

# Robotic Bin-Picking under Geometric End-Effector Constraints: Bin Placement and Grasp Selection

Irja Gravdahl  
Dept. of Engineering Cybernetics  
Norwegian U. of Science and Technology  
Trondheim, Norway  
irja.gravdahl@ntnu.no

Katrine Seel  
SINTEF Digital  
Trondheim, Norway  
katrine.seel@sintef.no

Esten Ingar Grøtli  
SINTEF Digital  
Trondheim, Norway  
esten.ingar.grotli@sintef.no

**Abstract**—In this paper we demonstrate how *path reachability* can be taken into account when selecting among predetermined grasps in a bin-picking application, where grasps are supplied independently of the robot at hand. We do this by creating a map of the workspace to optimally place the bin with regards to the existence of an inverse kinematic solution and a collision-free path, a necessary condition for systems with obstructions in the workspace. Furthermore, we densely re-map this region and based on this map predict whether a grasp is reachable by the robot. Moreover, an algorithm is implemented to weight the grasps in terms of path existence, length and time consumption. The algorithm was tested with grasps generated by the neural network in simulation and the results indicate that faster picking can be achieved when taking path reachability into consideration.

**Index Terms**—bin-picking, robot kinematics, grasping

## I. INTRODUCTION

Bin-picking is a concoction of 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. Following this reasoning, much research has been done on one of two things; finding high quality grasp candidates based on the geometry and pose of the objects to be picked, and finding good trajectories to reach a pose associated with the grasp. However, this dividing strategy can be problematic: it may not be feasible to reach the pose for the robot due to constraints in the workspace [1].

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, e.g.: [2]. A grasp may be deemed good through the appropriate metrics, for example the force closure property [3], but must be rejected if the robot cannot reach it [1]. For a given grasp, the existence of an Inverse Kinematic (IK) solution is sufficient, if the workspace is otherwise clear of obstructions [4]. However, for most practical implementations, both arm kinematics and reachability considerations are necessary.

This work is funded by the Research Council of Norway through the project "Dynamic Robot Interaction and Motion Compensation" (contract number 270941) and the centre for research based innovation "SFI Manufacturing" (contract number 237900).

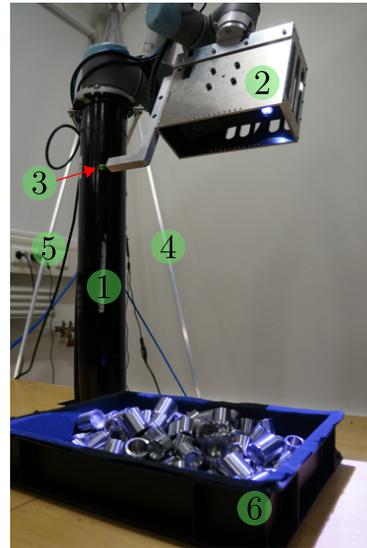


Fig. 1: Photo of the bin-picking set-up at SINTEF Digital, Trondheim. Notice the pedestal (1), camera housing (2), picker (3), fastening mechanism (4)(5), and bin with reflective steel parts (6).

As an example, consider the system seen in Fig. 1, which has additional constraints in terms of an eye-in-hand vision system, is placed upon a pedestal, and the bin it is to pick from constitutes an additional obstruction in the workspace. These constraints influence the robot's ability to move freely in its workspace. An inverse kinematic solution is hence only necessary, and the existence of a collision-free path to the appropriate pose must be ensured, for the grasp pose to be reachable for the robot.

The use of a grasp planner (e.g.: [5], [6] and [7]) is a popular choice to generate grasps for picking. The grasps chosen for picking in this set-up are supplied by a dual-resolution convolutional neural network trained on simulated data [8]. The input to the network is a point cloud of the current distribution of parts in the bin, and the output is multiple grasp pairs,  $\{\mathbf{d}_i, \mathbf{v}_i\}$ , where  $i \in \{1, \dots, N\}$ ,  $N \in \mathbb{N}$ ,  $\mathbf{d}_i \in \mathbb{R}^3$  is a point and  $\mathbf{v}_i \in \mathbb{R}^3$  is an approach vector. The network supplies four lists of grasp pairs, one for each quadrant in the bin. A grasp is said to be valid if there are no local collisions with other objects in the bin. The output pairs from the network are ordered based on their closeness to the world  $z$ -axis and in the

direction of the camera frame. The motivation behind focusing solely on grasp planning in the neural network, is that it can be robot agnostic. The implementation to include reachability considerations to re-arrange the output pairs in this paper is designed as a separate module to sustain the modular nature of the bin-picking system.

