NTNU

Specialization Project

Grasp selection in bin picking tasks for robotic manipulator arm with end-effector geometric constraints

Irja Gravdahl

Abstract

The objective of this project was to investigate grasp selection in bin-picking tasks for a robotic manipulator arm with end-effector geometric constraints, and to make a study of how easy it is for a robot arm to reach a given grasp in a bin-picking task. The importance of bin-picking in the automation picture is significant, with respect to what positive consequences such a system would have, should a more general solution be found. Considering its task-specific nature, bin-picking is a valuable issue to research where a mixture of different technologies come together in an intuitive and exciting manner.

By looking at a specific section of the workspace and investigating its level of reachability and using this as a grasp quality metric in terms of how easy it is to obtain this grasp pose for the robot manipulator, insight into the constraints of the robot emerged. Furthermore, investigating a specific region of the workspace, gave input to further optimal placement of the bin for this set-up, and what regions or approaches were optimal.

By searching the area chosen for inverse kinematic solutions and viable paths from a start configuration to map this part of the workspace, it was discovered that the average reachability of the area was at best 26.95%, indicating that further work with the workspace is a necessity, when this seems like a result with the possibility for improvement. Even though good results were obtained for inverse kinematic solution coverage of the chosen part of the workspace in this project, there was not always a path, which limited the total accessibility of the region.

Preface

As part of the final year of the 2 year masters degree program in Cybernetics and Robotics at the Norwegian University of Science and Technology a project is to be completed in the penultimate semester worth 7,5 credits. As part of this process each student is to complete a relevant project connected to ones chosen education program. This project is completed by 5th year student Irja Gravdahl, in the study program Robot Engineering and Vessel Control Systems, in cooperation with SINTEF Digital Trondheim. The project is supervised by Senior Researcher Esten Ingar Grøtli at SINTEF Digital Trondheim and Professor Kristin Ytterstad Pettersen at the Department of Engineering Cybernetics, NTNU.

I would like to thank my supervisor Esten Ingar Grøtli for his patience and assistance with this project, his invaluable insights and his valuable help. I would also like to thank professor Kristin Ytterstad Pettersen for excellent guidance and help throughout the semester. In addition I would like to thank Katrine Seel MSc at SINTEF Digital for her words of encouragement, insights about the system and otherwise continuous help throughout this whole project period.

Table of Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

This project will deal with aspects of grasp selection in bin-picking tasks for a robotic manipulator, more specifically the set-up employed by SINTEF Digital in Trondheim with end-effector geometric constraints. In the coming pages the following will be introduced; a brief introduction to bin-picking, a problem description, limitations of the project, its contributions and an overview of the project structure.

1.1 What is bin-picking?

Imagine that you are tasked with moving 10 apples from one crate to another. This seems a simple enough problem to solve. You grab one apple, with either hand, move the apple to its desired location and you put it down or drop it in the second crate. And then you repeat the process nine more times.

In this task you have used your eyes as a 3D sensor, scanned the environment, estimated the distance to the apples, registered its position and orientation with a "reference apple" in mind and decided which apple it is best to grab first based on how they all are positioned in the crate. You have used previous experience and knowledge to estimate its consistency, weight and mass centre. This enables you to know where to grab the apple for an optimal grip, how hard you should grab it, how tight you should hold it when you move it and when you can drop it into the bin without damaging it. Using your fingers and palm as a gripper with force feedback to hold the apple enables you to adjust your grip as to not damage it as well, as you can feel the force you are exerting on the apple.

When you scanned your environment you also knew how far away the second crate was and what the optimal path you should take to unload the apple was. The process of moving the apples was not difficult, but you used your own sensors and actuators in real-time to achieve the goal. Perhaps you also tried to optimize the process after having moved a few of the apples? Maybe you moved because you thought you could get the job done faster if you stood closer to the second crate, but then moved back because this made it harder to grab an apple from the first crate? All these aspects and processes we take for granted to solve this problem must be transferred to a robot for it to manage the same task, but how can we make it see its surroundings and let it know or teach it how it should grab the apple without damaging it? And how can we make a robot aware of whether it actually can grab the apple or not when it has figured out how to grasp it?

Robotic bin-picking is a classic robot problem, where the objective is to achieve a pick-and-place routine in a randomized environment. The robot is presented with a bin, containing some object or parts and its task is to pick an item, one at the time, and place them safely in the next bin. In an industrial setting this could for example be on to the next conveyor belt or a new work station. This is a tedious task for humans where we perhaps do not reflect on the complexity of the task, and being able to employ robots in our place would allow an increase in efficiency and predictable timing of for example an assembly process.

1.2 Problem description

The title of this project is "Grasp selection in bin picking tasks for robotic manipulator arm with end-effector geometric constraints" and focuses especially on the setup employed by SINTEF Digital Trondheim, see figure 1.1.

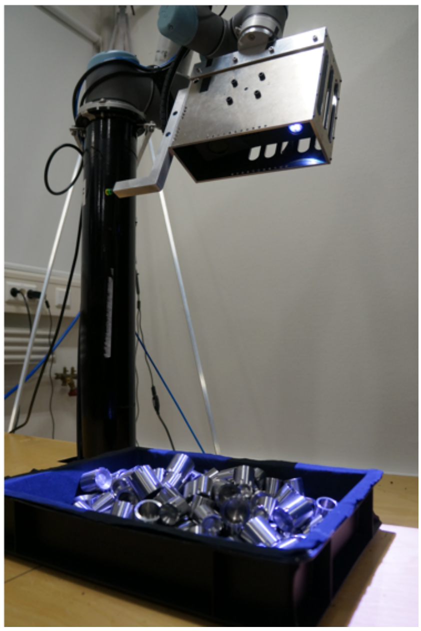

Figure 1.1: Picture of the SINTEF setup for bin-picking. Courtesy of Katrine Seel MSc SINTEF Digital

Bin picking is the problem of grasping objects randomly placed in a bin. This is a problem that often occurs in industrial settings where objects come out of a production line packaged in bulk, without isolating individual objects, and where the objects are transported to a second production line that subsequently must isolate and process these objects individually. Information from a 3D-sensor is used to compute many possible grasps based on how the objects are placed in the bin. When the 3D-sensor is attached to the robotic arm performing the grasps this imposes additional constraints on how the robotic manipulator arm can move while avoiding self-collisions and collisions with the bin or other parts of the environment. The overall goal of this assignment is to find a way to judge which grasps are favourable for grasping with the robotic manipulator arm.

The assignment consists of the following general bullet points:

- Make a literature review of state-of-the-art methods relevant to achieve the described goal.
- Design one or more metrics suitable to evaluate different grasps, which then can be used to judge the performance of different methods. The picking should preferably be carried out as fast as possible, but it might be useful to consider other metrics, for instance related to safety.

To summarize; this project will deal with and investigate how a given grasp can be judged in the sense of how easy this grasp configuration can be reached by the robot. In its simplest form, imagine that we are given a pose A, a position and orientation of what has been judged a favorable grasp, and a pose B which is the current pose of the gripper. How easy is it to achieve $A = B$? In this setting, what was just named pose A is not actually a pose, but a mere point and direction vector which is the output of the neural network grasp planner employed by SINTEF Digital (this will be explained later in section 3.2). When viewing figure 1.2, the objective is to obtain $z_{tcp} = z_{grip}$, and to optimize or at least make feasible the rotation about this now common z-axis (placement of the x and y-axis), as to obtain a successful grasp, and judge how accessible it is.

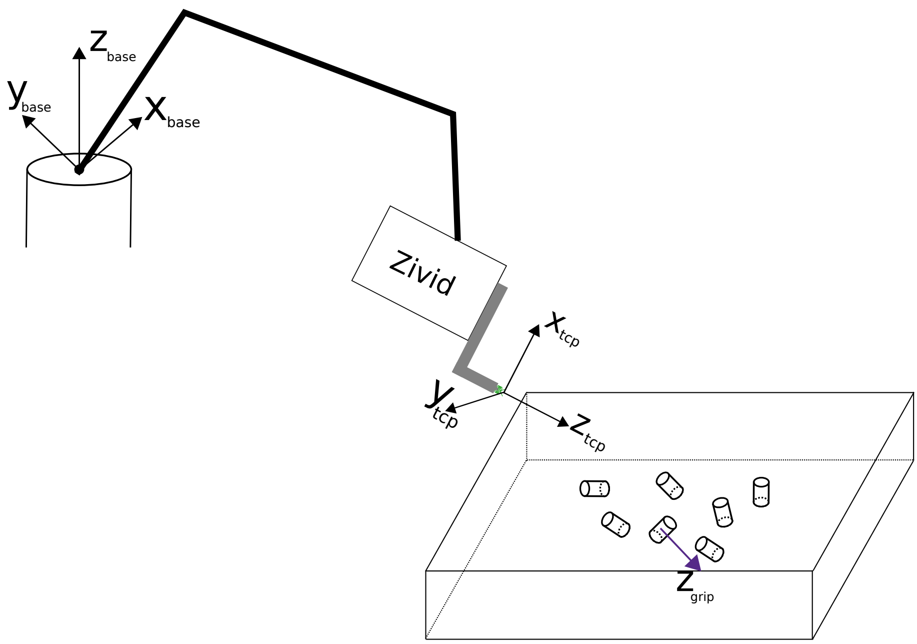

Figure 1.2: Simple schematic of the system

1.3 Limitations

This project will deal with grasp selection for an eye-in-hand (3D sensor at the end-effector) bin picking system, the physical setup at SINTEF Digital Trondheim, and will not consider implementations for other set-ups or a general investigation which can be independently implemented for an arbitrary system. Apart from a literature review of existing work and solutions for comparison and discussion, the majority of the following work will be done with the SINTEF system in mind.

Furthermore, a solid background in robotics is needed to complete the assignment and as such, basic robotics is also presented as background for the project. In addition, to be able to judge a grasp, a solid understanding of what a good grasp actually is is also presented in the literature part of this project. More weight will be placed upon the hardware part of the system, and the inner workings of for example deep neural networks is outside the scope.

1.4 Contribution

The contributions of this project are:

- An overview of bin-picking in a technology and manufacturing perspective, chapter 2.
- An overview of robotic grasping, what a so-called "good grasp" is, and how they are selected based on mathematical principles.
- An overview of literature considering and implementing robotic manipulator reachability and its use in combined grasp and motion planning, chapter 5.
- Investigation of reachability properties of an UR5 eye-in-hand robotic manipulator with end-effector geometric constraints, chapter 6.
- Implementation and testing on the UR5 a reachability metric considering inverse kinematic solutions and the existence of paths, and presented the results as heatmaps for easy visualization, chapter 6.

1.5 Project structure

This report is divided into four main parts; the introduction, the literature review, method and design employed to reach the goal, and finally a conclusion which discusses the results and gives suggestions for future work.

In the literature review part of the report, an introduction to bin-picking in industry context will be given, along with an overview of the importance of the subject. Furthermore, a brief presentation of robotics is given, before the SINTEF system is presented. The literature part of the project is rounded of with a chapter on robotic grasping and one on the combination of grasping and motion planning and its importance.

In the part considering method and design, a thorough description of the work executed on the SINTEF set-up is given. A description of the experiments and the results are presented.

Finally, in the concluding part of this report, a brief discussion of the results and the project will be given, along with a list of suggestions for future work and some concluding remarks.

