposent sur le suivi visuel des pièces. Les occlusions récurrentes causées par le mouvement du manipulateur diminuent la productivité de la ligne et peuvent provoquer des pannes. Ce travail fournit une solution complète pour maintenir une ligne de visée sans occlusion entre une caméra à pose variable et l’objet à saisir par un manipulateur 6R non cloisonné au poignet. Nous considérons les occlusions potentielles par le manipulateur ainsi que par l’opérateur travaillant au poste d’assemblage. Une caméra actionnée détecte l’objectif de l’objet (pièce à prélever) et suit l’opérateur. L’approche consiste à utiliser l’ensemble complet des solutions obtenues à partir de la dérivation de la solution de l’équation polynomiale univariée à la cinématique inverse (IK). Par rapport aux méthodes de résolution itérative numérique, notre stratégie nous accorde un ensemble de positions articulaires (posture) pour chaque racine de l’équation dont nous extrayons le meilleur (minimisant les risques d’occlusion). Notre méthode analytique, intégrant des optimisations d’évitement de collision et d’occlusion, peut contribuer à améliorer considérablement l’efficacité et la sécurité des postes de travail d’assemblage collaboratifs.

Mots-clés: robotique; cinématique; tâches collaboratives ; minimisation de l’occlusion

An Analytical Approach to Safe Collaborative Assembly Station with Cobots

Hamed Montazer Zohour

ABSTRACT

Robotics is an interdisciplinary science encompassing vast fields of research: vision, motion planning, control, locomotion, design, and so on. The objective of the creation of robots in the early sixties was to relieve man of certain hard jobs such as: handling a heavy object, and repetitive tasks often tiring or even sometimes infeasible manually. Following this situation, several kinds of manipulators were created . Historically, the manipulator arms were the first manufactured and still are widely used in industry. These robotic systems have the ability to act on the environment through the realization of manipulation tasks such as the grasping of objects, the assembly of pieces, etc. They are nevertheless very limited in their operational workspace and in the type of work they can achieve.

New applications where robots are employed near humans are growing rapidly. Unlike industrial robots, which are used in cages to guarantee high precision and safety, collaborative robots, also called Cobots, are designed with a high degree of compliance to ensure safety in presence of humans.

Several industrial pick-and-place applications, such as collaborative assembly lines, rely on visual tracking of the parts. Recurrent occlusions caused by the manipulator motion decrease line productivity and can provoke failures. This work provides a complete solution for maintaining an occlusion-free line of sight between a variable-pose camera and the object to be picked by a 6R manipulator that is not wrist-partitioned. We consider potential occlusions by the manipulator as well as the operator working at the assembly station. An actuated camera detects the object goal (part to pick) and keeps track of the operator. The approach consists in using the complete set of solutions obtained from the derivation of the univariate polynomial equation solution to the inverse kinematics (IK). Compared to numerical iterative solving methods, our strategy grants us with a set of joint positions (posture) for each root of the equation from which we extract the best (minimizing the risks of occlusion). Our analytical-based method, integrating collision and occlusion avoidance optimizations, can contribute to greatly enhancing efficiency and safety of collaborative assembly workstations.

Keywords: robotics; kinematics; collaborative tasks; occlusion minimization

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

ETS	École de Technologie Supérieure
PPO	pick-and-place operations
IKP	Inverse Kinematics Problem
HRC	Human-Robot Collaboration
PRM	Probabilistic Roadmap
RRT	Rapidly-exploring Random Tree
GO	Goal-Oriented
CHOMP	Covariant Hamiltonian Optimization for Motion Planning
STOMP	Stochastic Trajectory Optimization for Motion Planning
OMPL	Open Motion Planning Library
EE	End-Effector
DH	Denavit-Hartenberg
ROS	Robot Operating System
SDK	Software development kit
AL	Artificial intelligence
RGBD	Red, Green, Blue, Depth

INTRODUCTION

Robotic manipulators can be found in a wide range of industrial applications, namely to conduct pick-and-place operations (PPO). Moreover, the introduction of collaborative robots (cobots) in assembly lines to conduct PPOs usually comes with the need to track the parts with cameras. In these conditions, visual occlusion can reduce productivity and even cause assembly line failures. A system coping with occlusion is essential to reduce potential hazards (Zhu *et al.*, 2021b), (Wada, Okada & Inaba, 2019b). The goal of this work is to provide a complete automated solution for commercial collaborative manipulators. Indeed, cobots are increasingly used in small and medium businesses globally every year (of Robotics, 2017). This is partially explained by their ease of installation and use, as well as the reduced initial investment required. As explained by Bortolini, Ferrari, Gamberi, Pilati & Faccio (2017), Industry 4.0 puts cobots at the core of manufacturing lines. Since they are low-cost, easy to set up, easy to program, and safe to use, they can increase the flexibility of production systems. Moreover, they can also expand automation for the purpose of the new application (Bloss, 2016).

A symbolic solution to the inverse kinematics problem (IKP) is a powerful tool to achieve versatile control. The vast majority of industrial manipulators with 5 or 6 revolute joints (commonly referred to as 5R and 6R) are said to be wrist-partitioned (such as the Kuka KR15 and ABB IRB). With these manipulators, we can obtain easily a closed-form solution to their IKP. However, specific conditions must be met so the inverse kinematics of 6R serial manipulators can be decoupled and these conditions are hard to combine with collaborative manipulators characteristics. Conditions on architecture parameters to solve the orientation and positioning problem separately often lead to wrist joint analog to a spherical configuration (Pieper, 1968). This condensed configuration of the wrist is complex to fully enclose, such as to prevent any finger of a user to be trapped or pinched. This presents a major limitation for collaborative operations. Some robots, such as the Kinova Gen3 series represented in Fig. 3.1 by the Gen3 lite model, are designed to optimize safety and the reachable workspace but are more complex non-wrist-partitioned manipulators architecture.

The complexity of the kinematics of these manipulators steers most of the research energy over collision-free trajectory planning (Chen & Song, 2018), (Ahmed Zaki, Mohamed Fathy, Carnevale & Giberti, 2022) and compliant grasping techniques (Babin, St-Onge & Gosselin, 2019). The numerical solvers used in these works do not provide enough flexibility and cannot guarantee that a solution will be found. In this work, we proposed a complete solution to minimize tracking occlusion for collaborative pick-and-place tasks.

On the perception side of the problem, several works cope with adaptive sensing for object tracking with occlusion. Many consider *occlusions inevitable*, and rather suggest approaches based on multimodal sensing, for instance with sensitive finger tips (Zhu *et al.*, 2021a). Learning strategies may also help reconstruct known objects from partial images (Wada, Okada & Inaba, 2019a). With an occlusion-free strategy, these works will serve to enhance robustness to occlusion from the environment only. As this work calls for several bodies of knowledge, a review of the related work is divided into subcategories in the following section.

The flow chart detailing the procedure is shown in Fig. 0.1. In block 1, we detect the object to grasp as detailed in section 5.2. If it fails, we move the camera (block 2), hook as a robotic manipulator end-effector, to another pose (section 5.2). If the camera is able to detect the target, its pose is used to compute the IK polynomial roots (block 3). We leverage a methodology first introduced by Gosselin and Liu (Gosselin & Liu, 2014) to obtain a univariate polynomial equation for the IKP (Sections 5.1 and 3.1), giving us all possible postures (joint space) of the robot for the same end-effector pose (Cartesian space). Section 3.3.1 presents a validation of the IKP solution with examples and compared with the solutions obtained with a numerical IKP solver. In block 4, we select the best solution to avoid occlusion between the actuated camera and all known objects in the workspace (section 4.1). Among these obstacle, the operator is

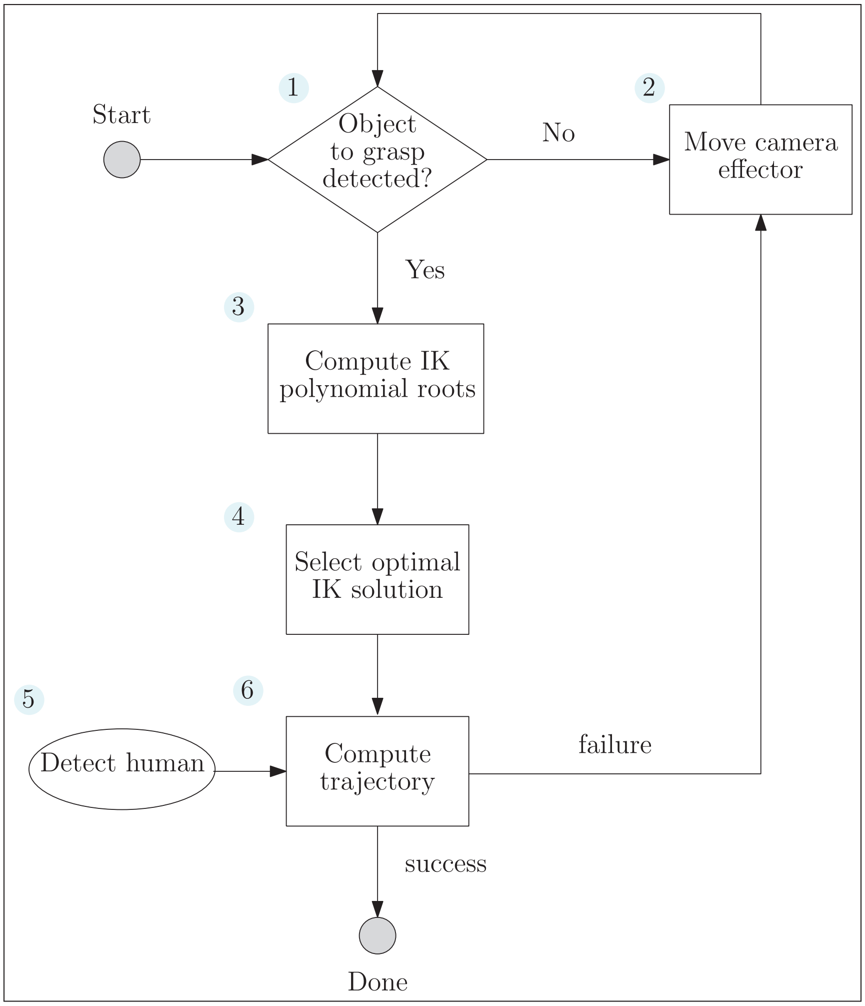


Figure 0.1 Flow chart of our occlusion-free pick-and-place solution for collaborative assembly tasks

tracked by the camera (block 5, section 5.2). Finally, a path-planner is used to compute the trajectory between the original and final (optimal) posture, taking into account the occlusion and obstacles (block 6, section 4.2). If the planning fails, we change the camera point-of-view (block 2), and start the previous steps again. The proposed methodology was validated in simulation and with experiments (section 5.3). The Python script used to solve the IKP, compute all real solutions (postures) and select the optimal posture is available online (Montazer Zohour, Belzile & St-Onge, 2021).

CHAPTER 1

LITERATURE REVIEW

1.1 Full-stack Solutions and Industrial Approaches

Transitioning from the standard isolated industrial robots setup to shared space and task between workers and robots brings several safety risk and performance concerns. It is often considered difficult to automate an assembly process crowded with workers, product parts, and tools. Therefore Cherubini, Passama, Crosnier, Lasnier & Fraisse (2016) observe that collaborative robotic systems arouse interest on heavy and laborious tasks while allowing humans to work on more value-added task. Hanna, Götvall, Ekström & Bengtsson (2018) studied an industrial application for which autonomous robotic systems were alongside manual assembly lines. They highlight several challenges and requirements related to safety, intuitive interaction, adaptation to variable processes, and the need for highly flexible communication and control. Similar results were discussed by Marvel, Falco & Marstio (2014). Using the manual assembly station from that study as a starting point, Hanna, Bengtsson, Götvall & Ekström (2020) then focused on safety aspects of a cobot station. They conduct a risk assessment and then apply several risk reduction measures to increase workers safety as well as efficiency and flexibility. While a trained worker can safely use a complex and ultimately dangerous robotic system, the transition of the industry requires additional safety measures. In this context, a robot integrator must think outside the current standards and guidelines. Hanna *et al.* (2020) suggest to consider a new collaboration mode: deliberative planning and acting. To support the transition, Ogorodnikova (2009) introduces danger/safety indices that indicate the level of risk during an interaction with a robotic system. The indices are based on the characteristics of the robot and the operator physical and cognitive capacities. He stressed that the system must be intuitive, easy to use for the worker, in order to be safe. With this in mind, we designed an interaction modality centered on the worker safety, that does not increase the actions (commands) of the worker.

The vast majority of research and industrial use case study on the transition from manual to cobot-equipped assembly stations present the need to reduce the physical risk for the worker. These studies, such as the one of Salunkhe, Stensöta, Åkerman, Berglund & Alveflo (2019), focus on decreasing ergonomic issues, maintaining or decreasing cycle time at the station, and maintaining or increasing product quality. However, they do not cover how the workers interact with the system, they rather split the tasks and maintain (close) distinct work spaces. As well, Hanna *et al.* (2019) discuss the current challenges with planning and preparation processes for traditional final assembly and the requirement and possible solutions when implementing collaborative and intelligent automation systems. This experience proves that integrated virtual planning and preparation is the key and the prerequisite for this chain. Both engineers and operators need new knowledge in order to develop and execute these collaborative and intelligent automation systems. This control system must be modeled and analyzed early in the design process according to well-defined formal requirements. Marvel *et al.* (2014) state that collaborative operations can lead to hazardous situations due to the close proximity of humans and robots. In order to develop collaborative applications safely, hazardous situations must be controlled.

On the hardware side, cobots can be designed specifically for safe collaboration such as UR10 (Uni, 2022) and KUKA iiwa (Kuk, 2022): they detect collisions with any part of their structure, carry a smaller load, and have a shorter reach. The last two attributes may enhance safety, but they limit their application. Gopinath, Ore, Grahn & Johansen (2018) argue that close collaboration with large industrial robots can be safe and they show two experimental workstations. The key is to better understand the task, the operator and thus how to make stations safe. Smart control strategy tailored to a good understand of the application is also what Shadrin *et al.* (2020) leveraged to increase safety by modeling the objects and environment. Researches demonstrate the need for smart adaptive solutions to human-robot collaboration (HRC) in assembly lines. Our solution provides a flexible and optimal way to avoid any collision and ensure a safe collaboration.

1.2 Serial Manipulators

Inverse kinematics is the inverse process of the forward kinematics and one of the fundamental tools required to control the motion of robotic manipulators in particular positions and orientations.

Inverse kinematic methods generally fall into two categories namely closed-form and numerical Singh, Kukshal & Yadav (2021). Numerical methods provide only one solution and are unstable near singular configurations Waldron, Waldron, Schmiedeler & Schmiedeler (2007). The closed-form solution method utilizes robot-specific geometry to formulate the mathematical model based on an analytic expression. A closed-form solution can be classified further into two types: algebraic and geometric. The algebraic method reduces a transcendental equation into a single variable by using significant equations containing joint parameters. Then the joint variables are determined by mathematical calculations (Asfour & Dillmann, 2003; Ho, Kang & Lee, 2012). Researchers have therefore been interested in the polynomial solution of the inverse kinematic problem of 6R serial manipulators. According to Lee & Liang (1988), there are 16 possible solutions to the inverse kinematic problem of 6R manipulators of general architecture. A similar approach was proposed by Raghavan & Roth (1993) to obtain the univariate 16th-degree polynomial. The algorithm was implemented by some researchers based on well-known algorithms such as eigenvalue problems to make it more efficient and stable.