In this paper, we show that by mapping the workspace of the robot manipulator arm, positioning the bin based on this mapping, and thoroughly re-mapping this space with potential grasps offline can increase picking success by placing the bin in a more accessible region, as well as prioritizing grasps corresponding to short path lengths and time consumption. Using this mapping data in conjunction with the output from the neural network to connect these subsystems, we can re-arrange the preferred output from the neural network in terms of the robots ability to pick the objects, prioritizing grasps reachable by the robot. Mapping in this context refers to creating a map of the robot abilities. Due to the additional constraints on the system imposed by the pedestal, the camera housing, and potential collisions with the bin, in addition to an IK solution existing, a collision-free motion plan must exist for a pose to be reachable.

The contributions of this work include utilizing the combined result of IK solutions and motion-plan existence in the workspace to place the bin optimally. Optimally in this context refers to the region of the workspace with the highest concentration of IK solutions and motion-plans. Furthermore, in this optimal region, the planner LKBPIECE1 [9] (Lazy Bi-directional KPIECE with one level of discretization) with optimization objective *path length* from the Open Motion Planning Library (OMPL) [10] was investigated in terms of several metrics; path existence, path length, planning time and execution time. Moreover, a method for introducing the aforementioned metrics of the robot into the grasp selection process, without the need for explicitly querying an IK solver, path-planner or collision-checker is described.

Building on the work of [1], [11] and [12] where the existence of an IK solution is used as a criterion when selecting a grasp, we propose in this paper also to include existence of a collision-free path in the grasp selection process. This collision checking is particularly useful when considering geometric constraints, exemplified by the large volume of the 3D sensor in the system at hand.

The rest of this paper is organized in the following way; in section II, previous research on combined grasp- and path-planning will be discussed. Section III will formulate the problem to be solved, and section IV will deal with the methods used, and a discussion of the results. Lastly, the paper is concluded and future work is discussed.

## II. PREVIOUS RESEARCH

To the extent of the authors' knowledge, there exists some ambiguity on the use of the term reachability. In [13], they define the reachability of a robot as "*its ability to move its joints and links in free space in order for its hand to reach the given target*", indicating the term involving some movement

from one state to another. However, for example in [11], [12] and [14] it refers to the existence of an inverse kinematic solution only. In the rest of this paper, reachability will refer to the existence of an IK solution, and the term *path reachability* will refer to instances where an IK solution and a collision-free path exists.

Due to the presence of the pedestal and the camera housing, collisions in the workspace can occur. A pose in the workspace that is reachable might not be path reachable due to the geometry of the camera housing which could collide with the robot or the pedestal during the traversal of a path.

When combining motion-planning and grasp planning, there exists a substantial amount of research that either implements grasp planning in motion planners or robot capabilities in grasp planners, as a way to include reachability.

In [14], information on the robot kinematics, the local environment of the object to be grasped, and the force-closure property of the grasp is encoded in a *grasp-scoring function*, which is used to rank a precomputed set of grasps.

The use of offline generated "capability maps" for manipulators, a term used by [11] and applied to improve grasp planning in [12], is useful when incorporating robot kinematics with grasp planning. The capability map contains information about the reachability of the robot and aids in predicting if a grasp is reachable. In addition, [11] include directional preferences in the map, so that information on appropriate approach directions can be incorporated. When capturing the workspace structure in this map, the whole workspace of the robot was discretized, and sampled to obtain a uniform distribution of possible Tool Centre Point (TCP) configurations. For each of these TCP configurations, IK calculations were done, and if a solution existed, the point was marked reachable. This procedure results in a representation containing the probability of a grasp being reachable by the robot. By feeding a grasp planner this capability map, the grasp planner is decoupled from explicit implementation of the robot kinematics, but information on the success probability will be available such that unreachable grasps can be discarded early.

An integrated planner, Grasp-RRT, is presented in [16], combining the search for a valid trajectory, a feasible grasp and an inverse kinematic solution, the central elements of grasping. The method does not rely on precomputed grasps like in [11] and [12], but finds feasible grasps whilst planning a path for the robot.