Part II

Literature review

Chapter 2

Bin-picking

In "Automatic Grasp Generation and Improvement for Industrial Bin-Picking", Kraft et al. (2014) describe the bin-picking problem by these simple, yet individually challenging, steps;

1. Use a sensor system (typically based on a camera, range scanner or a combination of these) to detect an object in the bin and its pose
2. Select an appropriate way to grasp the object
3. Execute the appropriate motion and grasp the object
4. If the grasp was successful, move the object to a desired location

2.1 Randomized bin-picking

Figure 2.1: Picture of the parts to be picked from the bin in the SINTEF set-up. Courtesy of Katrine Seel MSc SINTEF Digital

In automated assembly lines, the problem of locating, gripping and moving an object that is randomly positioned and unsorted is known as random bin-picking. Many, if not all, bin-picking systems today are tailored to a very specific need and application, and are simply known as bin-picking systems. Should bin-picking be achieved and solved generally such that a system is equipped to pick-and-place any type of object regardless of the environment, it can become a routine robot application in the industry and assist many small scale businesses in increasing their efficiency and expanding their production. The "dream" is that a bin-picking system will become routine and easy to use in all parts of industrial production and not just in large scale production as is the main rule today (Bogue, 2014).

Random bin-picking has the potential to reduce costs and improve efficiency in production on a large scale. By eliminating the human element in for example sorting and pick-and-place operations, one will be able to increase the output volumes of for example production lines. Today, without a bin-picking system, large specific machinery take up a large volume of the production locations and are not applicable for other tasks than the ones they are specifically designed for. To be able to increase the flexibility of a system could assist in cost reduction and reduce the cycle time of a process.

A general solution to randomized bin-picking has not been reached yet, despite the rapid development in the field of computer vision and the known vast benefits of a solution, should it be found. Computer vision is important as it enables us to equip a robot with eyes, and this development has gained momentum in the search for a solution. Since a bin-picking system consists of different technologies, the dependency on accurate and fine-tuned calibration procedures has long been at the centre of the issue. If the system fails on two out of three grasps throughout the production time, the reliability of the system is not sufficient to be used industrially on a large scale. Keeping this is mind, there have been improvements with both calibration and set-up routines (Bogue, 2014).

Random bin-picking is not one-size and must be adjusted to the need of the buyer in terms of robot reach and payload, shape, size, geometry and characteristics of the parts and how randomly they are arranged is critical, and following this argument a completely general solution seems challenging. One could for example not use the same system to pick large cylinders of metal, uniformly distributed in a bin, as one would use to pick glass vials positioned more systematically. As bin-picking is far more application-specific than many other robot applications it is only just now gaining momentum in regards to finding a solution. (Bogue, 2014)

2.2 Motivation and importance

Random bin-picking or unstructured bin-picking has the potential to revolutionize industry. Manufacturing technologies have increased their use of robots and this has pushed the boundaries of an already competitive market in terms of productivity and innovation. One of the many advantages of bin-picking is that it is an intuitive process to understand and a solution is not limited to academics or specialists, there is a long list of interested parties, motivated to find a solution (Marvel et al., 2012)

One of the many challenges associated with bin-picking is dependency on many forms of technology seamlessly integrated, working together towards the goal; localizing, approaching, gripping and moving. An image of the robots surroundings must be generated to find the bin and the parts to be picked. The important information such as orientation and position of the object must be separated from the non-critical information in the image. In a human context, we filter an image we see in real-time dependant on our goal. If I would like to pick up my water bottle, I would scan my desk, process the image to find the bottle, and grab it without thinking twice, even though I probably calculated and executed a perfect pick-and-place operation. Since the parts in the bin are not precisely positioned and we do not know their exact location, traditional robot vision is not sufficient, due to varying lighting conditions, a lack of distinctive features and entanglements and collisions with other objects in the container, or the container itself (Bogue, 2014).

From the image, an estimate of the objects location must be calculated and its position and orientation derived, this is known as pose estimation and is a challenge in itself. The pose must be calculated relative to the camera and the robot base, this demands a good calibration of the camera to sustain repeatability and exact gripping. Furthermore a path to approach the grip must be calculated and executed, the object must be grasped and a new path must be calculated to carry the object to its desired location. In the last few years rapid successes in the field of computer vision, or machine vision, and 3D cameras as well as the algorithms that process the image information, have made this technology readily available. Some types of software are even tailored to function specifically with regards to the bin-picking problem. (Bogue, 2014)

The acquisition of a single part from a collection of parts is considered an integral part of manufacturing, including, but not limited to, palletizing, packing, assembly and kitting. Since bin-picking is not at the moment one size fits all, current goals include easing the transition of using a system to pick object A one day and then picking object B the next day, and simplifying the effort needed to reprogram the system for a different use. (Marvel et al., 2012)

According to Marvel et al. (2012), there are three primary challenges associated with the integration of bin-picking in manufacturing: sensing, hardware issues and solution integration issues. Sensing is related to difficulty in sensor development associated with pose estimation, finding on object's pose in the environment, and the associated challenges with lighting, reflections, shadows etc. Hardware issues include the gripper, robot and the parts to be picked, and the challenge of choosing the right hardware for a specific application without over- or under equipping the system. Solution integration problems include factors such as cost, financial burden, time used to train and tune the system, and how long it takes to re-purpose the system in terms of flexibility. (Marvel et al., 2012)

In essence, the bin-picking problem is vast in its application and valuable to the industry as well as an interesting classical robot problem in academia. It is of importance and worth looking into in terms of the positive ripple effects it would have in both large and small scale industry, to increase output volume and contribute to a higher degree of automation in tedious, repetitive tasks.

Chapter 3

Hardware

This chapter will be used to give an overview of robot manipulators since this is the tool at the centre of the operation. Without a robot up for the task, problems will arise in the execution of the picking. After this brief repetition of robotics, the SINTEF system will be briefly presented, before the next chapter introduces robotic grasping.

Figure 3.1: ABB's IRB 120 6 axis robot (ABB, 2018)

3.1 Robot manipulators

The definition of a robot is *a programmable, multifunctional manipulator designed to move material, parts, tools or specialized devises through variable programmed motions for the performance of a variety of tasks*. From this definition we can extract the concept of reprogrammability, meaning that we may again and again utilize a robot in different tasks, and this gives a robot its utility and adaptability. Some of the most commonly stated advantages of the introduction of robots is decreased labour costs, an increased flexibility with regards to its reprogrammability and the creation of more humane working environments where robots are able to perform dangerous, repetitive and dull operations. (Spong et al., 2006)

3.1.1 Workspace

A robot is composed of links with joints in between to form a kinematic chain. A joint can either be revolute, have a rotation, or prismatic, have an extension. The workspace of a robot is the volume swept out by the manipulators end-effector, or more easily put, its reach. The end-effector is its tool, commonly attached at the end of the final joint. This space is limited by the manipulators geometry and is designed with the task the robot is to perform in mind. Constraints on motors and actuators will need to be taken into account as well as the robot work cell and possible obstacles. The workspace is commonly split into the dexterous workspace and the reachable workspace. The reachable workspace is a subset of the dexterous workspace and is the space reachable by the manipulator and the dexterous space is the set of points the robot can reach with an arbitrary orientation of the end-effector. The dexterous workspace is thus the full space it can reach. (Spong et al., 2006)

Consider a simple planar robot with two DOF, one elbow joint and one wrist joint. Moving the joints changes the (x, y)-coordinates of gripper and the elbow. The robot can thus be described by the coordinates of the elbow and gripper, (x_e, y_e) and (x_g, y_g) respectively, relative to the environment. These four coordinates describe the full workspace of the robot and the manipulation of these coordinates allow full reach in the workspace and can describe the location of the manipulator in the environment. If a robot was to avoid an obstacle or locate an object in its work cell, using workspace coordinates is advantageous since the coordinates of the robot position and the position of the object would be given in the same coordinate frame. However, not all coordinates by this representation are within reach, since the robot is non-linearly constrained by its own geometry, for example by the link between the two joints. Using the configuration space assists in solving this problem. (Russell and Norvig, 2016)

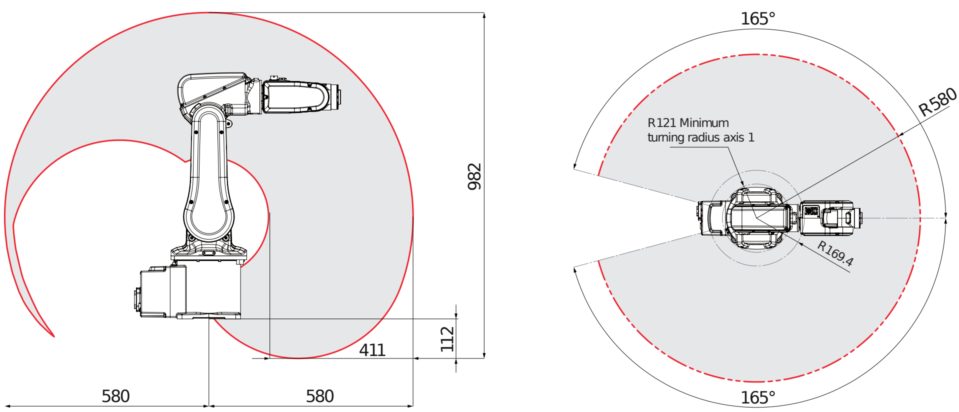

Figure 3.2: Workspace of the IRB 120 (ABB, 2018)

3.1.2 Configuration space

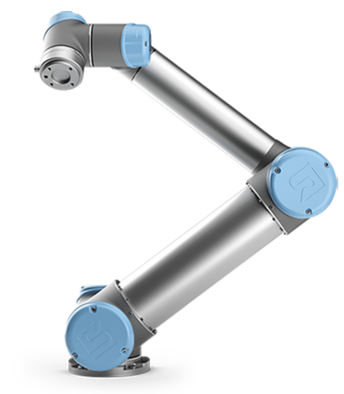

Figure 3.3: Photo of the UR5 6DOF robot (Universal Robots, 2018)

The configuration of a manipulator is the complete specification of the location of every point on the manipulator. The set of all possible configurations is known as the configuration space (Russell and Norvig, 2016). Instead of representing the robot position in Cartesian coordinates as in section 3.1.1, we represent in the configuration space the robot as a function of its joint variables. If we further build on the example already mentioned, we may represent the robot configuration of a two DOF planar robot by its joint angles instead of joint positions; (Russell and Norvig, 2016)

$$(x_g, y_g) \wedge (x_e, y_e) \iff (\theta_g, \theta_e) \tag{3.1}$$

If the robot is a rigid body, its base is fixed and we know the values for the joint variables it is straightforward to find any point on the manipulator. If these demands are met, we may denote a vector q and state that a robot is in configuration q when the joint variables take on the values $q_1, ..., q_n$, where $q_i = \theta_i$ for a revolute joint and $q_i = d_i$ for a prismatic joint. The subscript n denotes the degree of freedom for the rigid body, DOF for short. For example, a robot with six revolute joints will have six degrees of freedom since we have three parameters to denote its position and three to denote its orientation in $\mathbb{R}^3$. (Spong et al., 2006)

3.1.3 Rigid bodies and motion

In robot manipulation, geometry of the three-dimensional space and rigid motion plays a central role. Even though this project is specifically concerned with grasp selection and the ability to reach a given grasp, these subjects are important. To select a good grasp with the robot end-effector, the robot behaviour and limitations are central, to make sure a grasp is valid in terms of constraints. In order to represent the relative position and orientation of one rigid body with respect to another, for example an end-effector and an object to be grasped, we attach by convention coordinate frames to each body and then specify the geometrical relationship between these coordinate frames.