Polynomial solutions for different manipulators can be found in the literature (Manseur & Doty, 1989; Gosselin & Liu, 2014) with similar methodologies as the one described in this work. Considering 16th degree polynomial equations are prone to numerical ill-conditioning as well as the possibility of *polynomial degeneration* with roots yielding an angle of π, Angeles & Zanganeh (1993) proposed a semi-graphical solution to the inverse kinematics of a general 6R serial manipulator. However, these techniques do not apply to non-wrist-partitioned manipulators. Numerical methods have also been applied by several researchers (Chen, Yang & Kang, 1999; Aghajarian & Kiani, 2011; Duleba & Opalka, 2013), but these are commonly known to be prone to instability near singular postures. Moreover, they only give one possible solution, which may not be optimal. Several algorithms, including the ones proposed by Mavroidis, Ouezdou & Bidaud (1994), Husty, Pfurner & Schröcker (2007) and Qiao, Liao, Wei & Su (2010), can be found in the literature to find the 16th degress univariate polynomial equation for a 6R robotic manipulator, the latter notably using double quaternions.

1.3 Optimal Solution to the IKP

Symbolic solutions to the IKP, as the one presented above, mostly result in several viable configurations for a given end-effector pose. Thus, a strategy is required to select the best fitted solution; a single set of joint angles. A wide range of procedures can be used to select that optimal solution following the task (such as manipulating fragile objects) and the application context (such as low energy requirements).

Among the 16 solutions to the IKP, a wide range of methodologies has been proposed to select the best posture. As these solutions are theoretical, one must first discard the one that cannot be implemented: non real roots, exceeding joint limits or resulting in a self-colliding posture. From there, simple algorithms such as the minimization of the amount of joint rotation can easily be implemented. Task-dependent optimization can also be used for certain applications and performance indices based on the kinematics (eg., kinetostatic conditioning index) and the stiffness (eg. deformation evaluation index) of the robot Lin, Zhao & Ding (2017). (Guo, Dong & Ke, 2015) used a method based on the Jacobian matrix to solve the robot posture optimization model with the aim of increasing the stiffness of the robot in machining applications. Zargarbashi, Khan & Angeles (2012) reported posture-dependent indices based on kinetostatics to optimize the posture of a redundant robot for the given task. Our approach is task-related: preventing camera occlusions for pick-and-place tasks.

1.4 Trajectory Planning

From the set of joint angles for the manipulator goal, we need to derive the optimal path. These motions are typically synthesized to achieve functional goals, such as minimizing time, maximizing efficiency, and providing sufficient clearance around obstacles. Lozano-Perez, Jones, O'Donnell & Mazer (1992) were among the firsts to use the concept of task planning, and, since then, a large range of algorithms have been proposed. The initial focus of the motion planning research is concentrated on finding a complete planning algorithm, where an algorithm is said to be complete if it terminates in a finite time, returning a valid solution if one exists, and failure otherwise. Early work focused on finding trajectories that satisfy constraints imposed by the environment of the application, but they were not necessarily optimal. Yang, Pan & Wan (2019) provide a selection of optimal motion planning algorithms studied in terms of three main components: the decision variables, constraints, and objectives. The two most influential families of path planners are the sampling-based algorithms (Kavraki, Svestka, Latombe & Overmars, 1996; Kuffner & LaValle, 2000; Kang *et al.*, 2019) and the optimization-based ones (Ratliff, Zucker, Bagnell & Srinivasa, 2009; Kalakrishnan, Chitta, Theodorou, Pastor & Schaal, 2011; Park, Pan & Manocha, 2013). While the former is often more efficient for collision avoidance, the latter grants more flexibility on the optimization criteria.

To plan the trajectory for robots with high degrees-of-freedom (DoF), such as the industrial robots (usually six or seven DoFs) and mobile manipulators (usually more than 7 DoFs), one main contribution to the motion planning field was the development of sampling based algorithms (Kavraki *et al.*, 1996). The sampling-based planning algorithm is one of the most powerful tools for collision avoidance. Moreover, planners such as Probabilistic Roadmap (PRM) and Rapidly-exploring Random Tree (RRT) algorithms, along with their descendants, are now used in a multitude of robotic applications (Kavraki *et al.*, 1996; Kuffner & LaValle, 2000). Both algorithms are typically deployed as part of a two-phase process: first find a feasible path, and then optimize it to remove redundant or jerky motion. Kang *et al.* (2019) proposed a goal-oriented (GO) sampling method for the motion planning of a manipulator.

In that second family, Ratliff *et al.* (2009) proposed the Covariant Hamiltonian Optimization for Motion Planning (CHOMP): a novel method for generating and optimizing trajectories for robotic systems. Unlike many previous path optimization techniques, the requirement that the input path be collision-free is dropped. Kalakrishnan *et al.* (2011) presented the Stochastic Trajectory Optimization for Motion Planning (STOMP) using a series of noisy trajectories that can deal with general constraints. Otherwise, Park *et al.* (2013) developed a novel algorithm to compute real-time optimization-based collision-free trajectories in dynamic environments without the requirement for a prior knowledge about the obstacles or their motion. These algorithms and several others were integrated in the Open Motion Planning Library (OMPL) (Sucan, Moll & Kavraki, 2012). Our solution leverages these powerful contributions.

1.5 Object Detection and Pose Estimation

In this section, we are going to discuss the state-of-the-art of locating objects automatically to pick and avoid collision in collaborative assembly stations which is one of our challenges in running the experiments. Computer vision relies heavily on object detection for a variety of applications. The purpose of object detection is to locate and classify objects using rectangular bounding boxes. The detection of objects is related to the classification of objects, the semantic segmentation of objects, and the segmentation of instances. Objects detection is an important area of computer vision and has many practical applications in industrial production and scientific research, such as face detection (Taigman, Yang, Ranzato & Wolf, 2014), text detection (Zhang *et al.*, 2016b)), pedestrian detection (Zhang, Lin, Liang & He, 2016a), logo detection (Hoi *et al.*, 2015), video detection (Kang *et al.*, 2017), vehicle detection (Fan, Brown & Smith, 2016), and medical image detection (Litjens *et al.*, 2017).

There are two major approaches to object detection at present: traditional machine learning and deep learning. Traditional object detection algorithms include DPM (Felzenszwalb, Girshick, McAllester & Ramanan, 2010), Selective Search (Uijlings, Van De Sande, Gevers & Smeulders, 2013), Oxford-MKL (Vedaldi, Gulshan, Varma & Zisserman, 2009), and NLPR-HOGLBP (Yu *et al.*, 2010), etc. R-CNN and YOLO are the two main schools of deep learning algorithms. A key neural network in computer vision is Convolutional Neural Network (CNN) which was proposed by Girshick, Donahue, Darrell & Malik (2014). This algorithm, deep learning is applied to detecting objects for the first time.

In order to ensure accurate, robust localization and 6D estimation of objects, fiducial markers are commonly used. Markers contain a fixed square border and coding/decoding systems with various IDs (Garrido-Jurado, Muñoz-Salinas, Madrid-Cuevas & Marín-Jiménez, 2014). ARToolkitPlus, AprilTag, and Aruco are some of the most commonly used fiducial markers for accurate localization. AprilTags are used in robotics for a variety of functions, including calibration of cameras and robotic localization.

A number of articles have been published on object detection and pose estimation which includes high-performing algorithms for both techniques. In terms of industrial applications of the two technologies, state-of-the-art is more limited, but still, a few papers have tested them. Girshick *et al.* (2014) introduced the use of CNN with region proposal techniques to increase the speed of data processing, while still maintaining precision, toward real-time object localization and classification. The development of real-time object detection with high precision has been proposed by Redmon, Divvala, Girshick & Farhadi (2016). As one of the pioneers of real-time object detection, Redmon *et al.* (2016) has made significant contributions to this field. The paper by Li, Huang, Li & Huang (2019) presents an application of CNN based object detection in logistics warehouses. As well, Dwivedi, Gorjup, Kwon & Liarokapis (2019) has proposed a method for intuitive and affordable telemanipulation using EMG and fiducial marker-based pose tracking to control the robot arm-hand system naturally without limiting the hand movement. Eventually, due to the ability of fast processing and more reliable positioning, we will run our experiments based on fiducial markers, specifically Apriltag for the purpose of object detection and localization while their sensitivity to features in the environment based on the physical setup of the environment still remains as a weakness.

CHAPTER 2

METHODOLOGY

In this chapter, in order to fulfill research objectives and according to the state-of-the-art we have provided in first chapter, we have extracted the research opportunities and have addressed what is lacking in our problem. Then we present the hardware and software that the proposed approach will be implemented on them to achieve our objective. Eventually, the method and test plan will be presented.

2.1 Problem Statement and Research Objectives

Collaborative robots is a new technology poised to revolutionize manufacturing. To ensure safety, traditional robots perform work isolated from human workers. In contrast, collaborative robots are designed with inherent safety relying on lightweight materials, rounded edges, and the limitations of speed and force, or by sensors and software that guarantee safe behavior, so they can work alongside humans as true partners. In this project, we aim to provide a complete automated solution to minimize tracking occlusion in collaborative non-wrist-partitioned manipulators based on an analytical approach to conduct a pick-and-place operation in manufacturing assembly stations.

Our literature review showed a gap in the inverse kinematics solution of the non-wrist-partitioned manipulators. Several techniques have been applied by researchers but those are well-known to be prone to instability near singularity and also given one possible solution, so we intend to use our analytical-based method which consistently provides several viable sets of joint positions for a given end-effector. In addition, the literature review also shows a wide range of methodologies have been proposed to select the best posture and collision-free trajectory. Our approach is task-related and prevents camera occlusions for pick-and-place tasks. Also, we leverage OMPL trajectory planning which, compared with other methods, provides occlusion and collision-free path. Eventually, as we have stated in the literature review chapter specifically in full-stack solutions for manipulators deployment in cooperative tasks, Many researchers enhance safety in collaborative assembly stations by limiting their application. But as we have addressed, the key to have a safe collaboration with the cobot is to better understand the task and the operator, thus our solution provides a flexible and optimal way to avoid any collision and occlusion to ensure a safe collaboration.

To aim this first we detect the object to grasp. We have considered small cubes with fiducial markers as target objects which are supposed to detect by the Intel Realsense D455 camera. To be able to control the pose of the camera and guarantee to keep a line-of-sigh on the target camera is mounted on the second manipulator's end-effector. The geometrical transformation from the camera to the second arm's end-effector which is in charge of grasping the object has been computed. The visualization and control are done in the Robot Operating System (ROS) environment, a static transform node has been implemented inside Rviz. Therefore the Realsense camera detects an Apriltag marker attached to the cube and the location of the object can be calculated and projected in the worker arm reference frame. As for the operator, the Nuitracker AI provides full-body skeleton tracking based on RGBD data and we will be able to extract the Cartesian coordinates of the operator's shoulders and elbows. Then ROS node generates cylinders in the virtual workspace visualized in Rviz as obstacles. By the way, if the camera fails to detect the object, we move the camera to another pose.

When the camera is able to detect the target successfully, its pose is used to compute the IK polynomial roots. We leverage a methodology to obtain a univariate polynomial equation for the IKP, to reach possible postures (joint space) of the robot for the same end-effector pose (Cartesian space). Among all possible solutions, we select the best solution to avoid occlusion between the actuated camera and all known objects in the workspace. Virtual obstacles representing the line-of-sight between the camera and the objects are included as well as the addition of the operators' tracked arms. While several solutions to the IKP may result in an occlusion-free final posture, finally, a path planner is used to compute the trajectory between the original and final (optimal) posture, taking into account the occlusion and obstacles. If the planning fails, we change the camera point-of-view and start the previous steps again. The proposed methodology in terms of reliability, repeatability, and robustness has been validated in simulation and with experiments.

The main objective of this research is to develop a theoretical method for replicable trajectory planning of non-wrist-partitioned manipulators in collaborative pick-and-place tasks. These innovative contributions have been clearly stated and the proposed method is compared with the state-of-the-art control strategies in the field of robotics. The novel contribution in this research is a method that consistently provides the best set of joint positions and guarantees a collision-free path for the robot, while numerical iterative methods, i.e. state-of-the-art commercial/industrial solutions, provide variable results on each run. Eventually, our analytical-based method, integrating collision and occlusion avoidance optimizations, can contribute to greatly enhancing the efficiency and safety of collaborative assembly workstations.

As well our approach to finding the optimal posture of the manipulator is based on the symbolic computation of the complete set of solutions to the IKP. This is done with the derivation of a univariate polynomial equation. This strategy, as well as all the tools we derive, can be implemented for any non-wrist-partitioned manipulator. For the one that can be wrist partitioned, simpler solutions to the IK have been demonstrated already. The final solution we get is indeed tailored to the Kinova Gen3 lite manipulator.

In addition, our methodology compared with other methods is more robust. In our case, the polynomial inverse kinematic solution will always provide all solutions to the pose problem if any exists, as mentioned for a similar earlier derivation by Gosselin & Liu (2014), while other researches stated in the literature review more specifically, Shadrin *et al.* (2020) discussed the robustness of their method by changing the values of the parameters in the controller and showing that it still converges.

Finally, we are using the path planner to reach the final pose without collision/ and occlusion. Our algorithm has been tested with varying conditions (different camera positions, different initial positions for the arm) and demonstrated that it was always able to find a collision-free path when possible. In a particular case where it was impossible to reach the object without causing an occlusion with the camera fixed, our method still worked by first moving the camera to a more suitable position. We also compared our algorithm to a standard implementation of the OMPL path planner in MoveIt! by performing repeatability tests. We observed that in some cases OMPL was not able to find an occlusion-free solution, while our proposed methodology always found at least one feasible path that was collision and occlusion-free.

2.2 Hardware and Software

2.2.1 Kinova Gen3 Lite Manipulator

Kinova Gen3 Lite is a 6 Degree of Freedom (DOF) robot manipulator, the newest and most compact member in the Kinova ultra lightweight robot series which offers more options for one or multiple professional-grade robots to perform light manipulation and mobile application tasks. It weighs 5.4 kg with a maximum payload of 0.5 kg and reachability of 760 mm (Gen, 2022).