In [1], a framework for workspace aware online grasp planning is provided. By using a precomputed representation of the reachable workspace called the *reachability space* where potential TCP poses are queried, they use this to bias the robot towards more reachable regions of the workspace. Planning for a grasp is only done in these more accessible regions, limiting the time spent searching for grasps in less reachable regions.

## III. PROBLEM FORMULATION

The bin-picking system set-up used in this paper comprises a UR5 robotic manipulator arm, a Zivid 3D camera and a vacuum gripper to pick reflective parts from a bin. To supply

a grasp, it is important to obtain sufficient depth information from the images of the distribution of parts in the bin. An eye-in-hand configuration provides flexibility in this regard. Information from the camera is used to compute multiple grasps based on how objects are placed in the bin. When the sensor is attached to the robotic arm performing the grasping, additional constraints on how the manipulator can move whilst avoiding self-collisions and collisions with the bin or other parts of the environment is imposed. A characteristic of the grasps supplied by the network [8], is that they are decoupled from the robot tasked with reaching them. The network has no knowledge about the existence, and kinematics, of the robot. This raises the issue of reachability, and the need for coupling these two aspects; optimal grasp generation in terms of the object geometry, and prioritizing grasps that are reachable for the robot.

The issue at hand is determining with what amount of ease the robot can reach a specific pose in the workspace, and to find a way to judge which grasps are favourable for reaching with the robotic manipulator arm. The following sections detail the results obtained attempting a solution to this problem. The following results were obtained in simulation, using a simulator supplied by Universal Robotics [17], and upon it, working with the Robot Operating System (ROS) workspace structure for the physical system.

#### IV. METHODS AND RESULTS

##### A. Current placement of bin in workspace

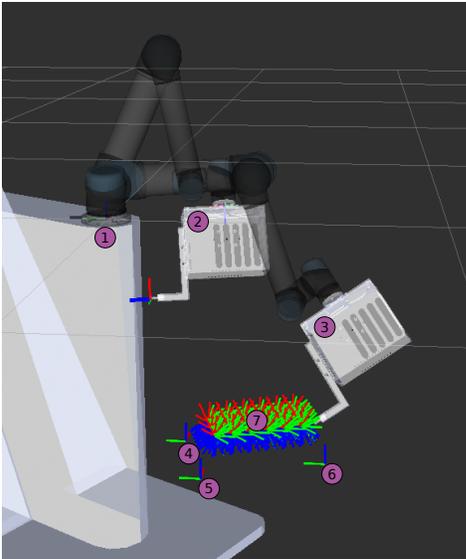


Fig. 2: Visualization of current set-up as seen in RViz. 1) pedestal where the robot is placed, 2) robot pose at the scan configuration and Zivid 3D camera, 3) example path from scan configuration to an experimental grasp, 4,5,6) actual placement of the bin corners in the physical setup, 7) experimentally generated grasps for investigation in the current bin area.

The current placement of the bin is based on the optimal range of the Zivid 3D camera, which is 50-60cm from the objects. This 3D camera is placed within a camera housing

of substantial size (marked 2 and 3 in Fig. 2) which further limits the arm configuration space. Due to this demand, the UR5 was placed upon a pedestal, see Fig. 2. Since the output of the neural network is a point and an approach vector only, a change in joint configurations whilst keeping the TCP stationary at the point, may lead to multiple viable solutions. As a result of this characteristic, several coordinate frames were sampled with the origin at the same point, but with different orientations. It is worth noting, that since additional constraints on the system in terms of the pedestal and the camera housing were present, the need for finding a collision-free path was detrimental.

##### B. Expanding possible regions for bin placement

First, the level in the workspace at which the reach of the robot was the greatest, constrained by the geometry of the workspace and itself had to be identified. Utilizing RViz as a visualization tool, a larger portion of the workspace was sampled and tested. It was vital that the sampling was broad enough in all directions to capture also the limits of the workspace. Viewing Fig. 3, the portion of the workspace that was investigated can be seen. The workspace part of interest was divided into two cubes of equal size, on each side of the robot base. The rest of the workspace was deemed less accessible due to the layout of the robot cell, and therefore unnecessary to map. The triangular structure at the back of the robot for example, is physical.