3.1.3.1 Rotations

A rotation matrix in three dimensions belongs to the group $SO(3)$, the special orthogonal group (Spong et al., 2006). The set of all matrices that are orthogonal and have a determinant equal to the identity matrix exist in this group, defined by:

$$SO(3) = \{\mathbf{R} \mid \mathbf{R} \in R^{3\times 3}, \quad \mathbf{R}^T\mathbf{R} = \mathbf{I} \quad \text{and} \quad \det \mathbf{R} = 1\} \tag{3.2}$$

For rotations around the principal axes, x, y and z, the following holds and can be proven, where $s\alpha = \sin\alpha$ and $c\alpha = \cos\alpha$:

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}, \mathbf{R}_y(\theta) = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}, \mathbf{R}_z(\psi) = \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

The rotation matrix $\mathbf{R}_b^a$ from a to b can be interpreted in two ways; either as the coordinate transformation from b to a or as a rotation matrix where a vector $\mathbf{p}^a$ in a is rotated to the vector $\mathbf{q}^b$ by $\mathbf{q}^a = \mathbf{R}_b^a\mathbf{p}^a$. The rotation of a composite rotation is the product of the rotation matrices, such that (Egeland and Gravdahl, 2002):

$$\mathbf{R}_c^a = \mathbf{R}_b^a \cdot \mathbf{R}_b^c \tag{3.4}$$

3.1.3.2 Rigid motion

After now having defined a rotation matrix $\mathbf{R}$, we can define rigid motion by also introducing a vector $d \in \mathbb{R}^3$. A rigid motion is a pure translation together with a pure rotation. Suppose p is attached to a frame $o_1x_1y_1z_1$, with local coordinates p^1. We can then express the coordinates of p in frame $o_0x_0y_0z_0$ using:

$$p^0 = \mathbf{R}_1^0 p^1 + d^0 \tag{3.5}$$

This concept can be extended to three coordinate frames such that: (Spong et al., 2006)

$$p^0 = \mathbf{R}_1^0\mathbf{R}_2^1 p^2 + \mathbf{R}_1^0 d_2^1 + d_1^0 \tag{3.6}$$

$$= \mathbf{R}_2^0 p^2 + d_2^0 \tag{3.7}$$

3.1.3.3 Transformations

The concept of a rotation matrix can be expanded to include both orientation, as before with the rotation matrix, and position of one coordinate frame relative to a another frame. This is what is called a homogeneous transformation matrix T which is defined as:

$$T = \begin{bmatrix} \mathbf{R} & d \\ \mathbf{0}^T & 1 \end{bmatrix} \in SE(3), \tag{3.8}$$

where $SE(3)$ is the special Euclidean group defined by the following expression:

$$SE(3) = \left\{ \mathbf{T} \mid \mathbf{T} = \begin{bmatrix} \mathbf{R} & d \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad \mathbf{R} \in SO(3), \quad \mathbf{d} \in \mathbb{R}^3 \right\} \tag{3.9}$$

When utilizing a homogeneous transformation matrix, we may describe a rigid body in terms of either a pure rotation, a pure translation or a combination of the two relative to another coordinate frame. The inverse of the transformation matrix is defined as $\left(T_b^a\right)^{-1} = T_a^b$ and the product of composite transformations is the same as for rotation matrices $\mathbf{T}_c^a = \mathbf{T}_b^a \cdot \mathbf{T}_b^c$. (Egeland and Gravdahl, 2002)

Transformation matrices form the basis of the Denavit-Hartenberg convention which is a popular way to describe the forward kinematics of a robot configuration. The convention can further be used to find the dynamics of a robot manipulator, as will be further explained in section 3.1.5.

3.1.4 Forward Kinematics

The forward kinematics problem is concerned with finding the position and orientation of the last link in the kinematic chain, often the position of the end-effector with an attached tool, for example a gripper. This is done by attaching a coordinate system to each link in the chain making up the robot. Usually one does this design by attaching coordinate system $o_i x_i y_i z_i$ to link i, such that the coordinates of the i^{th} link are constant in coordinate system i.

With each joint, a joint variable q_i, is associated; often $q_i = \theta_i$ for revolute joints and $q_i = d_i$ for prismatic joints. The inertial frame is usually attached at the robot base and denoted as the 0-frame, $o_0 x_0 y_0 z_0$. When each link has been assigned a coordinate system and with the homogeneous transformation matrices in mind, suppose that A_i is the transformation matrix giving the position and orientation of coordinate frame $o_i x_i y_i z_i$ with respect to $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$. Since we have introduced the general coordinate q_i, A_i is a function of only one joint variable dependant on if the joint is revolute or prismatic, $A_i = A_i(q_i)$.

By following this thought we may now express every point on the robot in terms of its joint variables q_i and by multiplying $A_1(q_1)$ up until $A_n(q_n)$ we can find the position and orientation of the last attached coordinate system given in the inertial frame:

$$H = T_n^0 = A_1(q_1) \cdot \ldots \cdot A_n(q_n) \tag{3.10}$$

The forward kinematics problem is thus to find the pose of the final attached coordinate system, usually given in the inertial frame. This is the problem of knowing the joint variables and multiplying your way outwards in the kinematic chain. By using different conventions as the DH-convention this is a solvable problem with one solution.

3.1.4.1 Reachability

The reachability of a robot manipulator to a target is defined as its ability to move its joints and links in free space in order for its hand to reach the given target (Ying and Iyengar, 1995). When transferring this to a bin-picking system, the given target is the given grasp we would like to reach. How well a robot can reach this given grasp is closely related to its reachability. If the reachability of the robot is "good" for a given grasp we can be more certain that the robot will be able to reach the grasp.

3.1.5 Denavit-Hartenberg convention

The Denavit-Hartenberg convention, or DH-convention, is used to find the forward kinematics of a manipulator. By following this convention of choosing reference frame the problem is simplified and each transformation A_i is given as the product of four basic transformations. The final product for each link will be as follows:

$$A_i = \mathbf{R}_z(\theta_i) \cdot \mathbf{Trans}_z(d_i) \cdot \mathbf{Trans}_x(a_i) \cdot \mathbf{R}_x(\alpha_i) \tag{3.11}$$

$$= \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.12}$$

where θ_i, d_i, a_i and α_i are parameters associated with link and joint i. By a systematic and proven way of choosing the coordinate frames for the manipulator one can decrease the number of variables needed from six to four. For a complete overview of how to choose the frames according to this convention see for example the book "Robot modeling and control" by Spong et al. (2006).

3.1.6 Inverse kinematics

The general inverse kinematics problem is the opposite of the forward kinematics problem. Given the position and orientation of the end-effector or the desired origin of the last attached coordinate frames, how must the joint variables q_i be assigned to achieve this configuration? So, given a 4×4 transformation matrix with the structure of equation 3.8, find the q_i needed up until n to result in this transformation matrix. The issue with inverse kinematic solutions are that they are redundant, there exists an infinite number of solutions. This is generally not a problem on paper, but when programming this solution with a robot, the redundancy leads to choices. The simplest example is a two-DOF planar robot, which will have two solution, elbow joint up or elbow joint down. This is a choice that needs to be made by the engineer, often with a specific task in mind, or considering the robot work cell.

3.1.7 Motion planning

When dealing with robots, the objective is usually to allow the robot to execute some task, re-position and complete the same, or a different, task. Since all robotic manipulation includes some degree of motion, motion planning is central for a robotic system. We differentiate between path planning and trajectory planning.

3.1.7.1 Path planning

As previously defined, a robot has a unique configuration space which is the set of all possible robot configurations. To plan a collision free path, ensuring that the robot does not make contact with an obstacle is central. The configuration space can further be divided into the *configuration space obstacle* and the *free configuration space*. The configuration space obstacle is the set of all robot configurations for which the robot collides with an obstacle and the free configuration space is the set difference between the whole configuration space and the obstacle. Denoting the configuration space as $\mathcal{Q}$, the obstacle as $\mathcal{QO}$ and the set of collision-free configurations as $\mathcal{Q}_{free}$ gives the following relation

$$\mathcal{Q}_{free} = \mathcal{Q} \setminus \mathcal{QO} \tag{3.13}$$

The path planning problem can be summarized as finding a path from q_s, an initial joint configuration, a starting point, to q_f, a final joint configuration where obstacles are avoided when traversing the path. Formally defined by Spong et al. (2006); "a collision free path from q_s to q_f is a continuous map, $\gamma : [0, 1] \to \mathcal{Q}_{free}$ with $\gamma(0) = q_s$ and $\gamma(1) = q_f$". Methods for path planning are many and the most popular solution is to treat it as an optimization problem using for example gradient descent methods. This will not be discussed further here as it is outside the scope of this project. Path planning does not depend on the variable time, whilst the next section on trajectory planning, does just that. (Spong et al., 2006)

3.1.7.2 Trajectory planning

A trajectory, unlike a path, is a function of time $q(t)$. By employing the same notation as for path planning, $q_s = q(t_0)$ and $q_f = q(t_f)$, where the difference $t_f - t_0$ is the time needed to execute a given trajectory. From this we can present path planning as a subset of trajectory planning where a path is a trajectory completed during one time unit. In some cases, paths are specified by a series of end-effector poses $T_6^0(k\Delta t)$, where the inverse kinematics solution gives a sequence of joint configurations. Since the trajectory is time varying, velocity and acceleration information can be found by differentiation, and as such these variables can be controlled and used in the planning. Consider for instance the velocity close to an obstacle in comparison to the velocity which can be employed in the free space. For repetitive trajectories, it might be more efficient to employ a "jog-and-learn" approach where the robot is guided though the desired motion and repeats this, instead of solving the inverse kinematics problem. (Spong et al., 2006)

3.2 SINTEF system

Figure 3.5: Picture of the SINTEF set-up for bin-picking, overview. Courtesy of Katrine Seel MSc SINTEF Digital

The project is done in cooperation with SINTEF Digital Trondheim and solutions to the grasp selection problem will be done with their system in mind. Here follows a brief system description and overview.

The SINTEF system consists of a six degree of freedom revolute UR5 robot manipulator as can be seen in figure 3.3 mounted on a pillar. At the end-effector, a Zivid camera is mounted within a protective camera house where a vacuum gripper is attached at its side. Below the robot is a table where a rectangular plastic container is placed. The container is lined with felt to limit reflections from the lighting in the room. In the bin are cylinders for picking with the vacuum gripper. In figure 3.5 we see a photo of the setup, and a very simple schematic of the system and the objective is shown in figure 1.2.

Initially, the vision system captures a point cloud of the bin filled with parts from a scan configuration set by the user. This point cloud is sent to a deep neural network trained on simulated data whose output is an optimal grasp. This deemed optimal grasp is given to the system as a point and a unit vector, indicating the direction the grasp should be executed from and where the vacuum gripper should end up to pick up the part. This unit vector can be viewed as the z-axis of an incomplete coordinate system, where the x and y axis must be generated at a later time.