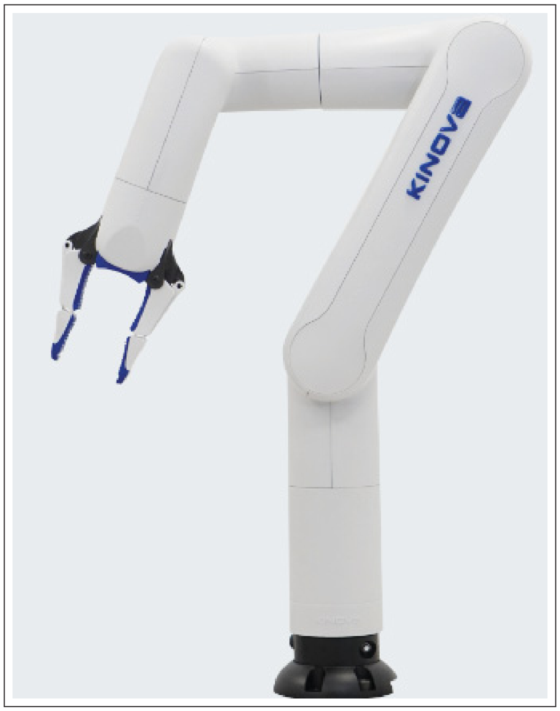


Figure 2.1 Kinove Gen3 Lite 6-DOF manipulator

Kinova robots are equipped with Kinova® Kortex™, which enables the user to configure and control the robot programmatically as well as adapt to specific needs. In addition, Kinova offers Robot Operating System (ROS) (Quigley, Faust, Foote & Leibs, 2009) packages. Therefore, we can program any kind of manipulation application ranging from simple grasping to complex manipulation using ROS, C++, and Python. The Kinova Gen3 Lite can be seen in Fig. 2.1. There are two levels of controlling the movement of the Kinova robot:

- High-level
- Low-level

The high-level control is the default on boot-up, and it is the safest and simplest to use. All commands are sent through the robot base, whether they are high-level or low-level. Control libraries for Kinova robots process these commands. The robot control library applies high-level control features like:

- Singularity avoidance
- Protection zones
- Cartesian and joint limits

Additionally, the control library breaks down the command into smaller commands that are sent incrementally to the individual robot actuators via the 1 kHz communication loop. In low-level control, the user sends a series of small, incremental commands to the actuator and gripper as part of a user-defined loop (at a rate up to 1 kHz). These commands are received by the base and routed to the actuators and gripper via a 1KHz communication loop. A high-level control offers more protection and is simpler to use, while a low-level control is lighter, faster, and has a finer range of control. (Gen, 2022).

2.2.2 Intel RealSense Depth Camera and Nuitrack

Intel® RealSense™ Stereo depth technology brings 3D to devices and machines that only see 2D today. Through stereo image sensing technologies which have been consisted of two cameras, devices are able to see, understand, interact with, and learn from their environment -

enabling intuitive and natural interaction. The Intel RealSense D400 family of depth cameras is suitable for both indoor and outdoor use under diverse lighting conditions and can be configured for multiple camera configurations without custom calibration. By extracting 3D information from scenes using the Intel RealSense Depth Camera D455, developers can enhance robotic navigation, object recognition, and other applications (Dorodnicov, 2022). The Intel RealSense D455 Depth Camera can be seen in Fig. 2.2.

Figure 2.2 Intel RealSense D455 depth camera

Nuitrack™ AI is an ultimate 3D body skeleton tracking middleware developed by 3DiVi Inc based on deep learning. In addition to skeleton tracking, Nuitrack AI can also track faces and objects. Nuitrack AI processes a depth stream as well as an RGB stream from an RGBD sensor, so based on RGB-D data processing algorithms and leveraging deep learning, tracking human skeletons and faces, and object detection are possible. Furthermore, the improved 360-degree tracking allows users to recognize difficult poses like sitting and lying down, as well as people who appear from the side (Tsibulevsky, 2022).

2.2.3 ROS, Gazebo, Rviz and Moveit!

To test our implementations, we will use industry-standard software, namely ROS (Robotic Operating System) and its integrated Gazebo and Rviz simulator for visualization in the development stage. ROS (Koubâa *et al.*, 2017) is an open-source robotics middleware suite to be used in complex robot control software systems which is able to facilitates greatly the development of robotic applications.

Gazebo is an open-source 3D robotics simulator that integrated the ODE physics engine, OpenGL rendering, and support code for sensor simulation and actuator control. Gazebo simulator can simulate Kinova Gen3 Lite robot in realistic settings.

Rviz (Moon, Bird, Borenstein & Frew, 2020) is a three-dimensional visualizer used to visualize robots, the environments they work in, and sensor data. It is a highly configurable tool, with many different types of visualizations and plugins. The main purpose of the use of ROS has been to ease a future integration of the software in a higher-level framework.

MoveIt! is software for mobile manipulation incorporating the latest advances in motion planning, manipulation, 3D perception, kinematics, control, and navigation. MoveIt! evolved from the Arm Navigation framework (Chitta, Jones, Ciocarlie & Hsiao, 2012; Chitta, 2016) in ROS and provides the core functionality for manipulation in ROS. MoveIt! works with motion planners through a plugin interface. This allows MoveIt! to communicate with and use different motion planners from multiple libraries, making MoveIt! easily extensible. OMPL (Open Motion Planning Library) is an open-source motion planning library that primarily implements randomized motion planners. MoveIt! integrates directly with OMPL and uses the motion planners from that library as its primary/default set of planners. The planners in OMPL are abstract; i.e. OMPL has no concept of a robot. Instead, MoveIt! configures OMPL and provides the back-end for OMPL to work with problems in Robotics.

2.2.4 AprilTag

AprilTags are square, black, and white tags that are easy to detect in a single grayscale camera image. They are robust and flexible visual fiducial systems popular in robotics research that are conceptually similar to QR Codes (Wang & Olson, 2016). Putting them in the environment allows correspondence to be achieved between the real world and the projections from the camera. The position of an object or a camera can be estimated in this way very easily. Due to the tag's internal identification codes, they can be used for robot localization and navigation as long as the tag's position is known. In addition, AprilTags can be used to locate and identify both static and dynamic objects (Kallwies, Forkel & Wuensche, 2020). In this research we are using Apriltag on the object to localize and get the position to grasp it. The Apriltag is illustrated in Fig. 2.3.

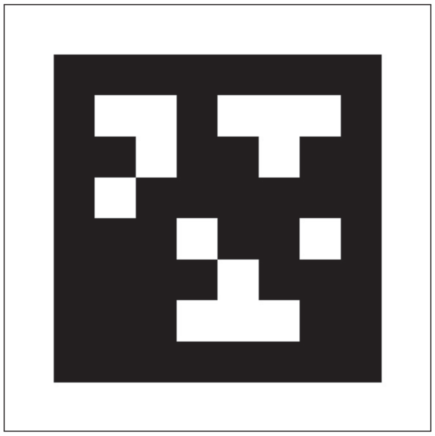
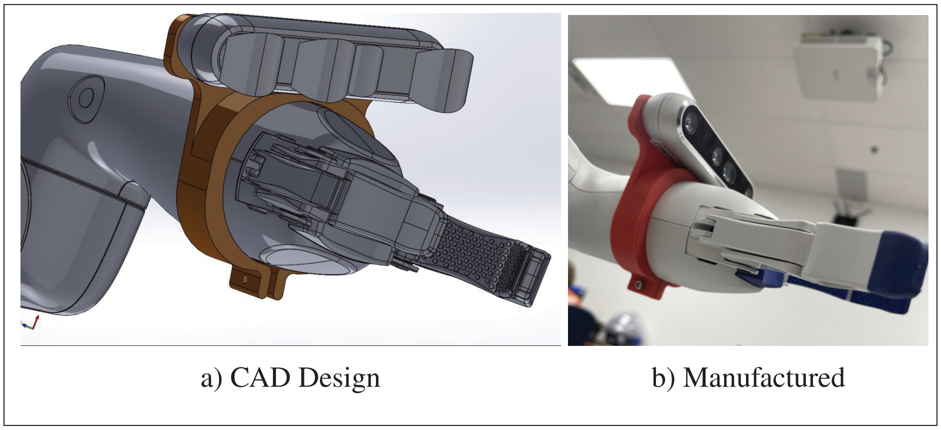


Figure 2.3 Apriltag used in this research

2.2.5 Camera Wrist Mounting and Cubes

Intel Realsense D455 camera is mounted on the second manipulator's end-effector. For this purpose mounting has been designed to fix the camera on the Gen3 Lite's wrist link. Camera wrist mounting is illustrated in Fig. 2.4.

For running the experiments, we designed small cubes with fiducial markers on them considered target objects that can be recognized and grasped by the manipulator. The cube is demonstrated in Fig. 2.5.


a) CAD Design

b) Manufactured

Figure 2.4 Camera wrist mounting

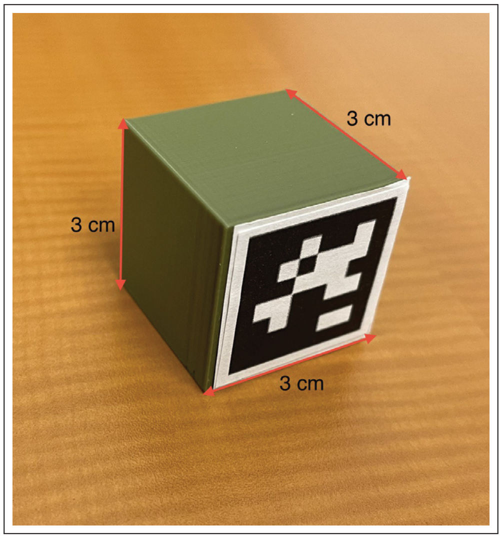


Figure 2.5 Printed cube as target objects

CHAPTER 3

6R COBOT KINEMATICS

3.1 Manipulator

As an example of a non-wrist-partitioned cobot, we tailored our derivation to the Kinova Gen3 lite, a serial manipulator with six revolute joints each having limited rotation and a two-finger gripper as the end-effector (EE). The Denavit-Hartenberg (DH) parameters of this robot are given in Table 3.1 (numerical values given in Section 3.3.1), where the non-zero parameters are identified. With the parameters in this table, it is clear that this robot is not wrist-partitioned since $b_5 \neq 0$. Thus, well-known methodologies to find the decoupled solution of the IKP cannot be used.

Table 3.1 DH parameters of the Kinova Gen3 lite

i	1	2	3	4	5	6
a_i	0	a_2	0	0	0	0
b_i	b_1	b_2	b_3	b_4	b_5	b_6
α_i	$\pi/2$	π	$\pi/2$	$\pi/2$	$\pi/2$	0

As shown in Fig. 3.1, a DH reference frame is attached to each link. It should be noted that these frames are not necessarily located at the joints. The rotation matrices Q_i and the position vectors $\mathbf{a}_i$ relating the successive reference frames defined on each of the links of the robot (Mavroidis *et al.*, 1994) can be written as

$$\mathbf{Q}_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \tag{3.1}$$

$$\mathbf{a}_i = \begin{bmatrix} a_i\cos\theta_i & a_i\sin\theta_i & b_i \end{bmatrix}^T \tag{3.2}$$

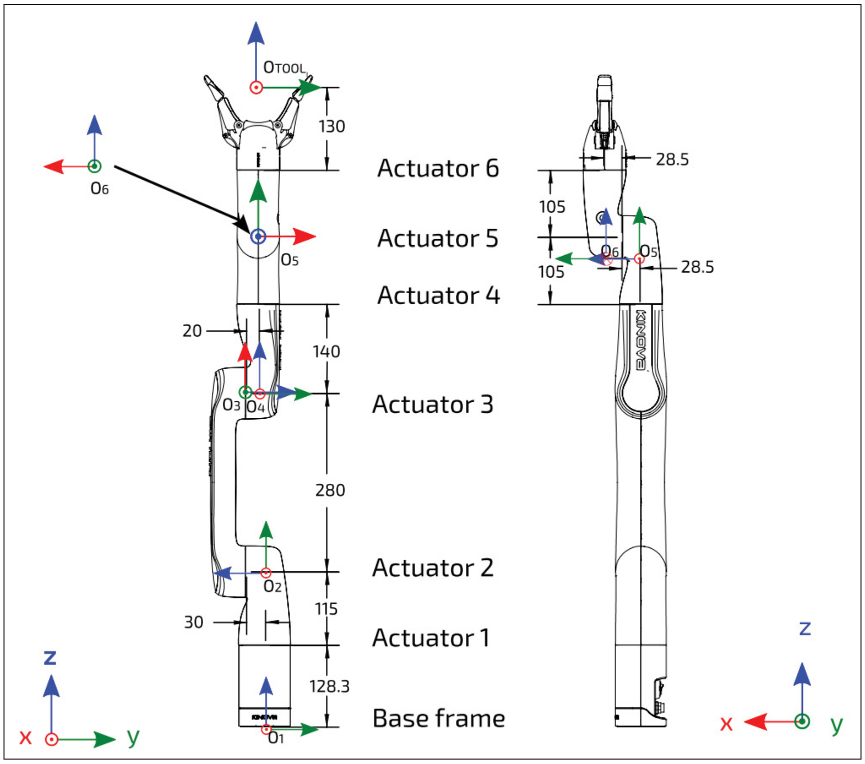


Figure 3.1 DH frames for each joint with the link dimensions (extracted from the manipulator user manual)

where the rotation matrix $\mathbf{Q}_i$ rotates frame i into the orientation of frame $(i+1)$ and vector $\mathbf{a}_i$ connects the origin of frame i to the origin of frame $(i+1)$. The joint variables are noted θ_i while a_i, b_i and α_i are the DH parameters representing the geometry of the Kinova Gen3 lite. The end-effector is located at the origin of frame 7, which is defined by the 3-dimensional vector $\mathbf{p}$. The orientation of the end-effector is given by the rotation matrix from frame 1 to frame 7, noted $\mathbf{Q}$.