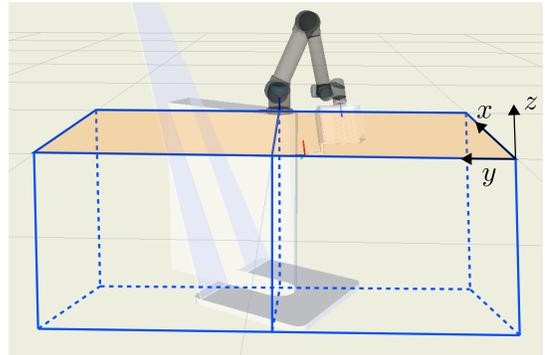


Fig. 3: Outline of the regions of the workspace being mapped. For implementation reasons the mapped region was divided in two.

The two cubes seen in Fig. 3 were discretized, and an  $11 \times 6$  grid in the  $xy$ -direction, with an increment of 0.1 in the  $z$ -direction, was created. MoveIt and OMPL [10] were used to plan a path directly to the sampled pose, succeeding only if there existed an IK solution and a collision-free path. The right cube was mapped first, and then the left.

In each  $xy$ -plane, a 2D heatmap was produced to visualize the results. Maps such as the one in Fig. 4 were created for all the increments in the  $z$ -direction to also reveal optimal height in terms of path reachability. The path reachability coverage from both cubes in Fig. 3 were combined, and the optimal level in terms of best average path reachability was revealed to be the level flush with the base at  $z = 0$ , seen in Fig. 3 as the opaque orange level. The level map at  $z = 0$  can be

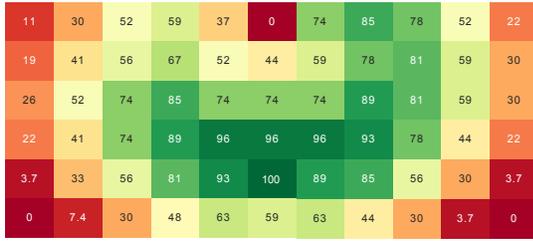


Fig. 4: Path reachability heatmap seen from above: Optimal level based on average hit rate along with a concentrated area with high reachability. The numbers in the squares represent percentage path reachability for the points and the colors reflect this scale from red (low) to green (high)

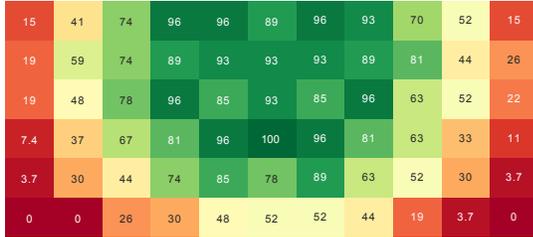


Fig. 5: Reachability heatmap seen from above: Coverage of IK solutions in the top level of the workspace. The numbers in the squares represent percentage IK solutions for the points and the colors reflect this scale from red (low) to green (high).

seen in Fig. 4. This result was not unanticipated as it is the level closest to the centre of the workspace, the base. It is also the region of the theoretical workspace with the largest span. Even though it was likely the best option, due to the imposed constraints of the pedestal and the camera housing, it had to be validated. This level had an average path reachability of 53.53%, but as can be seen from Fig. 4, there is a distinct area where the path reachability is very high. In addition to being the level with the highest average path reachability, the dark green "patch" in the middle was also the region with the highest concentration of successes when comparing this area with the other level maps in the  $z$ -direction. For comparison, the heatmap portraying only the IK solutions at this same level is shown in Fig. 5. It is clear that the path reachability heatmap is a more conservative estimate of accessible regions in the workspace.

Note, that the representation of the optimal level as can be seen in Fig. 4 is directionless and does not take into account which approach directions are more favourable for the robot. The number in the squares are path reachability indices calculated by the following formula [11];

$$n_{ij} = \frac{r_{ij}}{N_{ij}} \cdot 100 \quad i, j = 1, 2, \dots, 11 \quad (1)$$

where  $n_{ij}$  is the total path reachability score in percent for a point represented by a square in the level map,  $r_{ij}$  is the number of path reachable TCP frames in the point, and  $N_{ij}$  is the total number of possible TCP test frames in said point.