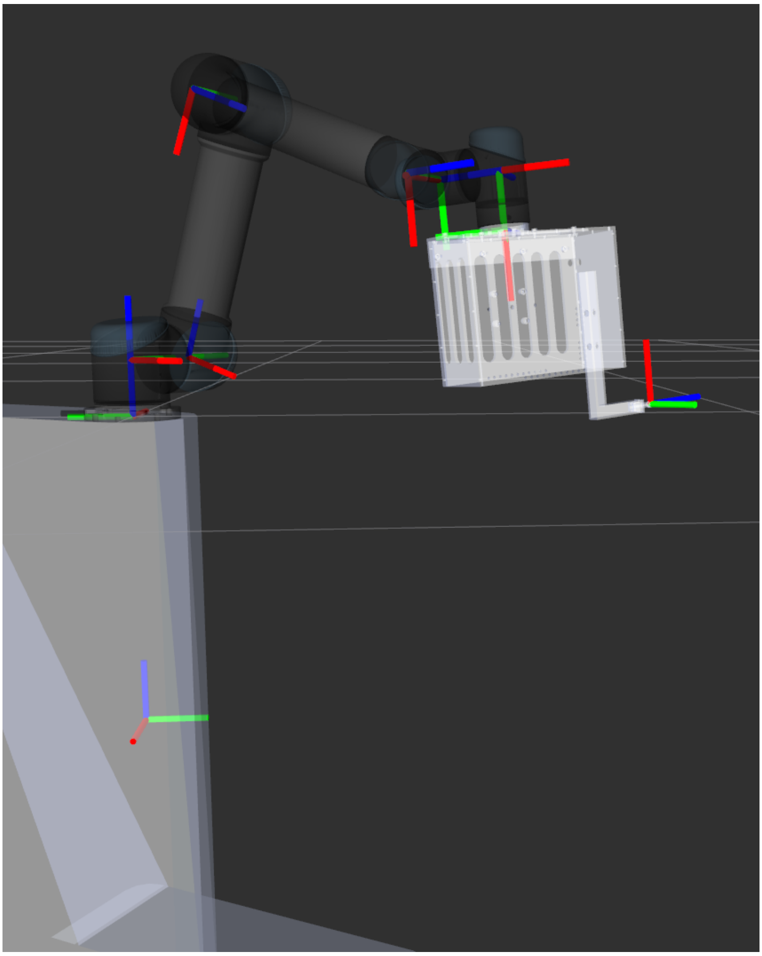

Figure 3.4: Screenshot of SINTEF set-up as visualized in Rviz with attached co-ordinate systems at joints

Chapter 4

Grasping

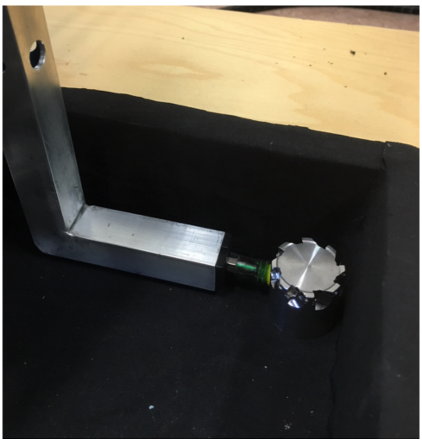

Figure 4.1: Successful grip with vacuum gripper. Courtesy of Katrine Seel MSc SINTEF Digital

Robotic grasping differs from human grasping, mostly by the different nature of the hands. Many robotic grippers such as the two- and three-finger grippers seek to mimic the behaviour of human hands. Human hands are not task-specific, in terms that we may use our hands in a variety of ways, where the three most important functions are to explore (haptics), restrain and manipulate objects. Since a robotic application often is task-specific, adapting the type of gripper one employs may be beneficial in hopes that it will increase the success of the application. In bin-picking and other robotic applications more emphasis has been placed on restraining and manipulation, rather than exploration. Restraining is often referred to as fixturing, whilst manipulation with the fingers is referred to as dexterous manipulation (Bicchi and Kumar, 2000).

In this chapter, robotic grasping is presented in short along with a few concepts important for what we call a ”good grasp”. The chapter is closed with an introduction to grasp planners which can take an object and a hand as inputs and give a grasp as an output. This part of the project does not take into account with what degree of ease a robot may reach said grasp, but defines what a grasp is as well as surrounding topics.

4.1 Defining grasping

To grasp is defined as to take something and hold it firmly, which is also what we would like robot to do when we ask it to grasp an object. Even though we would like the robot to grab and hold an object, does not mean it must do so with five fingers and a palm.

With regards to the definition of a grasp and the notion of good grasps, a few aspects are central. A grasp needs to resist external forces and torques to be maintained, and in that regard, it is necessary to define what a wrench is, as well as the wrench space. Furthermore follows a brief explanation of what force and form closure properties involve with regards to grasping.

4.1.1 Wrench and wrench space

From screw theory we find the concepts of screws, twists and wrenches. When a generalized force acts on a rigid body this force consists of a linear component which is pure force, and an angular component which is pure moment acting at a point. This force/moment pair is defined as a wrench $\mathbf{w}$. (Murray et al., 1994)

A wrench is presented as a vector $\mathbf{w} \in \mathbb{R}^p$, where in three dimensional space, $p = 6$. The vector is made up of a force $\mathbf{F} \in \mathbb{R}^{\frac{p}{2}}$ and a torque $\tau \in \mathbb{R}^{\frac{p}{2}}$, such that $\mathbf{w} = \left(\begin{smallmatrix}\mathbf{F}\\ \tau\end{smallmatrix}\right)$. Wrenches are used in robotic grasping to determine the net effect of forces (wrenches) that are applied at the contact points between the fingers and the object.

Furthermore we may define the magnitude of a wrench as

$$\|\mathbf{w}\| = \sqrt{\|\mathbf{F}\|^2 + \lambda \|\tau\|^2}, \tag{4.1}$$

where choosing $\lambda = 1$ is equivalent to measuring $\|\mathbf{w}\|$ according to an L_2 metric (equivalent to the Euclidean norm). The choice of λ is somewhat arbitrary since the torque vector can be scaled differently than the force vector. (Ferrari and Canny, 1992)

The forces and torques acting on an object can be represented in a 6-dimensional space having three variables for the three components of the total moment acting on the object, and three more for the total force. This space is called the wrench space and is denoted $\mathcal{W}$. (Ferrari and Canny, 1992)

4.1.2 Force and form closure

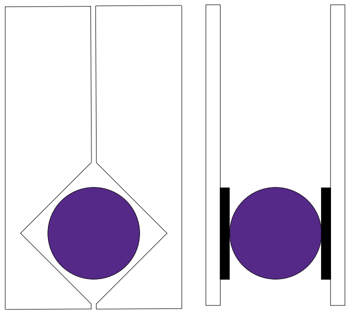

Figure 4.2: Form and force closure illustration. Illustration inspired by Bajd et al. (2010)

The force and form closure properties of a grasp are concerned with the capability of a grasp to partially or completely constrain an object and its motions, as well as apply an arbitrary contact force on the object without violating friction constraints at the contact points (Bicchi, 1995).

A grasp is said to be force closure if it is possible to apply forces and torques at the contact points such that any external force and torque can be balanced by the robot (dependant on the robot kinematics). Thus, put differently, a grasp is said to be force closure if it is in equilibrium for any arbitrary wrench. So, if there exists, for an arbitrary wrench w, a vector λ that satisfies the contact constraints inequalities such that

$$W\lambda = w, \tag{4.2}$$

where W is the combined wrench matrix, the grasp is defined force closure. (Bicchi and Kumar, 2000)

In the analysis of force closure, one considers frictional forces, whilst one in form-closure does not consider the friction component (Bicchi, 1995). Form closure is defined as complete constraint of an object, like holding a rubber ball in your fist, where the object grasped can withstand any disturbance wrench. By the same notation, a grasp is said to be form closure if the following is fullfilled:

$$W\lambda = 0, \tag{4.3}$$

and W has full rank and $\lambda > 0$. Form closure is a stronger condition than force closure and can be viewed as force closure with frictionless contact points. (Bicchi and Kumar, 2000)

4.2 Good grasps and grasp selection

In their 2014 paper, Kraft et al. place a focus on grasp selection and generation and ways to decrease the failure rates of grasps with bin-picking specifically. The paper in its entirety is based upon the assumption that all objects in the bin are the same, and known. The process of grasp selection can also be broken down into a few points, regardless whether one imposes limitations or not: Choose a set of "good grasps" $G = \{g_1, ..., g_n\}$ that covers the object in $SE(3)$ as well as possible.

And for each grasp g_α, a priority π_α should be defined based on an estimate of the success probability of that grasp.

The question of what a "good grasp" is, is complex and depends on the objective. For example; Is it of interest to place the object a certain way or in a certain configuration in the next bin, or when placing the item onto a conveyor belt in a bin-picking application? Are differences in material affecting where it should be grabbed? How can it be certain that the grasp will be maintained?

The early literature on this has been dominated by the use of metrics and the computation of grasp quality evaluated with regards to the operation objective. Grasps that fulfill the metric and has a high grasp quality have been labelled "good". *A popular notion of a good grasp is typically defined with an approximation of the form or force closure of a grasp, meaning how well a grasp can resist externally applied wrenches (forces, torques and combinations thereof)* (Kraft et al., 2014)

In their 1992 paper "Planning optimal grasps", Ferrari and Canny address the problem of planning optimal grasps, or "good grasps". In their work they present two optimality conditions that consider the total finger force and the maximum finger force exerted by the robot on the object. They further formalize the intuition of judging a grasp by considering the ratio between the magnitude of the maximum wrench to be resisted over all possible direction, and some notion of the applied (finger) force.

The criteria are formalized using various metrics on a space of generalized forces. Grippers exert forces and torques on the grasps object through the contact points. They ask the following question; Given the position of the gripper and the object to be grasped, how can we say "this is a good grasp"? In the paper they base their two criteria on that the force closure condition defined in section 4.1.2 is satisfied. Their work on optimal grasps have been referenced heavily in the literature.

4.3 Grasp plannners

Grasp planners are ready-made computer bases containing several options for hand and object interaction. Grasp planners are a tool for both gripper designers and those who implement grasping systems as one has the ability to investigate ones design of a gripper or the interaction between a gripper and an object without having to perform new full-scale experiments every time a new test needs to be undertaken.

4.3.1 GraspIt!

Figure 4.3: GraspIt! logo (Miller and Allen, 2018)

GraspIt! is an interactive grasping simulator that can import different robotic hands and objects and evaluates the grasps made by the hands on the object, making it easier to simulate both robotic hand design and functionality of systems. The focus on the grasp analysis in the simulator has been on force closure property grasps which are useful for pick and place operations, and the quality metrics for this property assumes nothing about the space of external forces that might be applied to the grasped object during the operation. As of 2004 the grasp planner does not include a sophisticated trajectory generator or path planner that is able to plan an approach from the robot initial pose to the desired grasp pose in space, but this functionality is listed as a point under future directions.