3.2 IKP Analytical Solution

The forward kinematic problem (FKP), i.e. the Cartesian position $\mathbf{p}$ and orientation matrix of the tool $\mathbf{Q}$, are straight forward and can be written as

$$\mathbf{p} = \sum_{i=0}^{5}\left(\left(\prod_{j=0}^{i}\mathbf{Q}_j\right)\mathbf{a}_{i+1}\right), \quad \mathbf{Q} = \prod_{i=1}^{6}\mathbf{Q}_i \tag{3.3}$$

where $\mathbf{Q}_0$ is the (3×3) identity matrix. The first step toward solving the IKP of the Gen3 lite is to reduce the number of unknowns, currently six for the six joint positions $\{\theta_i\}$, to only one, therefore reducing the problem to a univariate polynomial equation that can be solved. By finding expressions for $\sin\theta_i$ and $\cos\theta_i$ and substituting them in $\sin^2\theta_i + \cos^2\theta_i = 1$, we can readily reduce the number of unknowns. First, we need to compute $\mathbf{r}$, connecting the origin of frame 1 to the origin of frame 6, which can be written similarly to the lefthand part of Eq. (3.3) as

$$\mathbf{r} = \sum_{i=0}^{4}\left(\left(\prod_{j=0}^{i}\mathbf{Q}_j\right)\mathbf{a}_{i+1}\right) \tag{3.4}$$

It is noted that vector $\mathbf{r}$ is independent of θ_6 , so by premultiplying Eq. (3.4) by $\mathbf{Q}_1^T$ and isolating all expressions independent of θ_1 on the righthand side, we have a set of three scalar equations. Among them, two stand out as only being function of θ_1, θ_2 and $\theta_{(3-2)}$:

$$r_1c_1 + r_2s_1 = a_2c_2 + b_5c_{(3-2)}s_4 + b_4s_{(3-2)} \tag{3.5}$$

$$r_3 - b_1 = a_2s_2 - b_5s_{(3-2)}s_4 + b_4c_{(3-2)} \tag{3.6}$$

where r_i is the i^{th} component of $\mathbf{r}$, s_i, c_i, $c_{(i-j)}$ and $s_{(i-j)}$ stand respectively for $\sin\theta_i$, $\cos\theta_i$, $\cos(\theta_i - \theta_j)$ and $\sin(\theta_i - \theta_j)$. The last scalar equation remaining after premultiplying Eq. (3.4) by $\mathbf{Q}_1^T$ and will be needed later in the derivation:

$$r_1s_1 - r_2c_1 = b_2 - b_3 + b_5c_4. \tag{3.7}$$

which can be rewritten to obtain an expression of c_4:

$$c_4 = (r_1 s_1 - r_2 c_1 + b_3 - b_2)/b_5 \tag{3.8}$$

We are now able to solve Eqs. (3.5-3.6) for s_2 and c_2. Substituting the results in $s_2^2 + c_2^2 = 1$, we obtain

$$B_1 s_{(3-2)} + B_2 c_{(3-2)} + B_3 = 0 \tag{3.9}$$

where B_1, B_2 and B_3 are functions of the DH parameters and c_1, s_1 and s_4.

$$B_1 = 2(r_3 - b_1)b_5 s_4 - 2(r_1 c_1 + r_2 s_1)b_4 \tag{3.10}$$

$$B_2 = -2b_4(r_3 - b_1) - 2b_5 s_4(r_1 c_1 + r_2 s_1) \tag{3.11}$$

$$B_3 = (r_3 - b_1)^2 + (r_1 c_1 + r_2 s_1)^2 + b_4^2 + b_5^2 s_4^2 - a_2^2 \tag{3.12}$$

Having a first equation expressed as a function of $s_{(3-2)}$ and $c_{(3-2)}$, a second one is needed to compute $s_{(3-2)}^2 + c_{(3-2)}^2 = 1$. Matrices $\mathbf{Q}$ being orthogonal matrices, the righthand part of Eq. (3.3) can be recast into

$$\mathbf{Q}_4\mathbf{Q}_5\mathbf{Q}_6 = \mathbf{Q}_3^T\mathbf{Q}_2^T\mathbf{Q}_1^T\mathbf{Q} \tag{3.13}$$

This equation gives us a system of nine scalar equations. However, only five are relevant: the ones defining the first two components of the last row and the three components of the last column of the resulting matrices. On the one hand, the former can be used to obtain expressions of c_6 and s_6:

$$c_6 = (q_{11}c_1s_{(3-2)} + q_{21}s_1s_{(3-2)} + q_{31}c_{(3-2)})/s_5 \tag{3.14}$$

$$s_6 = (q_{12}c_1s_{(3-2)} + q_{22}s_1s_{(3-2)} + q_{32}c_{(3-2)})/-s_5 \tag{3.15}$$

These two equations will be useful later in the thesis. On the other hand, the components of the last column are not a function of θ_6, because the latter corresponds to a rotation of the last joint about the z-axis of the end-effector. Therefore, the last column, defining a unit vector parallel to this axis, must be independent of θ_6. With this column, we obtain the following scalar equations which are cast in array form with dialytic elimination:

$$c_4s_5 = q_{13}c_1c_{(3-2)} + q_{23}s_1c_{(3-2)} - q_{33}s_{(3-2)} \tag{3.16}$$

$$s_4s_5 = -q_{13}s_1 + q_{23}c_1 \tag{3.17}$$

$$-c_5 = q_{13}c_1s_{(3-2)} + q_{23}s_1s_{(3-2)} + q_{33}c_{(3-2)} \tag{3.18}$$

$$\mathbf{M}\mathbf{k}_5 = \mathbf{0} \tag{3.19}$$

where **0** is a three-dimensional zero vector and

$$\mathbf{M} = \begin{bmatrix} 0 & -c_4 & m_{13} \\ 0 & -s_4 & m_{23} \\ 1 & 0 & m_{33} \end{bmatrix}, \quad \mathbf{k}_5 = \begin{bmatrix} c_5 \\ s_5 \\ 1 \end{bmatrix} \tag{3.20}$$

with, after some simplifications,

$$m_{13} = (q_{13}c_1 + q_{23}s_1)c_{(3-2)} - q_{33}s_{(3-2)} \tag{3.21}$$

$$m_{23} = (-q_{13}s_1 + q_{23}c_1) \tag{3.22}$$

$$m_{33} = (q_{13}c_1 + q_{23}s_1)s_{(3-2)} + q_{33}c_{(3-2)} \tag{3.23}$$

In the above expressions, q_{ij} is the (i, j)th component of the end-effector orientation matrix **Q**. It can be seen that **M**, an homogeneous matrix, in Eq. (3.19) is singular, as vector $\mathbf{k}_5$ cannot vanish. Therefore, we have

$$\det(\mathbf{M}) = A_1 s_{(3-2)} + A_2 c_{(3-2)} + A_3 = 0 \tag{3.24}$$

where A_1, A_2 and A_3 are function of the EE pose, θ_1 and θ_4 only.

$$A_1 = -q_{33}s_4 \tag{3.25}$$

$$A_2 = q_{13}c_1s_4 + q_{23}s_1s_4 \tag{3.26}$$

$$A_3 = q_{13}c_4s_1 - q_{23}c_1c_4 \tag{3.27}$$

Equations (3.9 & 3.24) can now be solved for $s_{(3-2)}$ and $c_{(3-2)}$, and substituted in $s^2_{(3-2)} + c^2_{(3-2)} = 1$, yielding

$$c_{(3-2)} = (A_3B_1 - B_3A_1)/(B_2A_1 - A_2B_1) \tag{3.28}$$

$$s_{(3-2)} = (A_3B_2 - B_3A_2)/(B_2A_1 - A_2B_1) \tag{3.29}$$

and, finally,

$$\begin{aligned}(A_2B_3 - A_3B_2)^2 + (A_3B_1 - A_1B_3)^2 \\ -(A_1B_2 - A_2B_1)^2 = 0\end{aligned} \tag{3.30}$$

Having eliminated all expressions of θ_2 and θ_3 with the procedure above, Eq. (3.30) is only a function of θ_1 and θ_4, bringing us closer to our objective of finding a univariate polynomial equation. Equation (3.30) can be factorized as a function of powers of c_4 and s_4, giving us

$$\begin{aligned}F_1c_4^6 + F_2c_4^5 + F_3c_4^4 + F_4c_4^3s_4 + F_5c_4^3 + F_6c_4^2s_4 \\ +F_7c_4^2 + F_8c_4s_4 + F_9c_4 + F_{10}s_4 + F_{11} = 0\end{aligned} \tag{3.31}$$

where the coefficients $F_i, \quad i = 1, \ldots, 11$ are solely dependent of θ_1. With Eq. (3.8), Eq. (3.31) becomes

$$Vs_4 + W = 0 \tag{3.32}$$

with

$$\begin{aligned}V =&v_1c_1^3 + v_2c_1^2s_1 + v_3c_1^2 + v_4c_1s_1 + v_5c_1 \\ &+ v_6s_1 + v_7\end{aligned} \tag{3.33}$$

$$\begin{aligned}W =&w_1c_1^4 + w_2c_1^3s_1 + w_3c_1^3 + w_4c_1^2s_1 + w_5c_1^2 \\ &+ w_6c_1s_1 + w_7c_1 + w_8s_1 + w_9\end{aligned} \tag{3.34}$$

where v_i and w_i are only functions of the DH parameters and the orientation **Q** and position **p** of the tool. The above equation can be solved for s_4, then substituted, with Eq. (3.8) in $s_4^2 + c_4^2 = 1$. The resulting univariate equation is

$$b_5^2 W^2 + [(r_1 s_1 - r_2 c_1 + b_3 - b_2)^2 - b_5^2] V^2 = 0 \tag{3.35}$$

Equation (3.35) is one of degree 8 in terms of c_1 and of degree 1 in terms of s_1. The following substitutions are then used

$$c_1 = 1 - T_1^2/1 + T_1^2, \quad s_1 = 2T_1/1 + T_1^2 \tag{3.36}$$

Then we use the Weierstrass substitution of Eq. (3.36) in Eq. (3.35) in order to transform the latter into a polynomial in T1. After multiplying the expression by $(1 + T_1^2)^8$ to clear the denominator, a polynomial of degree 16 in T_1 is obtained which can be written as below where $T_1 = \tan(\theta_1/2)$:

$$\sum_{i=0}^{16} E_i T_1^i = 0 \tag{3.37}$$

where $\{E_i\}$ are functions of the DH parameters and the pose of the end-effector of the manipulator at hand. The roots of this univariate polynomial can then be computed to obtain T_1, leading to the values of θ_1. Some of these solutions may be complex numbers and some can be duplicates. For control, only the real roots can be considered.

3.2.1 Backsubstitution

When the univariate polynomial is obtained, we can compute its roots numerically. If the roots are real, each root can provide us with a solution for angle θ_1. Then the value of the other joint angles associated with a particular solution must be calculated. In this section, we present a back substitution algorithm that calculates the unique joint angles corresponding to a given value of

θ_1. First we solve Eq. (3.8) and Eq. (3.32) respectively for c_4 and s_4. Therefore s_4 and c_4 can be determined for a given value of θ_1, following that angle θ_4 will be obtained uniquely using

$$\theta_4 = \arctan2(s_4, c_4) \tag{3.38}$$

When angles θ_1 and θ_4 are obtained, coefficients A_i; i = 1;2;3 and B_i; i = 1;2;3 presenting in Eq. (3.28) and Eq. (3.29) can be calculated. Eq. (3.28) and Eq. (3.29) can then be solved for $s_{(3-2)}$ and $c_{(3-2)}$ which resulting in a unique solution for angle $\theta_{(3-2)}$, namely

$$\theta_3 - \theta_2 = \arctan2(s_{(3-2)}, c_{(3-2)}) \tag{3.39}$$

Since the numerical values for c_1, s_1, $s_{(3-2)}$ and $c_{(3-2)}$ are known, then c_5 can be computed by using Eq. (3.23). as well, using the values of s_1, c_1, s_4, c_4 and c_5, s_5 can be found by leveraging Eq. (3.22). Finally by having s_5 and c_5 a unique value for θ_5 yields, namely

$$\theta_5 = \arctan2(s_5, c_5) \tag{3.40}$$

Then equations (3.5) and (3.6) are solved respectively for c_2 and s_2, which results in a unique value of θ_2, namely

$$\theta_2 = \arctan2(s_2, c_2) \tag{3.41}$$

Since $\theta_{(3-2)}$ and θ_2 are now known, angle θ_3 is easily determined.

Finally Equations Eq. (3.14) and Eq. (3.15) are solved for c_6 and s_6, which leads a unique value of θ_6, namely

$$\theta_6 = \arctan2(s_6, c_6) \tag{3.42}$$

Eventually the backsubstitution procedure completes and we will be able to determine the value of all joint angles for each real solution.

3.2.2 Special Cases

Like the majority of similar algorithms, some special cases must be considered. The special cases considered here are similar to those pointed out by Gosselin and Liu Gosselin & Liu (2014) for another manipulator and their methodology can also be applied to this manipulator.

First, it is possible that coefficient V in Eq. (3.32) becomes equal to zero. Since, according to the procedure detailed in the previous section, both s_4 and c_4 are required, the value of θ_4 cannot be computed with *atan2*. Instead, *arccos* must be used, and two values of θ_4 for a single θ_1 will be obtained. Of course, since the total number of solutions cannot exceed 16, some will be repeated. Another possible special case arises when $(B_2A_1 - A_2B_1)$ is equal to zero. Thereby, Eqs. (3.28 & 3.29) cannot be computed. Instead, Eqs. (3.9 & 3.24) are solved for $\theta_{(3-2)}$ with the Weierstrass substitution previously mentioned, leading to two solutions for $\theta_{(3-2)}$ for a single θ_1. As always, no more than 16 unique sets of joint angles can be obtained, which means there will be some repeated solutions again.

3.3 Examples and Numerical Validation

3.3.1 IKP Solutions

This section presents and discuss two examples to illustrate the IKP presented above. A Python script was written to process all equations and is publicly available online Montazer Zohour *et al.* (2021). It should be noted that while some solutions may be theoretically possible, they are not feasible in practice because of the mechanical limits of the joints. The numerical values of the DH parameters and the joints' limitations are given in Table 3.2. Finally, the roll-pitch-yaw angles are used to give the orientation of the end-effector. Incidentally, the orientation matrix **Q** is defined as

$$\mathbf{Q} \equiv \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{3.43}$$

where ϕ, θ and ψ are the roll, pitch and yaw angles, respectively, and $c_\gamma \equiv \cos\gamma$, $s_\gamma \equiv \sin\gamma$, for $\gamma = \{\phi, \psi, \theta\}$.

Table 3.2 Numerical parameters of the Kinova Gen3 lite

i	1	2	3	4	5	6
a_i	0 m	0.28 m	0 m	0 m	0 m	0 m
b_i	0.243 m	0.03 m	0.02 m	0.245 m	0.057 m	0.235 m
α_i	90°	180°	90°	90°	90°	0°
Lower limit	−154°	−150°	−150°	−149°	−145°	−149°
Upper limit	+154°	+150°	+150°	+149°	+145°	+149°

Table 3.3 The position and orientation of the end-effector used in examples

Ex. #1	x [m]	y [m]	z [m]	ϕ [rad]	θ [rad]	ψ [rad]
	0.119	-0.04	0.763	-0.527	0.47	-0.759
Ex. #2	x [m]	y [m]	z [m]	ϕ [rad]	θ [rad]	ψ [rad]
	0.503	0.122	-0.002	3.077	-0.254	0.256

3.3.2 Validation of the IKP Solution Selection Criterion

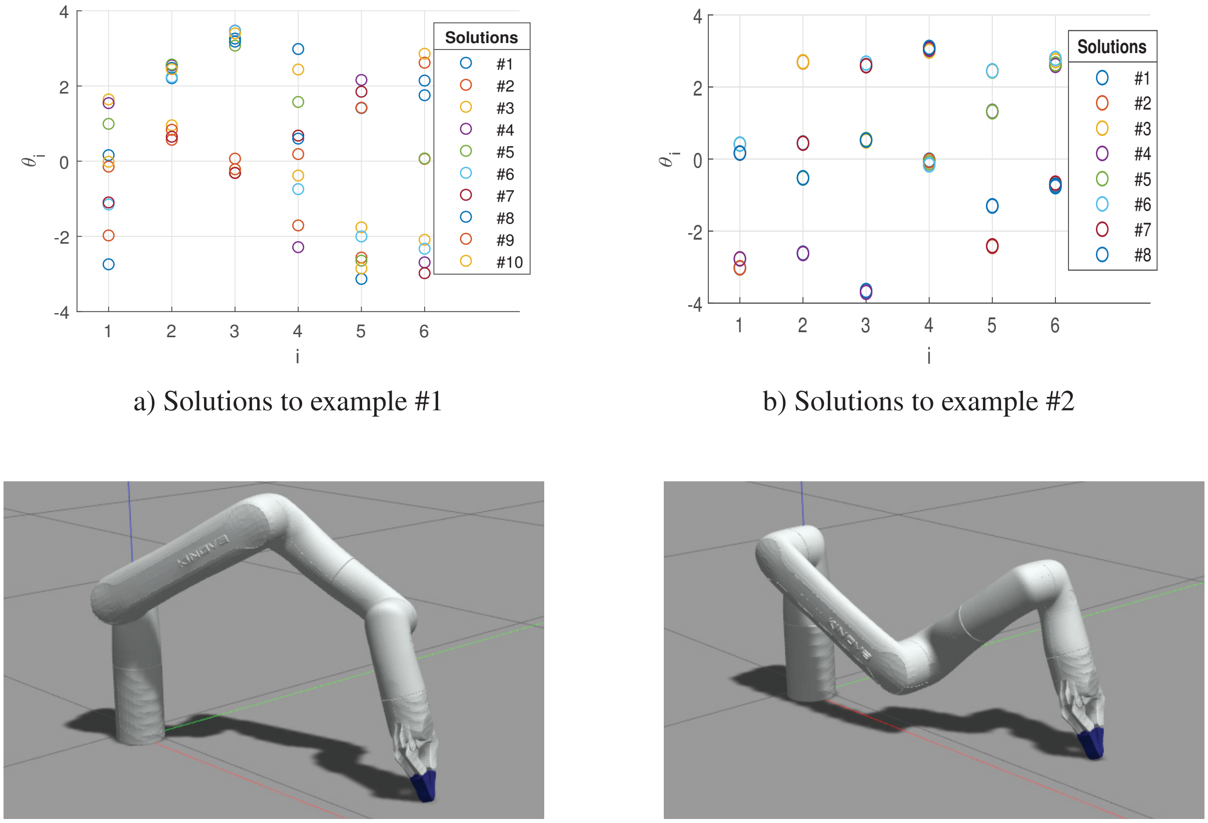
For this first example, the position and orientation of the end-effector are detailed in Table 3.3. The obtained solutions are shown in Fig. 3.2-a. It should be noted that 10 solutions were initially found by solving the IKP. However, only 6 were within the joint limitations, detailed in Table 3.4.