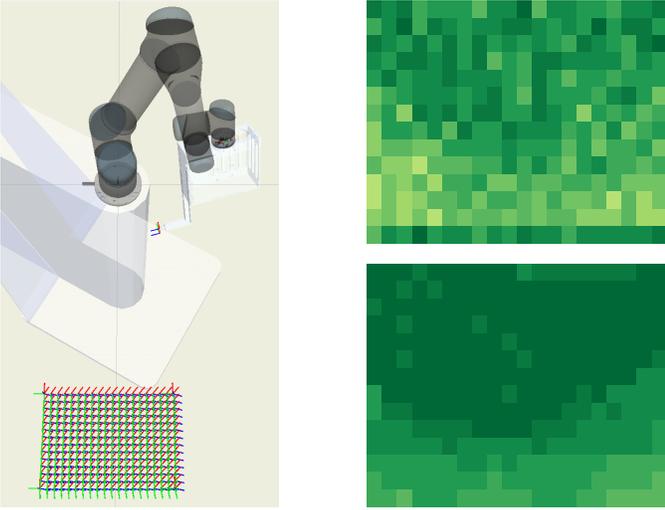
### C. New bin placement and mapping

When placing the bin based on the workspace mapping on path reachability, the bin was placed such that the point with 100% path reachability in the middle of Fig. 4 was roughly centered in the bin. As per this point, the bin was optimally placed in both  $x$ -,  $y$ -, and  $z$ -direction, in terms of maximizing path reachability.

After positioning the bin, it was desirable to thoroughly map this area in the same manner as the workspace was mapped, and use information on potential TCP placements as a compliment to the grasps supplied by the neural network to incorporate the robot kinematics and its abilities in the grasp selection process. Since the bin and its immediate surroundings is the main region of the workspace the robot will operate in, this can be viewed as a task-specific workspace mapping. The objective was to use known, offline gathered data such as path lengths and computing times, to constitute an additional step in the grasp selection process by being able to guide the grasps from the network,  $\{\mathbf{d}_i, \mathbf{v}_i\}$ , towards easy to reach regions of the bin.

When mapping this space, see Fig. 6a, several metrics were of interest to collect; the homogeneous transformation of the test grasp,  $\mathbf{T}_j^{\text{test}}$ , consisting of the rotation matrix of the test grasps given in the base frame,  $\mathbf{R}_j^{\text{test}}$  and the position vector of the test grasps in the base frame  $\mathbf{d}_j^{\text{test}}$ , the length of the calculated path to the test grasp, the planning time needed by the planner and the execution time of the path. Provided a grasp from the neural network, we could find the test grasp that most resembles the true grasp, and then be able to return information on path length and time. Take for instance the network outputting 50 grasps in order ranked on the quality of the grasp. We then wish to rearrange this ranked list based on similar test grasps and their attributes. Note, that for both the workspace mapping (volume shown in Fig. 3) and for the mapping of the new bin placement, it was necessary to attempt a plan several times to saturate the results [18]. This is due to the randomized nature of sampling-based motion-planners, which if given enough time will find a path if one exists, but is unable to return information on the existence of a path [19]. Tests were repeated 5 times, as this seemed a sufficient number to get a representative average, while at the same time not being too time-consuming. The effect of testing once for a path, and testing five time for a path can be seen in Fig. 6b, where the average path reachability goes from 86.93% to 95.70%.

The sampling in the bin is based upon the size of the objects to be picked and their size compared to the bin. The bottom area of the bin is  $26 \times 36$ cm and the radius of the cylinder objects is 3.2cm. In a worst case situation with regards to surface area available for grasping, the objects will stand upright, and approximately  $7 \times 10$  objects would fit on the borders. Allowing for two samples per part, there are  $14 \times 20$  points sampled evenly in the bin. In each of these points, there are 27 different orientations to ensure sufficient exploration of the possibility of another joint configuration providing a



(a) New placement of bin in the workspace. In each point sampled in the bin, one orientation of a test grasp is shown here, illustrated by one coordinate system in each point.

(b) Comparison between one planning instance and five instances, saturating the results. Top: one planning instance. Bottom: five planning instances.

Fig. 6: New bin placement and path reachability in the area

solution. This brings the number of total test grasps to 7560, for which it is possible to compare neural network grasps with, and conclude on the robots ability. In Fig. 6a the new placement of the bin is shown, along with one orientation of test grasps.