Furthermore grasp analysis is used to assess the quality of the grasp by examining the properties of the grasp. When the object and hand touch forces can be transmitted along the contact normal, and the Coulomb law is used in the planner to determine the magnitude of forces acting in the tangent plane of the contact that can be resisted by friction. (Miller and Allen, 2004)

4.3.2 OpenRAVE

Figure 4.4: OpenRAVE logo (Diankov and Kuffner, 2018)

The OpenRAVE (Open Robotics and Animation Virtual Environment) architecture supports a range of robotic tasks, and was at first designed to handle autonomous handling of objects. Initially, the focus of the planner was on grasping objects. This includes calculating contact points between the end-effector, (or tool provided a transformation exists from tool centre point to end effector), and the object to be grasped. Furthermore, it included calculations needed for computing force closure for grasps as well as grasp stability. OpenRAVE has expanded in terms of including path planning to meet demands for such functionality. (Diankov and Kuffner, 2008)

In their paper Diankov and Kuffner (2008) claim that, "most grasping research considers free-floating end-effectors able to approach a target from any direction". As this is not the case in most real world pick-and-place applications, inclusion of the environment is necessary. This is to improve grasping success in unison with optimal grasp configurations between the end effector and the object, as well as the limitations of the robotic arm in use. This leads to the inclusion of reachability, also in the cases where reachability is not simply a measure of whether there is an inverse kinematic solution to the problem or not (such as in mobile base applications). (Diankov and Kuffner, 2008)

4.3.3 OpenGRASP

Figure 4.5: OpenGRASP logo (León et al., 2018)

OpenGRASP is built upon OpenRAVE, and has some functionality for collision checking with the environment. León et al. (2010) have added a robot editor which allows for more streamlined implementation of the robot in use for a specific application in the simulation. This is a step in the direction of using and implementing something like reachability and environmental- or self-collisions which limit a robot from reaching a good grasp. In OpenGRASP one can define the kinematics of a robot via its Denavit-Hartenberg parameters. As this environment is based on OpenRAVE, sets of stable grasps are readily available and the robot in the simulation can manipulate the object and analyze contact points. (León et al., 2010)

To autonomously manipulate an object in the environment, there is a need for an IK solver that maps grasp locations into robot configuration joints. This is available in OpenRAVE, and thus also in OpenGRASP, and the solver returns all possible solutions. In their test of grasping known objects, the grasp simulator was used (off-line) to find a set of grasps that are force closure and as such, usable. By also utilizing a vision system, the environment was mapped and the poses of the objects estimated. That information was sent to the planning plug-in in the software.

Chapter 5

Grasping combined with motion planning

During the theory search for this project information on grasp physics, grasp properties, measures of grasp quality, methods to determine good grasps and grasp planners are found readily available in many different shapes and forms. The problem of finding a good grasp seems to have been solved for a majority of hands and fingers and for a multitude of objects.

The objective of this project however deviates slightly from these subjects. The basis for the problem to be looked into presupposes an already available grasp, which already has been ranked the optimal candidate for grasping. The question is not "is this a good or optimal grasp?", but rather, "given this optimal grasp, can the robot reach it? And if so, how easy is it to reach?"

5.1 Separated solutions

Bin-picking is, as previously mentioned, a concoction of different technologies, and branches within those technologies. Attempting a solution of the bin-picking problem by solving it part by part seems a good strategy due to the complexity of the system as a whole. Combining solutions to subsystems is reasonably assumed to lead to the solution of the system as a whole. As such, following this reasoning, much research has been done on one of two things; finding good trajectories to reach a desired pose and finding high quality grasp candidates. This leads to a new problem: an optimal pose for grasping may not be feasible to reach for the robot due to constraints in the workspace. Akinola et al. (2018)

There exists extensive previous work on the notion of grasping an object given that the end-effector is already at the appropriate contact point, to initiate the actual grasping. If one is given a grasp which has proved itself to be good through the appropriate metrics (for example the Ferrari-Canny metric), it does not matter that it is perfect if the robot cannot reach it (Akinola et al., 2018).

Taking into account that the robot must be able to reach the grasp pose is detrimental for a successful bin-picking system. Given an optimal grasp from a grasp planner, or a vision-based system as is the case with the SINTEF set-up, at least an inverse kinematic solution must exist for the given pose. If there are no constraints on the reachability of the robot, the optimal grasp is the best ranked grasp from the planner. However, considering that the arm kinematics and the reachability of the robot is a constraint on the system, additional consideration is necessary.

In its simplest form, checking for an inverse kinematics solution at the grasp candidate poses takes the arm kinematics and the robot into the equation. Once one has a list of grasp candidates, the robot constraints are introduced and the reachability comes into play. If there are no additional constraints, this is enough to find out if a grasp is feasible or not (Saut and Sidobre, 2012). If the workspace is clear of all obstruction and constraints other than the demand for reachability, an inverse kinematics solution is a sufficient condition. The process then goes from

→ Find good grasp candidates, given the hand close to the object pose
→ Choose the optimal candidate based on grasp quality measures

to

→ Find good grasp candidates
→ Check the IK solution for candidates
→ Choose the optimal candidate based on grasp quality measures and that an IK solution exists

5.2 Combining the problem

Combining the issues of motion planning and grasping, has improved the overall picking results in comparison to solving them separately in multiple papers. A few of these papers are presented here. In 2007, Berenson et al. published their work on "Grasp planning in complex scenes", in 2009, Zacharias et al. published their paper "Online Generation of Reachable Grasps for Dexterous Manipulation Using a Representation of the Reachable Workspace", and in 2018, Akinola et al. published their paper "Workspace aware online grasp planning". The methods and ideas in these papers will be discussed briefly in this section.

Berenson et al. (2007) combined grasp analysis and manipulation planning techniques to perform fast grasp planning in obstructed scenes. They also highlight the issue with regards to planning a grasp where both gripper and object are disembodied, and where the manipulator kinematics do not play a part. An illustrative example is planning to grasp an object sitting on a table. If both gripper and object were free-floating during the grasp planning phase and only operating on optimizing grasp properties, a grasp planner would spend as much time trying to grasp the object through the table, as it would attempt to find grasps from above. Furthermore it is stated, that the goal is not only to select a grasp that is stable, but also to ensure that it is feasible, which is exactly in terms of what this project is about.

Berenson et al. (2007) coin the term "Grasp scoring function" where they combine the demand for force closure, the feature of the object's environment and the robot kinematics. The method consists of a precomputation phase and an online one. The precomputation uses a geometric model of the manipulator and the object to be grasped to build a set of feasible grasps. The online phase computes a score for each grasp using the grasp scoring function where the grasps are ranked according to this function before they are validated by trying for an IK solution and checking for collisions in the environment. Once a feasible grasp is found, a motion plan is created to grasp. If the motion plan should fail, the algorithm returns to attempting a plan for the grasp ranked second. By combining the environment and the robot kinematics as well as grasp properties, they succeed in finding collision-free paths to reachable and stable grasps.

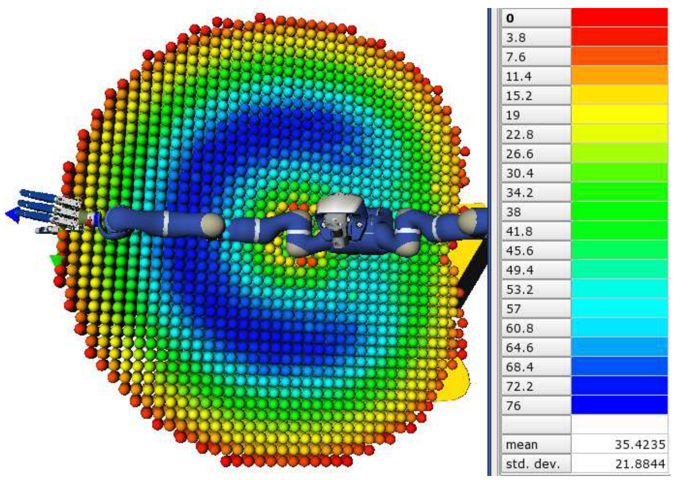

Figure 5.1: "Shows the reachability spheres across the workspace. The workspace representation was cut (across the workspace) for better visibility of the structure" (Zacharias et al., 2007)

Zacharias et al. (2009) look at two methods to ensure reachability of grasps. The first method integrates the robot's inverse kinematics into a grasp planner, and the second method integrates their previously developed model of the reachable workspace. In their first method, the grasp planner finds a valid grasp, before this pose is checked for an inverse solution and whether or not the robot can reach it.

Their second method uses a model of the reachable workspace denoted the "capability map", presented in "Capturing robot workspace structure: representing robot capabilities" (Zacharias et al., 2007). To create the capability map, the theoretically possible workspace is enclosed by a cube and divided into smaller cubes. In each of these smaller cubes, a sphere is inserted and on each of these spheres n points are uniformly distributed. In each point, a coordinate system is generated which serves as targets for checking the inverse kinematics. These spheres visualize the reachable points in the workspace, and when concatenated, the reachable workspace. This model can be used to estimate the workspace of the robot and inspect reachability across it.

The capability map is a representation of the reachable sphere map. By providing a grasp planner with this model, as a model of the robot's reachable workspace, the planner is able to predict the reachability of a grasp. See figure 5.1 for the reachability spheres. Notice that this image along with the caption was collected directly from Zacharias et al. (2007) and the paper "Capturing robot workspace structure: representing robot capabilities", without any modification.

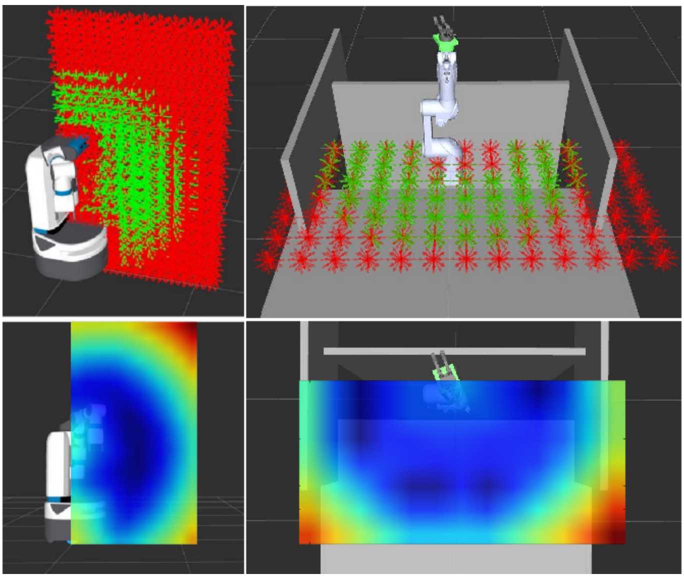

Figure 5.2: "Top Row: Visualization of cross sections of the precomputed reachable space for a Fetch Robot and Staubli Arm with Barrett Hand. Green arrows represent reachable poses, red arrows unreachable. This space is computed offline, once for a given robot. Bottom Row: Signed Distance Field generated from the above reachability spaces." (Akinola et al., 2018)

Akinola et al. (2018) have designed a workspace aware online grasp planner that considers the robot reachability along with grasp quality when deciding on the optimal grasp. By solving these two aspects, motion planning and grasp quality, jointly they show that it is possible to reduce the needed grasp planning time as well as improve the success rate of the grasps. The goal is reached by introducing a bias in the reachability space that drives the end-effector along with the grasp planner into accessible regions of the planning scene.