Table 3.4 Feasible solutions to example #1 (in radians)

Sol.	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
4	1.544	0.979	1.900	2.425	-0.982	2.021
5	0.993	1.001	1.502	0.005	0.496	-1.499
6	-1.151	0.665	1.895	-2.313	1.140	2.383
7	-1.098	-0.921	-1.885	-0.891	-1.029	1.734
8	0.160	0.910	1.609	-0.970	0.010	0.183
9	-0.145	-0.735	-1.786	-1.382	-1.718	1.049
MoveIt!	1.54	0.98	1.90	2.40	-0.98	2.00
Robot	1.59	1.00	1.93	2.39	-1.00	2.01

We also included in Table 3.4 the numerical solutions obtained with ROS MoveIt! IK package (FastIK) and the robot numerical IK embedded controller, both being among the solutions obtained with the procedure detailed in Section 3.1.

In our second example, we simulate a pick-and-place task. To be able to grasp the object, the position and orientation of the end-effector were first determined, as detailed in Table 3.3. Then our IKP script was used, leading to a set of eight solutions illustrated in Fig. 3.2-b. The four solutions respecting the joint limitations are detailed in Table 3.5, as well as the numerical solution obtained with ROS MoveIt! IK and the robot numerical IK embedded controller. An excerpt of two solutions is depicted in Fig. 3.2-c and 3.2-d from the ROS-Gazebo simulation. They will be used in the next section to illustrate the selection of the optimal posture.



a) Solutions to example #1

b) Solutions to example #2

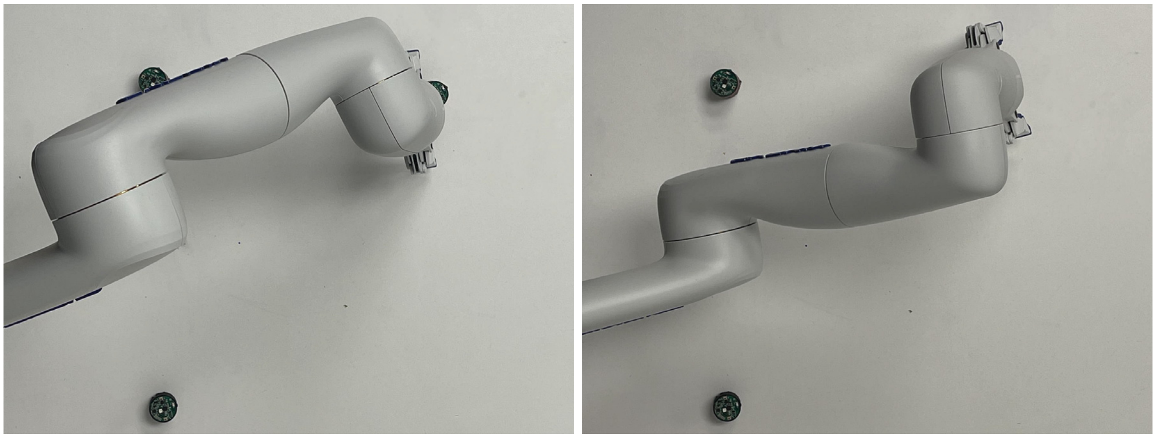
c) Ex. #2: Solution #6

d) Ex. #2: Solution #8

Figure 3.2 Possible postures from the examples

Table 3.5 Feasible solutions to example #2 (in radians)

Sol.	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
5	0.415	-2.010	-1.030	-1.678	-1.829	-1.444
6	0.414	-1.122	1.092	-1.733	-0.692	-1.292
7	0.166	-1.131	1.021	1.508	0.732	1.530
8	0.166	-2.091	-1.045	1.527	1.837	1.472
MoveIt!	0.40	-0.87	1.10	-1.55	-0.96	-1.05
Robot	0.45	-2.20	-1.19	-1.74	-1.76	-1.32



a) Solution #6

b) Solution #8

Figure 3.3 Two configurations of the arm for the same object picking task. On the left, the other object is almost completely hidden, while the right solution has a lot more margin

CHAPTER 4

OPTIMAL POSTURE AND TRAJECTORY PLANNING

4.1 Optimal Occlusion Avoidance Posture

After obtaining the solutions to the IKP for a particular end-effector pose for a pick-and-place task, the next step consists of selecting the optimal solution, as highlighted in the flowchart illustrated above (Fig. 0.1). This is done in two sub-steps, namely reducing the number of solutions to the one respecting an occlusion avoidance threshold, then choosing the one with the shortest path, as we detail in the following section.

A common setup for pick-and-place tasks is to rely on a top-view camera, positioned above the table work space. The optimization criterion is to maximize the field of view up to a certain threshold. This can be extended to several pick-and-place operations. The objective is thus to avoid the manipulator interfering with the camera's line of sight to the objects on the table. To this aim, simple line geometry is used and the shortest distance between all links and the line of sight with all objects is computed, as depicted in Fig. 4.1. In order to avoid occlusion, the latter must be kept above an arbitrary threshold d_{th}, i.e. there is no need to maximize it. We determine a unique threshold from the length of the largest link radius plus a buffer distance.

First, the position of a point along the straight line $\mathcal{P}$ from the camera, located at O_p, to a small object, located at O_z, is defined as

$$\mathbf{s}_i = \mathbf{O}_p + \Delta_{p,i}(\mathbf{O}_z - \mathbf{O}_p) \tag{4.1}$$

where $\Delta_{p,i}$ is a factor defining where along the line this point is located. Moreover, the Cartesian coordinates of points S_i, O_p and O_z are, respectively, arrayed in vectors $\mathbf{S}_i$, $\mathbf{O}_p$ and $\mathbf{O}_z$. Similarly, the position of a point P_i along the line $\mathcal{L}_i$ can be defined for any given link of the manipulator, i.e.

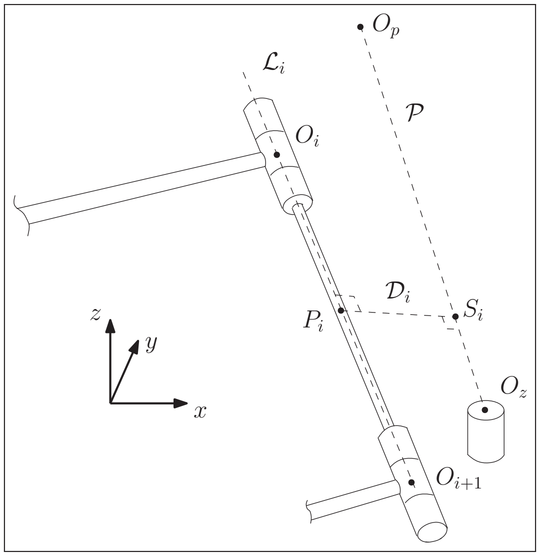


Figure 4.1 Schematic of the distance (D_i) computed between the arm link and the line of sight to the objects

$$\mathbf{p}_i = \mathbf{O}_i + \Delta_i(\mathbf{O}_{i+1} - \mathbf{O}_i), \quad i = 2, \ldots, 8 \tag{4.2}$$

where $\mathbf{O}_i$ and Δ_i are, respectively, the Cartesian coordinates of the intersections between the links and a factor defining where along the latter this point is located. If these two points are the closest pair along their respective lines, a unit vector, orthogonal to $\mathcal{L}_i$ and $\mathcal{P}$, thus parallel to $\mathcal{D}_i$, can be defined as

$$\mathbf{v}_i = \frac{(\mathbf{O}_z - \mathbf{O}_p) \times (\mathbf{O}_{i+1} - \mathbf{O}_i)}{||(\mathbf{O}_z - \mathbf{O}_p) \times (\mathbf{O}_{i+1} - \mathbf{O}_i)||} \tag{4.3}$$

Vectors $\{\mathbf{O}_i\}$ should not be confused with the locations of the DH frames, i.e. $\{\mathbf{p}_i\}$. Moreover, the first link, which is rigidly attached to the base, is not considered, since it does not translate. With these three vectors ($\mathbf{s}_i$, $\mathbf{p}_i$, $\mathbf{v}_i$), a close loop equation is formulated:

$$\mathbf{s}_i = \mathbf{p}_i + \Delta_{d,i}\mathbf{v}_i \tag{4.4}$$

where $\Delta_{d,i}$ is the shortest distance between $\mathcal{L}_i$ and $\mathcal{P}$. A set of three linear equations with three unknowns, Δ_i, $\Delta_{p,i}$ and $\Delta_{d,i}$, is thus obtained and can easily be solved.

The value of these three unknowns obtained, the risk of occlusion for an object on the table can now be computed. Indeed, the shortest distance between the robot and $\overline{O_p O_z}$, namely $\min(\Delta_{d,1}, \ldots, \Delta_{d,6})$, for a prescribed end-effector position and orientation must be larger than a certain threshold. Of course, if point P_i for a robot posture and a given link is not located within the limits of the latter, the corresponding $\Delta_{d,i}$ should be disregarded. It is the case, for instance, when O_p, O_i and O_{i+1} are aligned. Instead, the closest distance between a line ($\overline{O_p O_z}$) and a point (the corresponding link end) should be computed. This is done with the following equations:

$$\Delta_{d,i} = \frac{||(\mathbf{O}_p - \mathbf{O}_i) \times (\mathbf{O}_z - \mathbf{O}_p)||}{||\mathbf{O}_z - \mathbf{O}_p||}, \quad \text{if } \Delta_i < 0 \tag{4.5}$$

$$\Delta_{d,i} = \frac{||(\mathbf{O}_p - \mathbf{O}_{i+1}) \times (\mathbf{O}_z - \mathbf{O}_p)||}{||\mathbf{O}_z - \mathbf{O}_p||}, \quad \text{if } \Delta_i > 1 \tag{4.6}$$

The procedure above is valid if the object is relatively small, i.e. with external dimensions smaller than the threshold chosen. If it is not the case, the proposed technique can still be adapted. Indeed, the line of sight between each object in the workspace and the top-view camera is instead modeled as an irregular pyramid. Therefore, instead of having only one line $\overline{O_z O_p}$ for each object, the periphery of the latter, as seen by the camera, is discretized (with a step size smaller than $2d_{th}$), as depicted in Fig. 4.2. Therefore, the distance between each link of the robot and each line defining each pyramid must be computed for each feasible final posture. In this way, the equations detailed above can be used without any modification.

4.1.1 Validation of the Optimal Occlusion free Posture

With the postures presented in Table 3.5, solution #8, depicted in Fig. 3.2-b, is one of the potential final postures identified by the algorithm as respecting the criterion, i.e. the shortest

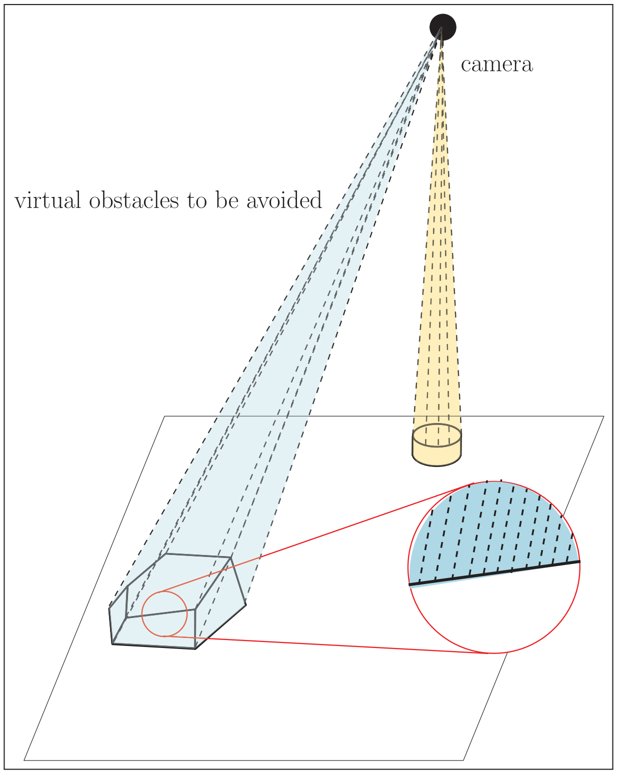


Figure 4.2 Pyramids representing the line of sight between the camera and two objects (only lines starting at the object's vertices shown on the left-hand side for clarity)

distance is above the threshold defined with two small objects that must remain visible to the camera above the workspace[1]. In this case, considering a threshold of $d_{th} = 6$ cm, which is larger than the objects' diameter of 3 cm, only one line for each is considered for the line of sight. The smallest distance between the robot and any of the two is, in this example, 0.0972 m. Moreover, this test was validated experimentally, as shown in Fig. 3.3. The photos are taken from the camera located at O_p, showing clearly that solution #8 is significantly better than solution #6 with respect to the occlusion risk for objects located at $O_{z,1}$ (top) and $O_{z,2}$ (bottom). Solution #7 is the only other feasible posture with a shortest distance above the threshold of $d_{th} = 0.06$ m.

[1] with $O_p = [0.32 \quad 0 \quad 1.3]^T$ m, $O_{z,1} = [0.25 \quad 0.15 \quad -0.002]^T$ m and $O_{z,2} = [0.25 \quad -0.15 \quad -0.002]^T$ m

4.2 Shortest Occlusion-free Path

After the initial stage, described in Section 4.1, where the set of feasible solutions to the IKP is reduced to only the postures respecting the occlusion avoidance criterion, the next step is to plan the trajectory, and, most importantly, to select the shortest path. This can be done with any trajectory planner, such as the ones available in the OMPL, integrated in ROS MoveIt!. Similarly to the section above, the line of sight between each object in the workspace and the variable-pose camera is modeled as an irregular pyramid as shown in Fig. 4.3. The apex of the latter is found at the location of the camera. These pyramids are included in the environment as virtual obstacles to be taken into account by the trajectory planner, which is fed with the postures where the shortest distance with the virtual obstacles is above the defined occlusion threshold to select the one with the shortest path.

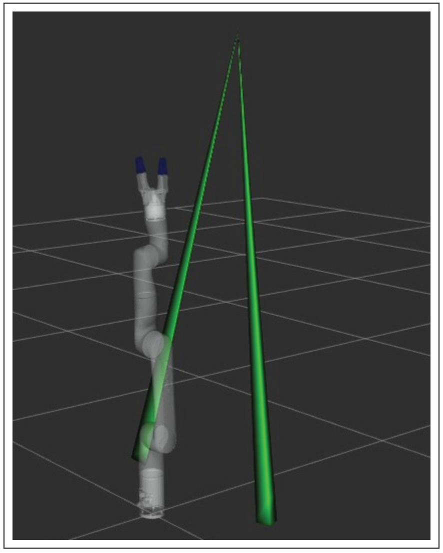


Figure 4.3 Virtual obstacles in Rviz representing the line of sight between two objects and the camera of example #2 section 3.3.1 (the third cone between the camera and the object to be picked is not shown)

4.2.1 Validation of the Path Planner

As a first validation step, we recall the example #2 from section 3.3.1, for which the optimal solution was already detailed among the set of possible solutions. The line-of-sight obstacle cones were modeled in MoveIt! OMPL. For each feasible (and occlusion-free) final posture, the trajectory planner is run to find the shortest path avoiding these virtual obstacles. Finally, among the solutions found, the shortest path is the optimal occlusion-free solution. Here, among the remaining feasible occlusion-free solutions, #8 has the shortest path with 2.22 m (2.53 m for solution #7) and, therefore, is our optimal solution.

CHAPTER 5

SAFE COLLABORATIVE ASSEMBLY STATION WITH COBOTS

5.1 Dual-arm Configuration

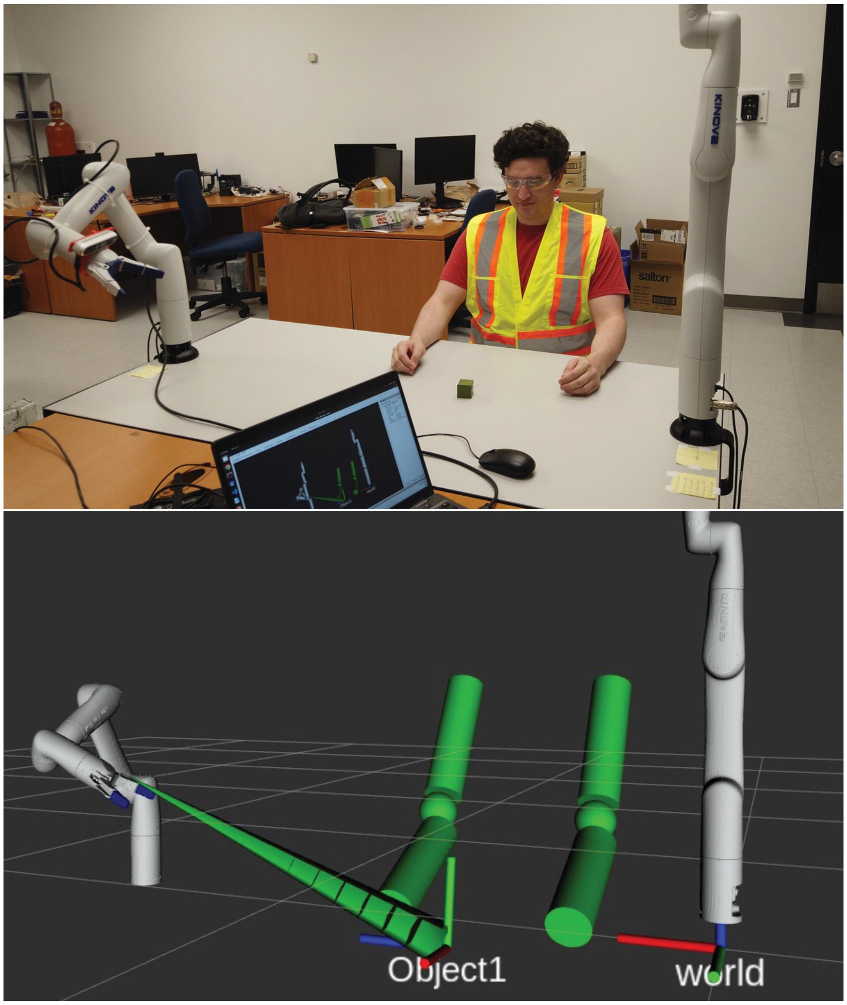


Figure 5.1 View of the experimental setup: at the top, a photo of the manipulators, camera, operator’s arms and the target object, at the bottom, the visualization (in Rviz) of the same scene, with the virtual obstacles in green

The assembly workstation we designed consists of an operator, and robotic arm helping the operator with picking specific parts. To support their collaboration we added a second robotic arm controlling the camera point-of-view. For staging the experiments, we designed small cubes with fiducial markers as the parts (target objects). An overview of the experimental setup, discussed in section 5.3 is shown in Fig. 5.1.

5.2 Target and Operator Tracking

The top-view camera configuration mentioned in the previous section is a common choice for assembly tasks without an operator. However, they are prone to occlusion by the operator head and torso whenever he/she bend over the table. Ultimately, no fixed camera position can provide a guarantee of keeping a line-of-sigh on the target. To easily control the pose of the camera in 3D space, we rigidly attached it to another identical robot. We thus have two Kinova Gen3 lite manipulators. As can be seen in Fig. 5.1 (top), an Intel Realsense D455 camera is mounted on the second manipulator's end-effector, visible on the left-hand side of the figure. Any other robotic arms equipped with a wrist camera can be used for this purpose (for instance the Gen3 and the RobotiQ wrist sensors). We then compute the geometrical transformation from the camera to the right end-effector frame (worker robot) using its forward kinematics (described in Section 3.1). Frames are illustrated in Fig. 5.2.

The visualization and control are done in the Robot Operating System (ROS) environment, a static transform node is used inside Rviz. The left manipulator is in charge of grasping the object, which is located in the workspace visible in the same figure. The pose of the object (cube) to be picked is obtained using an Apriltag marker attached to it. The Realsense camera detects the tag in the camera frame and we then project it in the worker arm reference frame (right-hand side of Fig. 5.2). Transformation tree extracted from ROS has been demonstrated in Fig. 5.3. As mentioned in the previous section, we add a *vision* cone from the tag to the camera in the virtual workspace as an obstacle to avoid occlusion.

As for detecting the operator, we leverage the Nuitracker software (Tsibulevsky, 2022) fed with the same camera (D455, RGB and depth). Nuitracker's AI Skeleton Tracking feature provides full body skeleton tracking based on RGBD data, as illustrated in Fig. 5.4. Using Nuitracker's SDK we extract the Cartesian coordinates of the operator's shoulders and elbows and broadcast

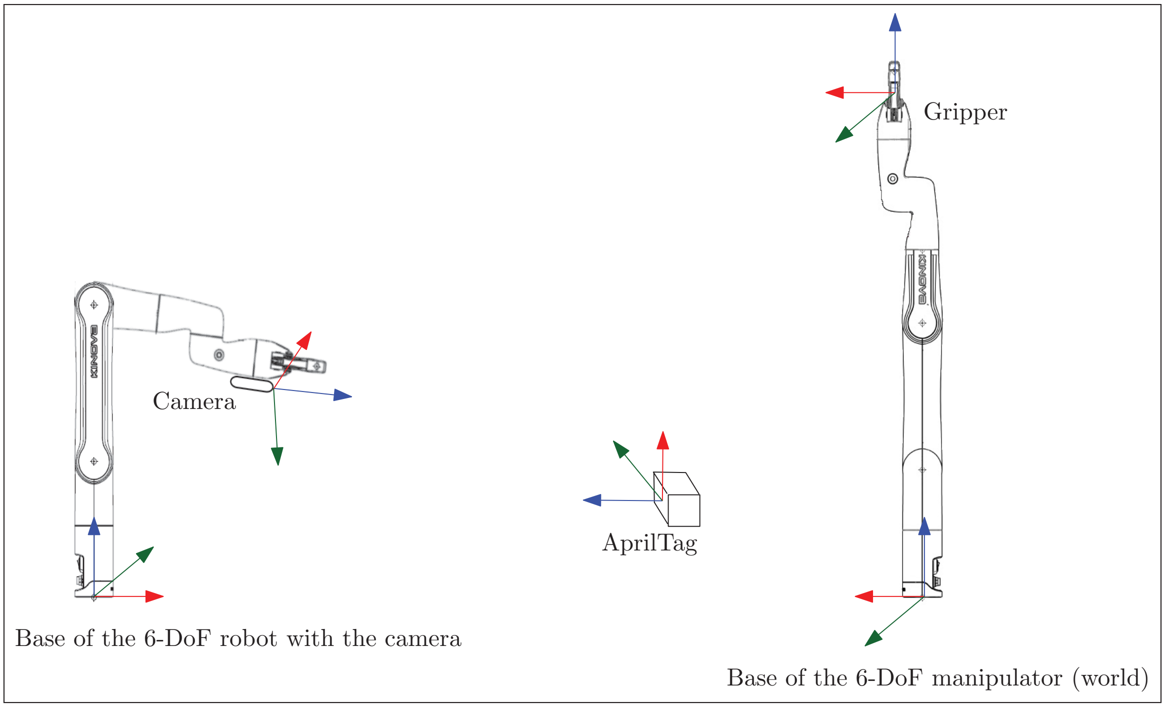

Gripper
Camera
AprilTag
Base of the 6-DoF robot with the camera
Base of the 6-DoF manipulator (world)

Figure 5.2 Frames in the experimental setup (x-axis in red, y-axis in green, z-axis in blue)

them as ROS topics. Since the arms are considered as obstacles, another node catches the topic and generates cylinders in the virtual workspace that can be visualized in Rviz, as shown in Fig. 5.1 (bottom).

The camera stays still most of the time unless it cannot detect the target or the path planner fails. In this case, we change the camera pose which leads to generating another occlusion cone to avoid and thus a different path planning. Currently, several manually tuned camera poses were recorded and selected randomly.

5.3 Experiments

In order to test and compare the performance of our proposal, six other test cases, illustrated on Fig. 5.5 were executed. For each test case, the two cubes were positioned in different locations within the workspace and the operator position changed slightly. A sample of the test configurations is illustrated in Fig. 5.6. As it was done with the previous scenario, virtual

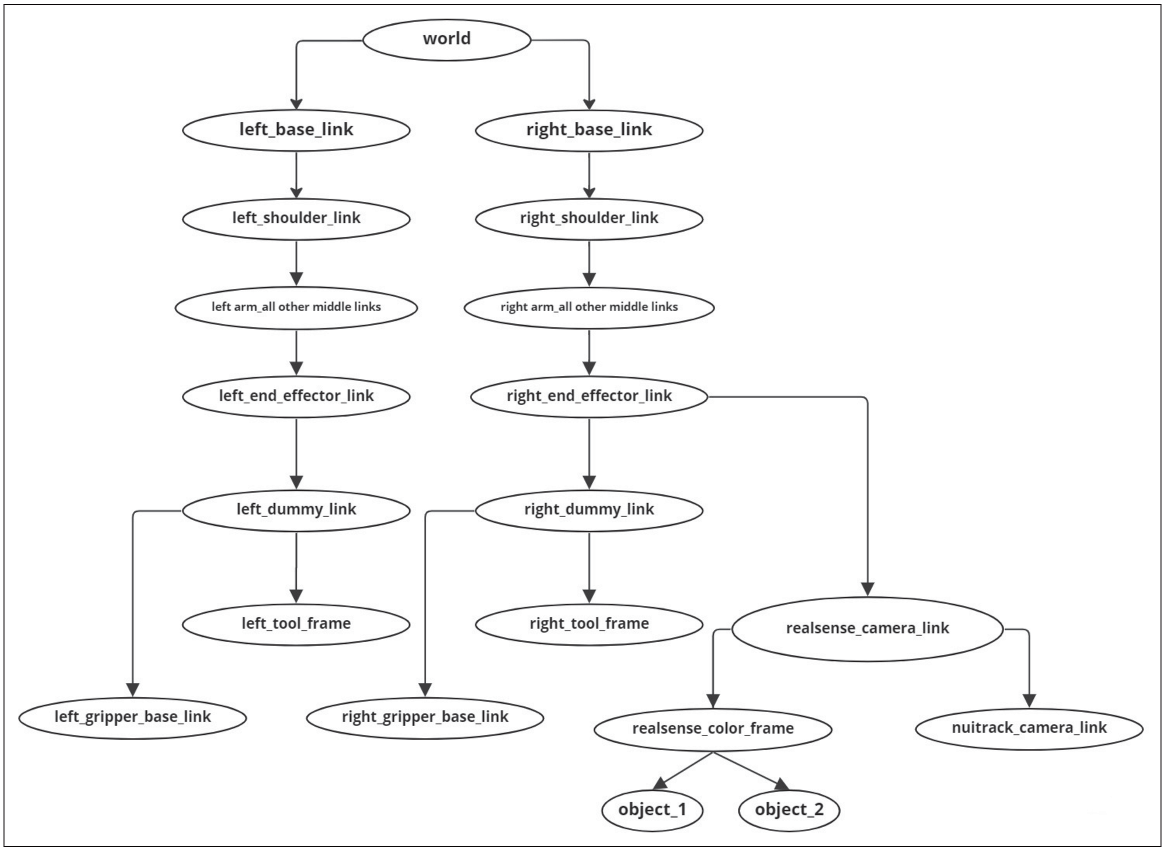


Figure 5.3   Transformation tree extracted from ROS

obstacles representing the line-of-sight between the camera and the objects are included, with the addition of the operators' tracked arms. Again, the shortest path between the solutions above the occlusion threshold is chosen.

While several solutions to the IKP may result in an occlusion-free final posture (respecting the threshold), the path planner must adapt the trajectory. The bottom configuration sample shown in Fig. 5.6 required the manipulator to first rotate it's first joint (revolute joint about a vertical axis) in order to avoid occluding the second object. To show that our proposed methodology is able to find collision and occlusion-free paths more reliably than a standard solution, we compared it to a *bare-bones* implementation inside MoveIt!, which uses a numerical solver for the IKP (only one solution found) and no obstacle detection (virtual–occlusion pyramids and real–operator's arms).