They generate a densely sampled reachability space offline, where a check for an inverse kinematic solution is done, to establish whether that point in the space is reachable. The reachability space is post-processed to become a signed distance field, as can be viewed in figure 5.2. When creating the reachability space by just checking for an inverse kinematic solution, the boundary to the unreachable portion of the space is included. By also considering the distance to this boundary, d_{sdf}, a gradient becomes apparent. If there are any static obstacles in the planning scene, these can also be added before computing the reachability space and the signed distance field, such that these become part of the constraints on the planner. Both the reachability space and the signed distance field are computed once, offline, before the rest of the process is done online.

Since their method uses an optimization technique called simulated annealing which handles nonlinearities well (Akinola et al., 2018), along with a significantly smaller search space due to the knowledge of the reachability space, online grasp planning can be done at a reasonable price computationally. Considering planners without the same notion, a lot of time will likely be spent on planning grasps that are not feasible. With the energy function describing reachability along with grasp property, the optimization process drives the end-effector towards reachable areas of the workspace by minimizing the energy, and finding "low-energy" areas in the workspace. After grasp planning, trajectory planning is done before the grasp is executed. The energy function is:

$$E = G + \alpha R, \tag{5.1}$$

where G is a measure (metric) of the grasp quality considering only the hand and the object, whilst R is a measure of the reachability of the grasp pose and α is a weighting parameter. (Akinola et al., 2018)

Akinola et al. (2018) build on the GraspIt! grasp planner when implementing their work, where they add their energy function containing the combined grasp quality term and reachability term. See figure 5.2 for the reachable space and the signed distance field. Notice that this image along with the caption was collected directly from Akinola et al. (2018) and the paper "Workspace aware online grasp planning", without any modification.

When including the robot kinematics and the environment in the grasp planning process, a higher success rate is achieved, as shown by the works presented here.

Part III

Method and design

Chapter 6

Implementation and simulation

Based on the theory sections of this project, some form of testing on the SINTEF set-up was essential to observe the application of some of the concepts discovered. Considering the system set-up for this project, attempting to implement something similar to Zacharias et al. (2009) and Akinola et al. (2018), and their work on reachable workspace was of interest. Looking into how available a portion of the workspace is, was of interest in terms of grasp success. By surveying the workspace of a robot and identifying the reachable space in it, it could be possible to say something about the availability of the grasps in that area. By looking into this for the SINTEF set-up, it is possible to outline a method of checking the quality of a grasp set.

→ Identify grasp candidates
→ Check for IK solution for candidates
→ Check if a collision-free path exists
→ Investigate and conclude on the reachability of this area

This chapter will be regarding own implementations and testing on the Sintef-setup, inspired by the work done by Zacharias et al. (2009) and Akinola et al. (2018) to investigate reachability. Furthermore, part of the project was to come up with and test a metric for the reachability of different optimal grasps, this is also covered in this chapter.

6.1 Metrics

A metric is a measurement of a specific characteristic of a system or phenomenon. In this project it is assumed that any grasp available for testing, such as the grasps generated to execute experiments on, are optimal and already fulfill some kind of grasp metric, for example the Ferrari-Canny metric. Hence, grasp metrics are outside the scope of this work, and assumed to hold.

In terms of metrics for how easy it is to reach the proposed grasp, that is a different demand. Two serial demands must hold for the grasps if they are to be considered successful, regardless of the fact that the grasps themselves are deemed optimal. The first demand we put on the grasps is naturally that there must exist an inverse kinematics solution for the robot at the grasp pose, otherwise grasping is impossible. Secondly, there must also exist a valid collision-free motion plan to said pose. If pose conditions hold, the robot is able to move to this pose and grasp the object. If there exists an IK solution and a plan exists, we deem the grasp reachable.

Tying this to for example the energy function Akinola et al. (2018) presented, where it is assumed that the grasp part of the energy function can be neglected due to its optimality, also making the trade-off variable r redundant, we get the following simple expression to minimize:

$$E = G + \alpha R \tag{6.1}$$

$$E = \alpha R \tag{6.2}$$

$$E = R \tag{6.3}$$

Since our now very simple energy function is only one variable, the objective is simply to choose the region of the workspace proven itself to have the highest degree of reachability, and as such having scored the highest according to the defined metric.

6.2 Description of experiment

From Akinola et al. (2018) and other sources it is known that a grasp is reachable if there exists a motion plan to move the robotic arm from an initial or current configuration to the goal configuration that places the gripper at the grasp pose. The number one reason causing a motion plan not to be found is a lack of inverse kinematic solution at the grasp pose. Other reasons include self collision or collisions with other objects in the workspace (Akinola et al., 2018). From figure 3.5, it is clear that with this set-up, self-collision and collisions with the environment is a possibility. Not finding an inverse kinematic solution is also a risk. We observe potential self-collisions with the camera house and collisions with the table, the bin (edges) and the pillar.

The program Rviz was used to visualize the robot, it's workspace and the generated grasp poses. The plug-in MoveIt! was used to plan a path to the potential grasps.

Observing figure 6.1, the objective of the bin-picking process is to align the coordinate system attached to the tool with the coordinate system at the grasp pose.

This is indicative of a successful grasp. At the bottom of the image, several coordinate systems are generated at a set height. These are meant to represent a set of grasp poses. A subset of the workspace was sampled and poses were generated with a set distance apart in both x and y direction with constant z, considering that the bin to be picked from is planar. 18 points were created, and the same 27 poses were created in all of them to investigate both the properties of the points in the space, and the poses.

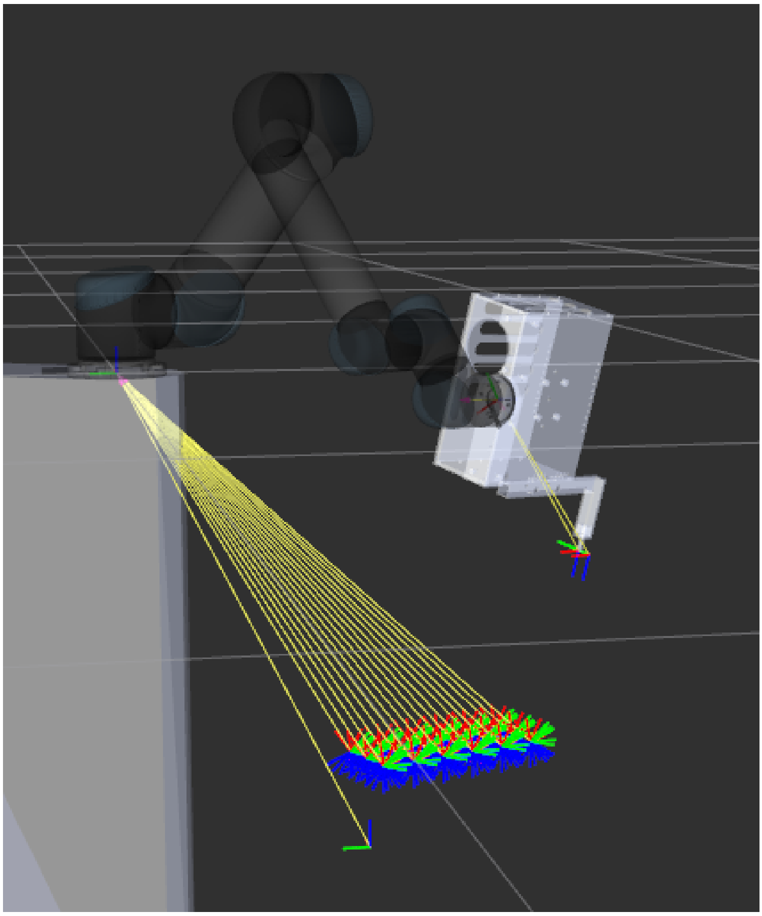

Figure 6.1: A selection of grasp poses (evenly spaced coordinate system in the bottom of the image) and the robot with camera house as viewed in Rviz

At each point, there are several rotated coordinate systems. This was done to investigate the reachability of multiple grasps at the same position, but with different orientation. This is needed due to the extra degree of freedom in the final wrist joint. Since the grasp given by the deep neural network is given only as a point and a unit axis, an arbitrary rotation around this axis might provide a solution.

The coordinate systems which are seen in the figure are thus manually generated, to check if a rotation provides a solution.

The goal was, for each possible pose at each point, to save the following information: the transformation matrix of the grasp pose given in the base coordinate frame; T_{grip}^{base}, a Boolean variable for whether or not there existed an inverse kinematic solution at this pose, and if there was one, the joint angles, q, needed to reach it. By first investigating if an IK solution was available, we avoided the task of attempting to plan a path for a grasp known to be unreachable. Also, by this two-step method, some computational effort is avoided, and the ability to check ones results underway is available. An option to checking for an inverse kinematic solution and then for a path with the corresponding joint angles, is to plan directly with the grasp pose as this also is a possibility with MoveIt!.

Furthermore, after establishing whether or not there existed an IK solution, planning a path to check for collisions was necessary. That there exists an IK solution is a good property, but is not sufficient for investigating reachability. So, if there was an IK solution for a given grasp pose, the corresponding joint angles were given to a path planner that checked for potential collisions. Since the program has available the description of the robot, the pillar and the camera house, using the plug-in MoveIt! would be able to provide the correct results.

After checking all poses for an IK solution and a collision-free path, if both properties were fulfilled, the grasp pose was said to be reachable and saved. If we compare this method to the work done by Akinola et al. (2018), it is clear that they sample the whole workspace, and we only a small section of it. Considering that the placement of the bin in the SINTEF set-up currently is decided, initial investigation into the reachability of this area and areas around this location was of interest.

6.3 Results

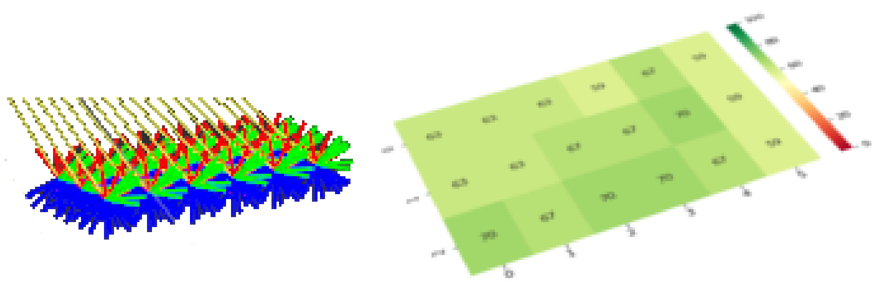

Figure 6.2: Result presentation

The next sections present the results from the experiment done on the SINTEF set-up. The experiment was split into two parts, one part for checking for solutions to the inverse kinematics problem at all generated poses, and one for investigating whether or not there existed a motion plan to the poses using MoveIt!s built in planner. The reason for using this planner was that it has readily available the robot description file and was able to take into account the camera house attached at the end-effector. In figures 6.3 to 6.7 the leftmost corner towards the reader corresponds to the leftmost bottom corner of the heatmaps, as indicated in figure 6.2.

6.3.1 Checking for inverse kinematic solutions

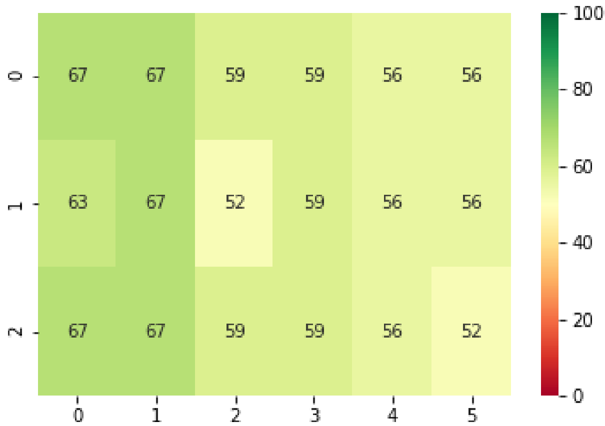

Figure 6.3: IK coverage using the scan configuration as seed to the solver

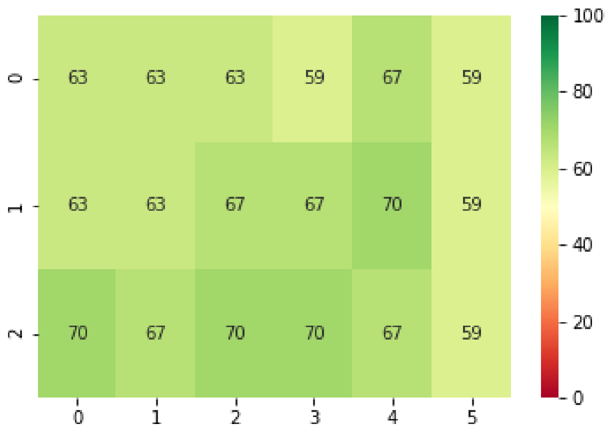

Figure 6.4: IK coverage using the first set of initial joint values as seed to the solver

As stated in the above section, the first test done on the grasp poses is the inverse kinematics test. The solver is called for each and every one of the poses, and indicates with a Boolean value whether or not there exists a solution for the particular pose. In this project we did two rounds of experiments, with two different seeds. A seed functions like an initial condition, and indicates where the solver should start looking for a solution to the problem. To investigate both the performance of applying different seeds as well as finding a good seed to increase the success of grasping, two seeds were considered.

The first round was done with the robot scan configuration as the seed, the pose where the manipulator scans the bin looking for the optimal grasp. This is also the pose where it needs to be able to plan a path from. By using this seed, we obtained the results seen in figure 6.3.

When looking at figure 6.3 we see the IK coverage when utilizing the scan configuration as seed to the solver. It is somewhat surprising to see that there is a higher coverage to the left of the area investigated and a lesser coverage to the right. It was somewhat expected that there would be more problematic to find solutions at the left most side as there previously have been some issues regarding collisions with the base when coming in from this side. However, since paths are planned at a later time and the inverse kinematic solution does not take into account collisions, it is reasonable nonetheless. The reason for not having obtained a higher coverage could be due to the constraints of the camera house.

The second round of experiments used the first set of joint angles generated as the seed. When the first pose was generated, there existed an inverse kinematic solution, and the joint angles needed to reach this, was used as a starting point for the solver. This means that this seed is significantly closer to the area where the grasps are, and where the gripper will need to be placed. By instead using this seed, we obtained the following results, presented in figure 6.4.

When viewing figure 6.4 we see a slightly higher coverage of found solutions. Considering that the poses are exactly the same, it seems that the choice of seed has a significant impact. This is supported by the observations made by Zacharias et al. (2007) where they state that since an inverse kinematic solution for a redundant robot (we have some freedom regarding the rotation of wrist 3) does not have a unique solution, a starting solution that is already near the desired solution is beneficial.

6.3.2 Checking for motion plans

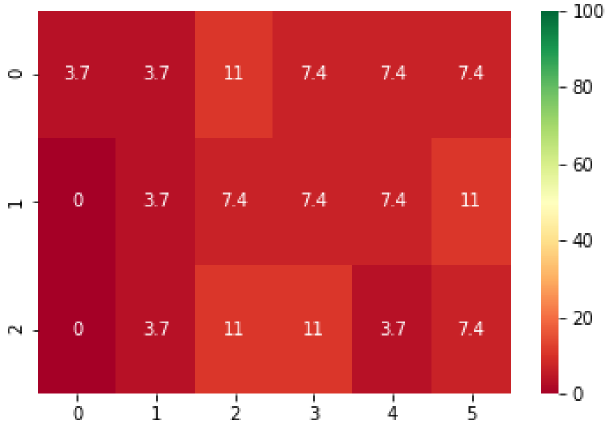

Figure 6.5: Motion plan coverage using the scan configuration as seed to the IK solver

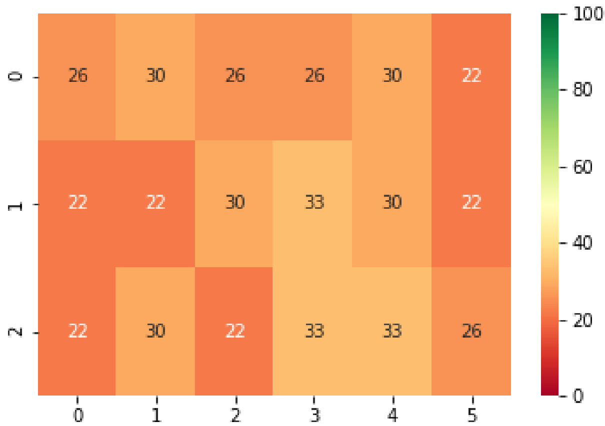

Figure 6.6: Motion plan coverage using the first set of initial joint values as seed to the IK solver

By using the results obtained for whether or not we had an IK solution, the next step was to see if it was possible to plan a path to obtain these joint angles and in extension if the grasp pose was reachable for the robot. It is of course expected that there exists no path if there is no IK solution, and in the following figures we see the statistics showing whether or not there exists a path from the scan configuration to the different poses. In figure 6.5 we see the results of attempting to plan a path to joint angles found from the IK solver when using the scan configuration as the starting point, and in figure 6.6 the same objective, but using the first obtained joint angles as seed.

Looking at figure 6.5, these are not very good results. We observe that there is close to no coverage of viable motion plans to the left, but fortunately there are some possibilities when going to the right, where perhaps it is more viable for the robot to avoid a collision with the pillar it is placed on. In figure 6.6 however, there is a significant increase in solutions, and there is a slightly better chance of a viable motion plan if one can approach the "rectangle of poses" from the right. Considering that the different seed to the solver is the only variable that has changed, this must contribute to the jump in solutions found. The next step in the procedure, was to combine the results.

6.3.3 IK + motion-plan

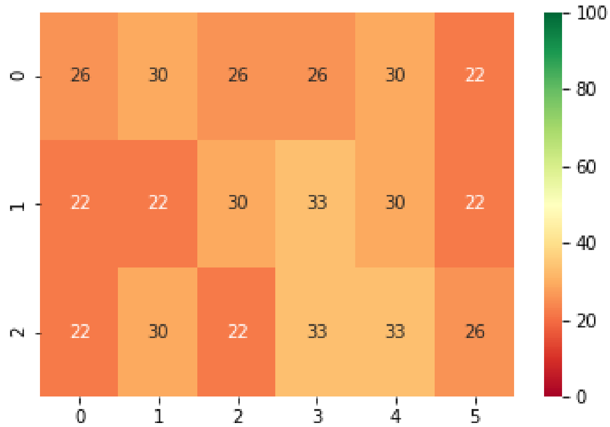

Figure 6.7: Combined reachability in the space of interest

Combining the results obtained for the inverse kinematics solution and the motion-plan results give the following overview. These points represent the percentage of reachable poses at each of the points considering all the poses at the point. This is simply an AND-operation between the Boolean values for the existence of an inverse kinematic solution, and whether or not there existed a path to the pose. It comes as no surprise then, that the existence of a motion plan plays the role of limiting reactant and that the same result as for motion plan coverage is obtained, as can be viewed in figure 6.7 for the second attempted seed. Hence, the result for using the scan configuration as seed was identical as the results seen in figure 6.5.

6.3.4 Interpreting the results

	Scan config. as seed	*First joints as seed*	*Difference*
$\overline{IK}$	59.67%	64.81%	5.14%
$\overline{Path}$	6.38%	26.95%	20.57%
$\overline{IK + Path}$	6.38%	26.95%	20.57%

Table 6.1: Overview of average results

When comparing the two seeds, there was a 5.14% increase in IK coverage from the first to the second. This is not a huge increase, but significant enough to conclude that it seems the better option. However, when planning a path, there was a 20.57% increase in motion plans found when changing the seed to the latter. In addition to this significant increase, the overall success of planning a path was significantly lower than that of an inverse kinematic solution. It is difficult to conclude exactly what has caused this effect, but it at least seems like the choice of seed to the solver is of great importance for feasible solutions for path planning.

Based on the first figures, indicating the coverage of inverse kinematics solutions, these seem like good results considering all the different rotations in each point. That there is a 70% coverage in some areas of the region of interest seems to be a positive result considering the constraint on the system in terms of the camera house. The increase in coverage when changing the seed to be closer to the solutions, also fit nicely with expectations.

It was of interest to observe that there existed so few motion plans compared to inverse kinematic solutions, considering that the closest obstacle in the environment to the robot is the pillar upon which it is placed. Even considering the size of the camera house and the constraint this puts on the system regarding self-collisions at the wrist, it seems a little peculiar to see that less than half of the poses which there existed an inverse kinematic solution for has a feasible path to it.

Since the previous sections only dealt with the quality of different **points** and rotations around them (18 points in space which contained 27 different poses), looking into the "quality" of the different **poses** was also of interest as a final check of the results obtained. Especially considering the results obtained when attempting path planning the following figures show how the different poses performed in the region of interest. For example when viewing figures 6.8 and 6.9, the first pose attempted had a full score for IK, it found a solution in all the points, whilst two of the points where unfeasible in terms of a path.

We may also observe that there was a much higher coverage of obtainable paths for the first five poses that were investigated. Considering the risks of a collision with the base or a self-collision with the camera house, perhaps an idea is to approach the grasping from an angle which minimizes these factors in terms of robot poses and not only which points in the workspace that have an inverse kinematic solution.