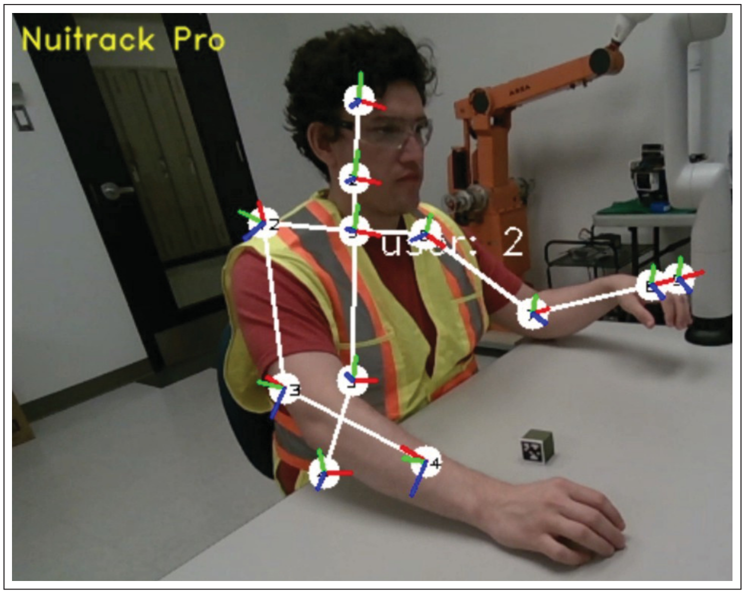
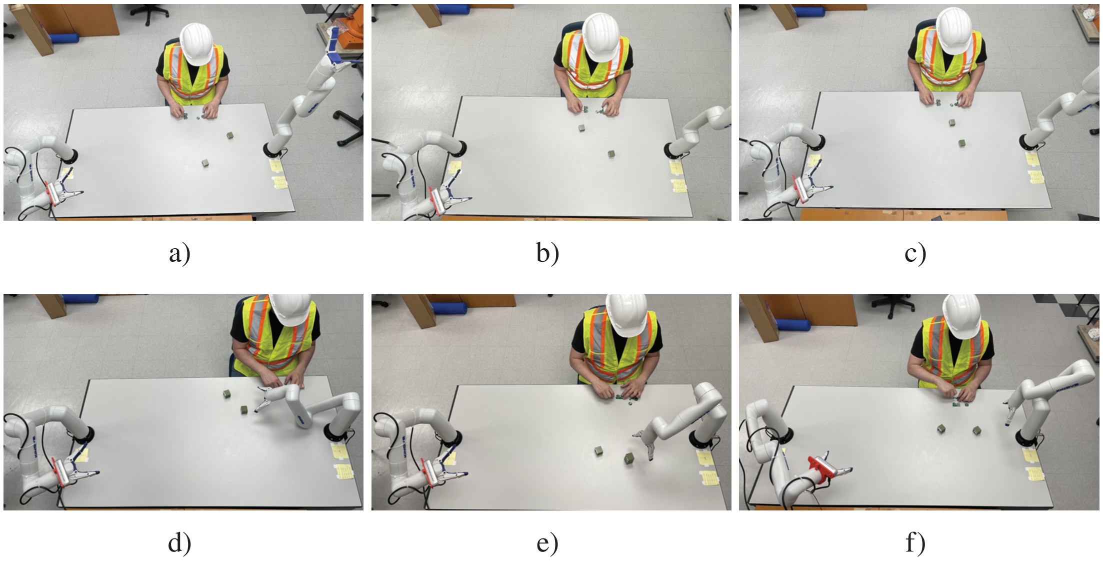


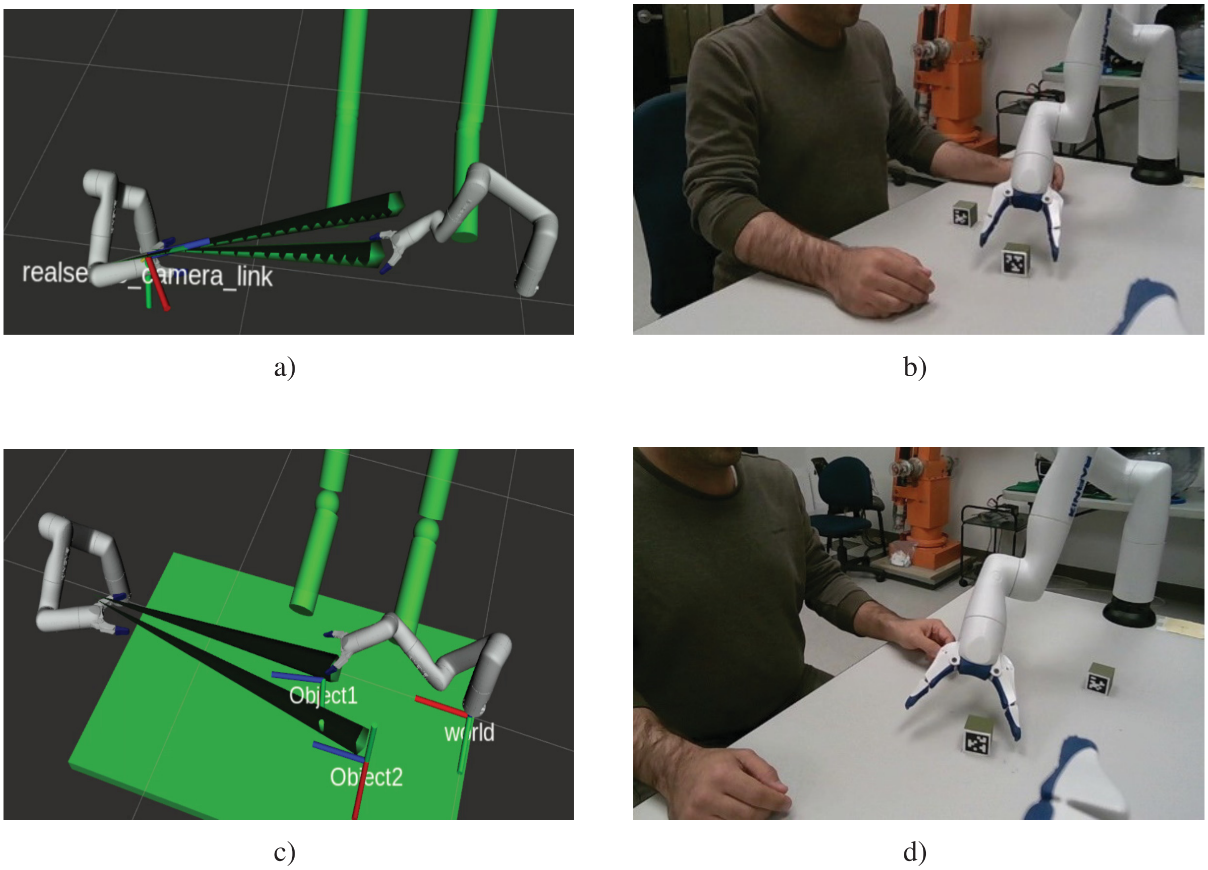
Figure 5.4 View from the Nuitracker application showing the operator's skeleton as detected by the software. Our solution extracts the joints and transfer their location in the manipulator reference frame to generate obstacles for the path planer


a) b) c)

d) e) f)

Figure 5.5 View of the positions of objects, operator, camera and initial posture of the grasping manipulator for each of the 6 experimental scenarios. (a) to (f) show scenarios 1 to 6, respectively

We repeated 10 times each scenario illustrated in Fig. 5.5 and compared the performance of the two methodologies in terms of the number of attempts that resulted in at least a partial



a)

b)

c)

d)

Figure 5.6 Two examples of the test cases: (a) and (c) are the Rviz visualizations of the virtual environment, while (b)-(d) are photos of the setup just before grasping the cube

occlusion during the manipulator's motion. Results are detailed in Table 5.1, including the success rate of each algorithm. Since our proposal considers all feasible postures for a given object's grasping pose, we also report the number of feasible paths found, and the number which were collision-free.

Table 5.1 Results of pick-and-place experiments (each scenario repeated 10 times)

Scenario		a	b	c	d	e	f
Our proposal	Feasible solutions	8	8	8	8	8	8
	Occlusion free solutions	8	8	8	8	8	6
	Attempts with occlusion	0	0	0	0	2	0
	Success rate (%)	100	100	100	100	80	100
Bare-bones MoveIt!	Attempts with occlusion	0	0	0	2	10	6
	Success rate (%)	100	100	100	80	0	40

In scenarios 1 to 3, the grasping manipulator started in a vertical posture, which led MoveIt! numerical solver to compute an optimal path in which the arm kept an *elbow-up* configuration, causing no collision or occlusion. Our methodology found a similar path to the object, again without occlusion nor collision. For scenarios 4 to 6, we changed the initial posture of the manipulator to different folded shapes, since a cobot is unlikely to go back to an upright joint configuration after each grasp in a practical application. These postures can be seen in Fig. 5.7. With these cases, the *bare-bones* MoveIt! version (with the numerical solution to the IKP, without virtual obstacles) was not able to reliably find an occlusion- and collision-free solution on all occasions. In fact, in scenario 5, all runs attempted with the *bare-bones* MoveIt! version failed to avoid any collision or occlusion. Meanwhile, our proposed methodology was always able to find at least one feasible path that was a collision- and occlusion-free, succeeding in completing the task 8 times over 10 attempts. It should be noted that the manipulator slightly occluded one of the objects during to trials.

Finally, scenario 6 is a particular case, as we positioned the objects in such a way that it was impossible to reach the object without causing an occlusion with the camera fixed at the previous position. In this case, according to block 6 of our flow chart (Fig. 0.1), the camera pose needs to be changed to successfully complete the task. Therefore, the camera is moved to a more favorable position, allowing our system to find occlusion-free paths. The resulting setup is shown

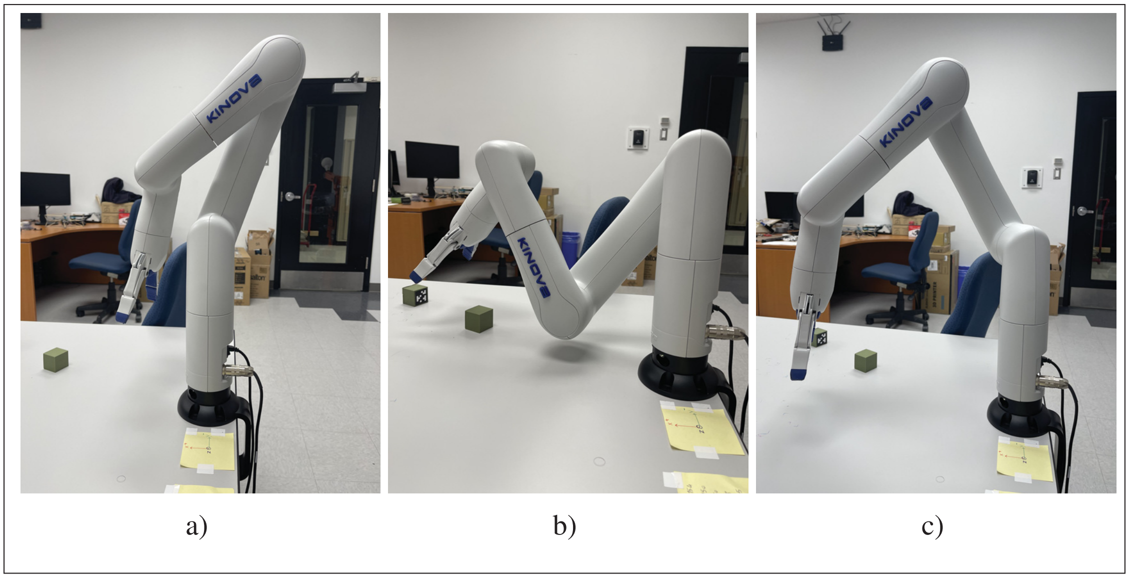


Figure 5.7 Initial positions of the grasping arm: (a) Folded; (b) Elbow-down; (c) Elbow-up

in Fig. 5.8. This demonstrates that our system can adapt to different scenarios by changing the pose of the camera.

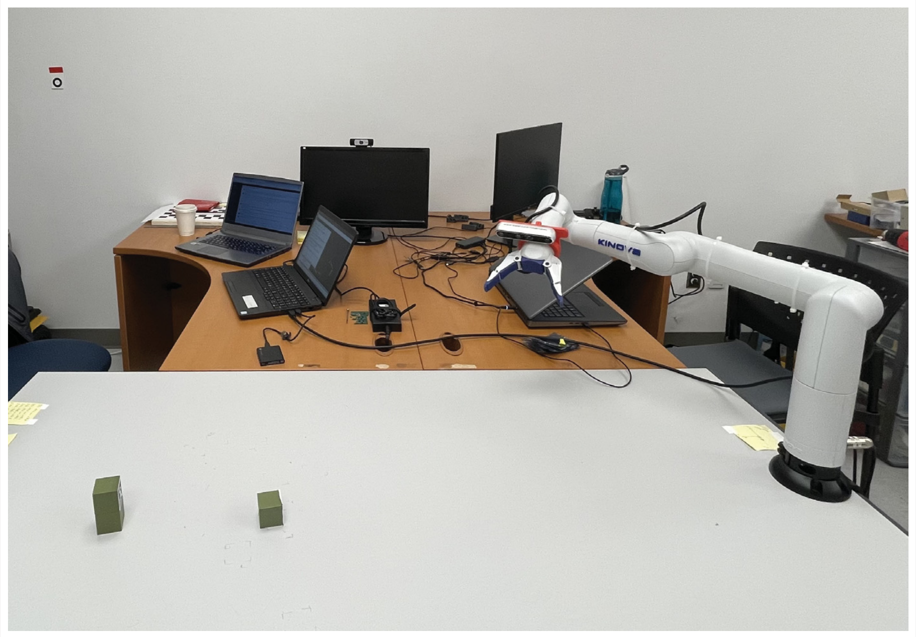


Figure 5.8 Alternative camera pose used for the 6th scenario, resulting in a successful occlusion-free path-planning

CONCLUSION AND RECOMMENDATIONS

In this thesis first, the inverse kinematic problem of a non-wrist-decoupled robot was studied using the example of the Kinova Gen3 lite manipulator. We solved it by deriving a univariate polynomial equation to find all possible values of one angle, θ_1, then finding the corresponding values of the other joint angular positions by back substitution. The Python script used to compute the solutions to the IKP is now public. Several examples were given and compared to the solutions obtained with ROS MoveIt! IK and the real robot controller for validation. In addition, a procedure to select the optimal solution in order to minimize the risk of occlusion while performing a collaborative pick-and-place task with the shortest path was proposed. The solution includes the use of a variable-pose camera to track objects within the workspace as well as the operator. Eventually, experiments to validate the procedure were included and discussed, clearly showing the usefulness of our proposal. We assessed the robustness of our algorithm based on repeatability tests in different scenarios, with varying conditions (different camera positions, different initial positions for the arm), and demonstrated that our solution was always able to find a collision-free path when possible. In a particular case where it was impossible to reach the object without causing an occlusion with the camera fixed, our method still worked by first moving the camera to a more suitable position. Compared to a standard implementation of the OMPL path planner in MoveIt!, our proposed methodology always found at least one feasible path that was collision and occlusion-free, while OMPL failed to do the same in some cases. Future work will include devising a methodology to find an optimal pose for the camera to minimize occlusion. In addition, the operator's gaze could be traced to be avoided its occlusion, similar to the methodology that we implemented on the camera's line of sight. Also, when the plan planner fails to find a solution, checking other feasible IKP solutions before moving the camera will be another potential methodology that can be added to this work. The summary of this thesis already has been published at the MDPI Sensor.