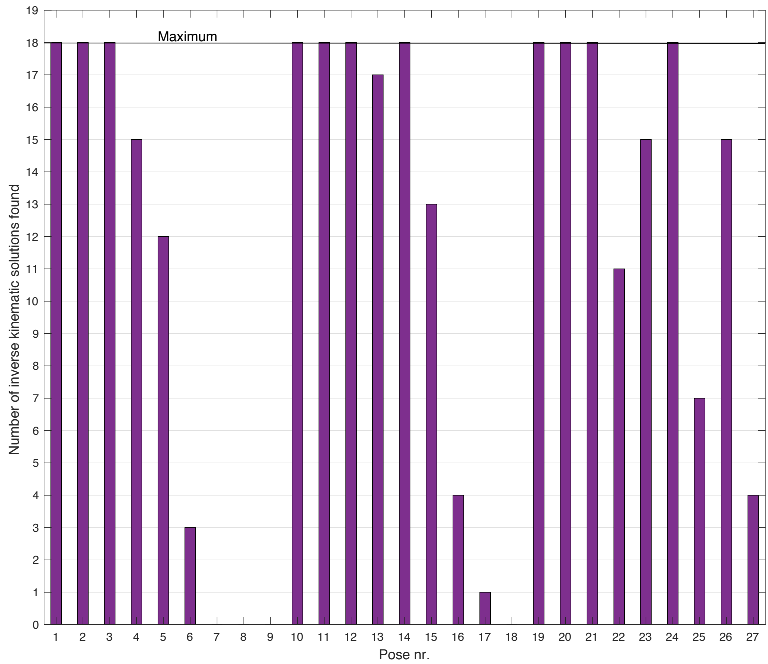

Figure 6.8: Inverse kinematics coverage for the different poses

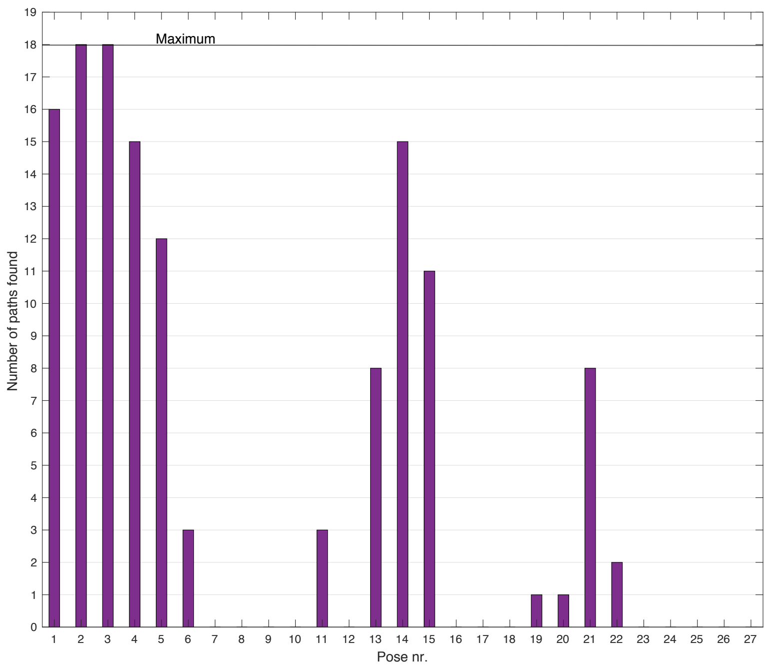

Figure 6.9: Path planning coverage for the different poses

Part IV

Conclusion

Chapter 7

Discussion and results

7.1 Discussion

This project has been on grasp selection in bin-picking tasks for robotic manipulator arm with end-effector geometric constraints, with a basis in the set-up employed by SINTEF Digital Trondheim. The project has looked into different aspects of robotic manipulators at the heart of the bin-picking problem and focused on the investigation of reachability as a metric to evaluate different optimal grasps, and how attainable they were. Furthermore, this simple metric of reachability was investigated on the real system.

Based on the interpretations of the results in the previous chapter, we have confirmed that an inverse kinematic solution is a necessary condition for a grasp to be reachable, as well as there must exist a path to reach this solution for it to be completely reachable. The investigations made into the SINTEF set-up with regards to the reachability metric were inspired by state-of-the-art work on reachability and the combination of grasp planning and motion planning as a single problem to be solved, as shown in chapter 5.

Based on the results presented, it seems that the seed given to the inverse kinematics solver has a great impact on the results obtained, especially with regards to path planning. This effect should be investigated further, to see if this truly is the case, or if there might be something wrong with the data. Furthermore, the reachability of the area investigated, using the seed which gave the most path planning solutions, ended up with a reachability averaging 26.95%. This result has potential for improvement, and parts of the future work should include a more comprehensive look at the whole workspace of the robot with all its constraints, to look into if there exists a better placement for the bin.

7.2 Future work

With regards to future work based on this project, a few things could be valuable to investigate further to obtain a more comprehensive study:

- Expand the part of the workspace that is explored for reachability to investigate where the optimal placement of the bin should be based on the reachability condition. By doing a wider analysis of the workspace, an optimal placement of the bin could reveal itself such that one knows one has the best base for bin-picking. It could also be beneficial to test different path planners and inverse kinematics solvers and seeds.
- Look at more comprehensive metrics in terms of ranking grasps from best to worst. In this project several grasps were looked at, but in terms of ranking, this is for now binary. Either the grasp was reachable or it was not. The points are then only ranked based on this score. An idea might be to look more closely at how much time the process of checking for an inverse kinematics solution takes in addition to planning time, as well as of course the execution time the process of picking an object at a specific grasp takes. Other metrics could include ranking grasps which demands for the least change in the robot configuration.
- Look more into different robot configurations and how others have chosen their configuration to best achieve good results. A theory search is required to map this, and which set-ups have yielded the best results. Also consider others who have used the UR5 robot and their solutions.
- It is worthwhile to look more into different types of set-ups in regards to sensor placement. As of now the 3D camera is mounted at the end-effector of the robot, and the gripper is placed on the side of the camera house resulting in geometric end-effector constraints. There are at least two different methods for camera placement; eye-in-hand and object-in-hand. These two methods could be investigated further to do an analysis of the placement of the gripper and if a change in set-up could increase the probability of reaching a grasp by increasing the reachable space of the robot.
- A more comprehensive study of the workings of the SINTEF set-up and the possibility to use the machine learning and computer vision to increase the probability for a good grasp. Look more at the software aspect of the system, and not limit the study to mostly hardware sensitive aspects.

7.3 Concluding remarks

The literature this project is based on gives insight into the importance of bin-picking in industry as well as the technological challenges associated with it. Combining the problem of grasp planning and motion planning as described in the literature review, has proven to increase the success rate of grasping significantly, even to a point where grasp planning online has been a feasible alternative to grasp databases and off-line created look-up tables. Based on these results, and own findings through testing and simulation, the combination of these robot problems includes the robot and the environment in the grasping process, which seems detrimental for success.

In this project the reachability of a UR5 manipulator with end-effector geometric constraints has been investigated. As a metric to rank with what ease the given optimal grasp poses were obtainable, a section of the workspace was investigated for the existence of an inverse kinematic solution and the existence of a motion plan to the said grasp poses. The coverage of inverse kinematics solutions were in the 70% range, and quite similar in value when comparing the two seeds used with the inverse kinematics solver, only an increase in around 5% separated the first seed from the (closer to the solution) second seed. However, when the process of looking for, and planning for, feasible paths from the set scan configuration in the bin-picking system, a much less favorable trend in coverage became apparent. The best results gave an average value of around 25% success, which is significantly less than the coverage for inverse kinematic solutions. To conclude this part of the project, some future work in investigating this prominent trend in less coverage for motion plans is of interest.

Bibliography

ABB (2018), 'IRB 120, for flexible and compact production'. Photo of the IRB 120. Visited: 14.09.18.
URL: *https://new.abb.com/products/robotics/industrial-robots/irb-120*

Akinola, I., Varley, J., Chen, B. and Allen, P. K. (2018), 'Workspace aware online grasp planning', *IROS 2018* .

Bajd, T., Mihelj, M., Lenarčič, J., Stanovnik, A. and Munih, M. (2010), *Robotics*, Vol. 43 of *Intelligent Systems, Control, and Automation: Science and Engineering*, Springer.

Berenson, D., Diankov, R., Koichi, N., Satoshi, K. and Kuffner, J. (2007), Grasp planning in complex scenes, *in* '2007 7th IEEE-RAS International Conference on Humanoid Robots', pp. 42–48.

Bicchi, A. (1995), 'On the closure-properties of robotic grasping', *International Journal of Robotics Research* **14**(4), 319–334.
URL: *https://doi.org/10.1177/027836499501400402*

Bicchi, A. and Kumar, V. (2000), Robotic grasping and contact: a review, *in* 'Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)', Vol. 1, pp. 348–353 vol.1.

Bogue, R. (2014), 'Random bin picking: has its time finally come?', *Assembly Automation* **34**(3), 217–221.

Diankov, R. and Kuffner, J. (2008), Openrave: A planing architecture for autonomous robotics, Technical report, Robotics Institute, Carnegie Mellon University. Pittsburgh, Pennsylvania 15213, CMU-RI-TR-08-34.

Diankov, R. and Kuffner, J. (2018), 'Welcome to OpenRAVE'. Accessed: 30.10.2018.
URL: *http://openrave.org/*

Egeland, O. and Gravdahl, J. T. (2002), *Modeling and Simulation for Automatic Control*, Marine Cybernetics, Trondheim, Norway.

Ferrari, C. and Canny, J. (1992), Planning optimal grasps, *in* 'Proceedings 1992 IEEE International Conference on Robotics and Automation', pp. 2290–2295.

Kraft, D., Ellekilde, L.-P. and Jørgensen, J. A. (2014), *Automatic Grasp Generation and Improvement for Industrial Bin-Picking*, Springer Tracts in Advanced Robotics, Springer, book section Chapter 8, pp. 155–176.

León, B., Ulbrich, S., Diankov, R., Puche, G., Przybylski, M., Morales, A., Asfour, T., Moisio, S., Bohg, J., Kuffner, J. and Dillmann, R. (2010), Opengrasp: A toolkit for robot grasping simulation, *in* N. Ando, S. Balakirsky, T. Hemker, M. Reggiani and O. von Stryk, eds, 'Simulation, Modeling, and Programming for Autonomous Robots', Springer Berlin Heidelberg, pp. 109–120.

León, B., Ulbrich, S., Diankov, R., Puche, G., Przybylski, M., Morales, A., Asfour, T., Moisio, S., Bohg, J., Kuffner, J. and Dillmann, R. (2018), 'Welcome to OpenGRASP'. Accessed: 30.10.2018.
URL: *http://opengrasp.sourceforge.net/*

Marvel, J. A., Saidi, K., Eastman, R., Hong, T., Cheok, G. and Messina, E. (2012), Technology readiness levels for randomized bin picking, *in* 'Proceedings of the Workshop on Performance Metrics for Intelligent Systems', ACM, 2393114, pp. 109–113.

Miller, A. T. and Allen, P. K. (2004), 'Graspit! a versatile simulator for robotic grasping', *IEEE Robotics & Automation Magazine* **11**(4), 110–122.

Miller, A. T. and Allen, P. K. (2018), 'GraspIt! User manual'. Accessed: 07.11.2018.
URL: *https://graspit-simulator.github.io/build/html/index.html*

Murray, R. M., Sastry, S. S. and Zexiang, L. (1994), *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Inc.

Russell, S. and Norvig, P. (2016), *Artificial intelligence, A modern approach*, Prentice hall series in artificial intelligence, third edn, Pearson, Essex, England.

Saut, J.-P. and Sidobre, D. (2012), 'Efficient models for grasp planning with a multi-fingered hand', *Robotics and Autonomous Systems* **60**(3), 347–357.

Spong, M. W., Hutchinson, S. and Vidyasagar, M. (2006), *Robot modeling and control*, John Wiley & Sons Inc.

Universal Robots (2018), 'UR5, technical specifications'. Technical specification of the UR5 with photograph.
URL: *https://www.universal-robots.com/products/ur5-robot/*

Ying, Z. and Iyengar, S. S. (1995), 'Robot reachability problem: A nonlinear optimization approach', *Journal of Intelligent & Robotic Systems* **12**(1), 87–100.

Zacharias, F., Borst, C. and Hirzinger, G. (2007), Capturing robot workspace structure: representing robot capabilities, *in* '2007 IEEE/RSJ International Conference on Intelligent Robots and Systems', pp. 3229–3236.

Zacharias, F., Borst, C. and Hirzinger, G. (2009), Online generation of reachable grasps for dexterous manipulation using a representation of the reachable workspace, *in* '2009 International Conference on Advanced Robotics', pp. 1–8.