Montazer Zohour, Hamed, Belzile, Bruno, Gomes Braga, Rafael et St-Onge, David. 2022. « Minimize tracking occlusion in collaborative pick-and-place tasks: An analytical approach for non-wrist-partitioned manipulators ». Sensors, vol. 22, nº 17

BIBLIOGRAPHY

(2022). Kinova Gen3 lite. Retrieved on 2021-04-21 from: https://www.kinovarobotics.com/product/gen3-lite-robots.

(2022). Kuka AG. Retrieved on 2017-03-01 from: http://www.kuka.com.

(2022). Universal Robot. Retrieved on 2017-03-01 from: http://www.universal-robots.com.

Aghajarian, M. & Kiani, K. (2011). Inverse Kinematics solution of PUMA 560 robot arm using ANFIS. *URAI 2011 - 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 574–578. doi: 10.1109/URAI.2011.6145885.

Ahmed Zaki, A. M., Mohamed Fathy, A. M., Carnevale, M. & Giberti, H. (2022). Application of Realtime Robotics platform to execute unstructured industrial tasks involving industrial robots, cobots, and human operators.

Angeles, J. & Zanganeh, K. E. (1993). The semigraphical determination of all real inverse kinematic solutions of general six-revolute manipulators. In *Lecture Notes in Control and Information Sciences* (vol. 187, pp. 23–32). Springer Verlag. doi: 10.1007/bfb0031428.

Asfour, T. & Dillmann, R. (2003). Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, 2, 1407–1412.

Babin, V., St-Onge, D. & Gosselin, C. (2019). Stable and repeatable grasping of flat objects on hard surfaces using passive and epicyclic mechanisms. *Robotics and Computer-Integrated Manufacturing*, 55(July 2018), 1–10. doi: 10.1016/j.rcim.2018.06.002.

Bloss, R. (2016). Collaborative robots are rapidly providing major improvements in productivity, safety, programing ease, portability and cost while addressing many new applications. *Industrial Robot: An International Journal*, 43(5), 463–468.

Bortolini, M., Ferrari, E., Gamberi, M., Pilati, F. & Faccio, M. (2017). Assembly system design in the Industry 4.0 era: a general framework. IFAC-PapersOnLine 50 (1): 5700–5705.

Chen, I. M., Yang, G. & Kang, I. G. (1999). Numerical inverse kinematics for modular reconfigurable robots. *Journal of Robotic Systems*, 16(4), 213–225. doi: 10.1002/(SICI)1097-4563(199904)16:4<213::AID-ROB2>3.0.CO;2-Z.

Chen, J.-H. & Song, K.-T. (2018). Collision-free motion planning for human-robot collaborative safety under cartesian constraint. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4348–4354.

Cherubini, A., Passama, R., Crosnier, A., Lasnier, A. & Fraisse, P. (2016). Collaborative manufacturing with physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, 40, 1–13.

Chitta, S. (2016). MoveIt!: an introduction. In *Robot Operating System (ROS)* (pp. 3–27). Springer.

Chitta, S., Jones, E. G., Ciocarlie, M. & Hsiao, K. (2012). Perception, planning, and execution for mobile manipulation in unstructured environments. *IEEE Robotics and Automation Magazine, Special Issue on Mobile Manipulation*, 19(2), 58–71.

Dorodnicov, S. (2022). Intel® RealSense™ SDK. Retrieved on 2021-03-01 from: https://www.intelrealsense.com/.

Duleba, I. & Opalka, M. (2013). A comparison of jacobian-based methods of inverse kinematics for serial robot manipulators. *International Journal of Applied Mathematics and Computer Science*, 23(2), 373–382. doi: 10.2478/amcs-2013-0028.

Dwivedi, A., Gorjup, G., Kwon, Y. & Liarokapis, M. (2019). Combining Electromyography and Fiducial Marker Based Tracking for Intuitive Telemanipulation with a Robot Arm Hand System. *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1-6. doi: 10.1109/RO-MAN46459.2019.8956456.

Fan, Q., Brown, L. & Smith, J. (2016). A closer look at Faster R-CNN for vehicle detection. *2016 IEEE intelligent vehicles symposium (IV)*, pp. 124–129.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9), 1627–1645.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J. & Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6), 2280–2292.

Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Gopinath, V., Ore, F., Grahn, S. & Johansen, K. (2018). Safety-focussed design of collaborative assembly station with large industrial robots. *Procedia manufacturing*, 25, 503–510.

Gosselin, C. & Liu, H. (2014). Polynomial Inverse Kinematic Solution of the Jaco Robot. *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V05BT08A055-. doi: 10.1115/detc2014-34152.

Guo, Y., Dong, H. & Ke, Y. (2015). Stiffness-oriented posture optimization in robotic machining applications. *Robotics and Computer-Integrated Manufacturing*, 35, 69–76.

Hanna, A., Götvall, P.-L., Ekström, M. & Bengtsson, K. (2018). Requirements for designing and controlling autonomous collaborative robots system-an industrial case. *Advances in Transdisciplinary Engineering*, 139–144.

Hanna, A., Bengtsson, K., Dahl, M., Erős, E., Götvall, P.-L. & Ekström, M. (2019). Industrial challenges when planning and preparing collaborative and intelligent automation systems for final assembly stations. *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 400–406.

Hanna, A., Bengtsson, K., Götvall, P.-L. & Ekström, M. (2020). Towards safe human robot collaboration-Risk assessment of intelligent automation. *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1, 424–431.

Ho, T., Kang, C.-G. & Lee, S. (2012). Efficient closed-form solution of inverse kinematics for a specific six-DOF arm. *International Journal of Control, Automation and Systems*, 10(3), 567–573.

Hoi, S. C., Wu, X., Liu, H., Wu, Y., Wang, H., Xue, H. & Wu, Q. (2015). Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks. *arXiv preprint arXiv:1511.02462*.

Husty, M. L., Pfurner, M. & Schröcker, H. P. (2007). A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator. *Mechanism and Machine Theory*, 42(1), 66–81. doi: 10.1016/j.mechmachtheory.2006.02.001.

Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P. & Schaal, S. (2011). STOMP: Stochastic trajectory optimization for motion planning. *2011 IEEE international conference on robotics and automation*, pp. 4569–4574.

Kallwies, J., Forkel, B. & Wuensche, H.-J. (2020). Determining and improving the localization accuracy of AprilTag detection. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8288–8294.

Kang, G., Kim, Y. B., Lee, Y. H., Oh, H. S., You, W. S. & Choi, H. R. (2019). Sampling-based motion planning of manipulator with goal-oriented sampling. *Intelligent Service Robotics*, 12(3), 265–273.

Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., Zhang, C., Wang, Z., Wang, R., Wang, X. et al. (2017). T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10), 2896–2907.

Kavraki, L. E., Svestka, P., Latombe, J.-C. & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4), 566–580.

Koubâa, A. et al. (2017). *Robot Operating System (ROS).* Springer.

Kuffner, J. J. & LaValle, S. M. (2000). RRT-connect: An efficient approach to single-query path planning. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, 2, 995–1001.

Lee, H. Y., Woernle, C. & Hiller, M. (1991). A complete solution for the inverse kinematic problem of the general 6r robot manipulator. *Journal of Mechanical Design, Transactions of the ASME*, 113(4), 481–486. doi: 10.1115/1.2912808.

Lee, H.-Y. & Liang, C.-G. (1988). Displacement analysis of the general spatial 7-link 7R mechanism. *Mechanism and machine theory*, 23(3), 219–226.

Li, T., Huang, B., Li, C. & Huang, M. (2019). Application of convolution neural network object detection algorithm in logistics warehouse. *The Journal of Engineering*, 2019(23), 9053–9058.

Lin, Y., Zhao, H. & Ding, H. (2017). Posture optimization methodology of 6R industrial robots for machining using performance evaluation indexes. *Robotics and Computer-Integrated Manufacturing*, 48, 59–72. doi: 10.1016/j.rcim.2017.02.002.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B. & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42, 60–88.

Lozano-Perez, T., Jones, J. L., O'Donnell, P. A. & Mazer, E. (1992). *Handey: a robot task planner*. Mit Press.

Manseur, R. & Doty, K. L. (1989). A Robot Manipulator With 16 Real Inverse Kinematic Solution Sets. *International Journal of Robotics Research*, 8(5), 75–79. doi: 10.1177/027836498900800507.

Marvel, J. A., Falco, J. & Marstio, I. (2014). Characterizing task-based human–robot collaboration safety in manufacturing. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(2), 260–275.

Mavroidis, C., Ouezdou, F. & Bidaud, P. (1994). Inverse kinematics of a six-degree of freedom 'General'and 'Special 'manipulators using symbolic computation. *Robotica*, 12, 421–430.

Montazer Zohour, H., Belzile, B. & St-Onge, D. (2021). *Kinova Gen3 Lite inverse kinematic optimal solution selection*. Lab INIT Robots. Retrieved from: https://github.com/INITRobots/OcclusionsfreeIKP.

Montazer Zohour, H., Belzile, B., Gomes Braga, R. & St-Onge, D. (2022). Minimize tracking occlusion in collaborative pick-and-place tasks: An analytical approach for non-wrist-partitioned manipulators. *Sensors*, 22(17), 6430.

Moon, S., Bird, J. J., Borenstein, S. & Frew, E. W. (2020). A Gazebo/ROS-based Communication-Realistic Simulator for Networked sUAS. *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1819–1827.

of Robotics, I. F. (2017). *World Robotics 2017: Industrial Robots*. VDMA.

Ogorodnikova, O. (2009). How safe the human-robot coexistence is? Theoretical presentation. *Acta Polytechnica Hungarica*, 6(4), 51–74.

Park, C., Pan, J. & Manocha, D. (2013). Real-time optimization-based planning in dynamic environments using GPUs. *2013 IEEE International Conference on Robotics and Automation*, pp. 4090–4097.

Pieper, D. L. (1968). *The Kinematics of Manipulators under Computer Control*. (Ph.D. thesis, Stanford University).

Primrose, E. J. (1986). On the input-output equation of the general 7R-mechanism. *Mechanism and Machine Theory*, 21(6), 509–510. doi: 10.1016/0094-114X(86)90134-5.

Qiao, S., Liao, Q., Wei, S. & Su, H. J. (2010). Inverse kinematic analysis of the general 6R serial manipulators based on double quaternions. *Mechanism and Machine Theory*, 45(2), 193–199. doi: 10.1016/j.mechmachtheory.2009.05.013.

Quigley, M., Faust, J., Foote, T. & Leibs, J. (2009). *ROS: an open-source Robot Operating System*. Now Publishers Inc.

Raghavan, M. & Roth, B. (1993). Inverse kinematics of the general 6R manipulator and related linkages.

Ratliff, N., Zucker, M., Bagnell, J. A. & Srinivasa, S. (2009). CHOMP: Gradient optimization techniques for efficient motion planning. *2009 IEEE International Conference on Robotics and Automation*, pp. 489–494.

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Salunkhe, O., Stensöta, O., Åkerman, M., Berglund, Å. F. & Alveflo, P.-A. (2019). Assembly 4.0: Wheel hub nut assembly using a cobot. *IFAC-PapersOnLine*, 52(13), 1632–1637.

Shadrin, G. K., Alontseva, D. L., Kussaiyn-Murat, A. T., Kadyroldina, A. T., Ospanov, O. B. & Haidegger, T. (2020). Application of compensation algorithms to control the movement of a robot manipulator. *Acta Polytechnica Hungarica*, 17(1), 191–214.

Singh, R., Kukshal, V. & Yadav, V. S. (2021). A review on forward and inverse kinematics of classical serial manipulators. *Advances in Engineering Design*, 417–428.

Sucan, I. A., Moll, M. & Kavraki, L. E. (2012). The Open Motion Planning Library. *IEEE Robotics Automation Magazine*, 19(4), 72-82. doi: 10.1109/MRA.2012.2205651.

Taigman, Y., Yang, M., Ranzato, M. & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708.

Tsibulevsky, A. (2022). Nuitrack™ SDK. Retrieved on 2022-010-16 from: https://github.com/3DiVi/nuitrack-sdk/.

Uijlings, J. R., Van De Sande, K. E., Gevers, T. & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104(2), 154–171.

Vedaldi, A., Gulshan, V., Varma, M. & Zisserman, A. (2009). Multiple kernels for object detection. *2009 IEEE 12th international conference on computer vision*, pp. 606–613.

Wada, K., Okada, K. & Inaba, M. (2019a). Joint Learning of Instance and Semantic Segmentation for Robotic Pick-and-Place with Heavy Occlusions in Clutter. *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9558-9564. doi: 10.1109/ICRA.2019.8793783.

Wada, K., Okada, K. & Inaba, M. (2019b). Joint learning of instance and semantic segmentation for robotic pick-and-place with heavy occlusions in clutter. *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9558–9564.

Waldron, K., Waldron, K., Schmiedeler, J. & Schmiedeler, J. (2007). Handbook of Robotics Chapter 1: Kinematics.

Wang, J. & Olson, E. (2016). AprilTag 2: Efficient and robust fiducial detection. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4193–4198.

Yang, Y., Pan, J. & Wan, W. (2019). Survey of optimal motion planning. *IET Cyber-systems and Robotics*, 1(1), 13–19.

Yu, Y., Zhang, J., Huang, Y., Zheng, S., Ren, W., Wang, C., Huang, K. & Tan, T. (2010). Object detection by context and boosted HOG-LBP. *ECCV workshop on PASCAL VOC*.

Zargarbashi, S., Khan, W. & Angeles, J. (2012). Posture optimization in robot-assisted machining operations. *Mechanism and Machine Theory*, 51, 74–86.

Zhang, L., Lin, L., Liang, X. & He, K. (2016a). Is faster R-CNN doing well for pedestrian detection? *European conference on computer vision*, pp. 443–457.

Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W. & Bai, X. (2016b). Multi-oriented text detection with fully convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4159–4167.

Zhu, F., Wang, L., Wen, Y., Yang, L., Pan, J., Wang, Z. & Wang, W. (2021a). Failure Handling of Robotic Pick and Place Tasks With Multimodal Cues Under Partial Object Occlusion. *Frontiers in Neurorobotics*, 15, 11. doi: 10.3389/fnbot.2021.570507.

Zhu, F., Wang, L., Wen, Y., Yang, L., Pan, J., Wang, Z. & Wang, W. (2021b). Failure handling of robotic pick and place tasks with multimodal cues under partial object occlusion. *Frontiers in Neurorobotics*, 15, 570507.