After a map of the bin space containing information on metrics of interest for each test grasp was saved in a look-up table, an algorithm which composes the path reachability test had to be implemented:

- Given the grasps from the neural network, the test grasp resembling it the most is to be identified.
- The path length, planning- and execution time for this corresponding test grasp is then looked up in the table.
- A cost function is evaluated based on the attributes of the test grasp for each grasp from the network.
- The grasps from the network are re-arranged based on this cost and returned to the picking loop.

The first step to identifying this test grasp is a check of the distance between the network grasp point and all test grasp points in the data set, and choosing the test grasp closest. So, for each grasp from the network and all the test grasps, we calculate the Euclidean distance  $\phi_1 : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}_{\geq 0}$ :

$$\phi_1(\mathbf{d}_i^{\text{nn}}, \mathbf{d}_j^{\text{test}}) = |\mathbf{d}_i^{\text{nn}} - \mathbf{d}_j^{\text{test}}|, \quad (2)$$

where  $\mathbf{d}_i^{\text{nn}}$  is the position vector of the grasp from the neural network and  $\mathbf{d}_j^{\text{test}}$  is the  $j^{\text{th}}$  test grasp position vector. After the point closest resembling the neural network grasp point is found, there are 27 options for rotation, where a comparison of the 3D rotations must be done to identify the approach angle closest in magnitude. Based upon the work of [20], the

following metric,  $\phi_2 : SO(3) \times SO(3) \rightarrow \mathbb{R}_{\geq 0}$ , is applied to all test grasps and compared to the neural network grasps,

$$\phi_2(\mathbf{R}_i^{\text{nn}}, \mathbf{R}_j^{\text{test}}) = \left\| \log(\mathbf{R}_i^{\text{nn}} (\mathbf{R}_j^{\text{test}})^{\top}) \right\|, \quad (3)$$

where  $SO(3)$  is the special orthogonal group of order 3,  $\mathbb{R}_{\geq 0}$  is the set of non-negative real numbers,  $\log(\cdot)$  is the matrix logarithm,  $\mathbf{R}_j^{\text{test}}$  is the rotation part of the transformation matrix describing the test grasp  $j$ , and  $\mathbf{R}_i^{\text{nn}}$  is the same for the neural network grasps. The  $\|\cdot\|$  (2-norm), gives the magnitude of the rotation angle [20]. The metric returns values in the interval  $[0, \pi)$ , where the objective is to find the smallest geodesic distance between the neural network grasp and test grasps. The pseudo code for the implemented algorithm is outlined in Algorithm 1.

---

**Algorithm 1:** Algorithm for re-arranging grasps to account for robot abilities

---

**Input:** Robot capabilities in terms of test grasps and list of grasps from neural network

**Output:** Re-arranged list of grasps in terms of path reachability

```

for all grasp from neural network  $\mathbf{T}_i^{\text{nn}}$  do
  for all test grasps  $\mathbf{T}_j^{\text{test}}$  do
    | calculate  $\phi_1(\mathbf{d}_i^{\text{nn}}, \mathbf{d}_j^{\text{test}})$  and  $\phi_2(\mathbf{R}_i^{\text{nn}}, \mathbf{R}_j^{\text{test}})$ ;
  end
  | return the closest test grasp in  $\phi_1$  and  $\phi_2$ ;
end
for each test grasp corresp. to a network grasp do
  | if the grasp is path reachable then
    | | calculate cost of each grasp;
    | |  $J_i = \alpha\phi_3 + \beta\phi_4 + \delta\phi_5$ ;
  | else
    | |  $J_i = \infty$ ;
  | end
end
  | sort grasps with respect to  $J$ ;

```

---

The algorithm ensures that path reachable grasps with low computation times and short paths are prioritized over less ideal grasps, and allows the highest rated grasp to be the one best for the robot. Information on IK, path-existence, collision-checking and time consumption is implicitly included in the cost function  $J$  as it is calculated offline, and is accessible faster than the time it would require to do the operations sequentially for each grasp from the network.

Viewing the algorithm,  $\phi_i$ ,  $i \in [3, 4, 5]$  refers to the normalized average values of path length, planning time and execution time of the path to the test grasps.  $\alpha$ ,  $\beta$  and  $\delta$  are weighting parameters enabling prioritization of characteristics that are more important than others.

#### D. Testing

The grasps from the neural network are given relative to the current placement of the bin, the placement seen in Fig. 2. Transforming the grasps to the new bin, as shown in Fig. 6a,

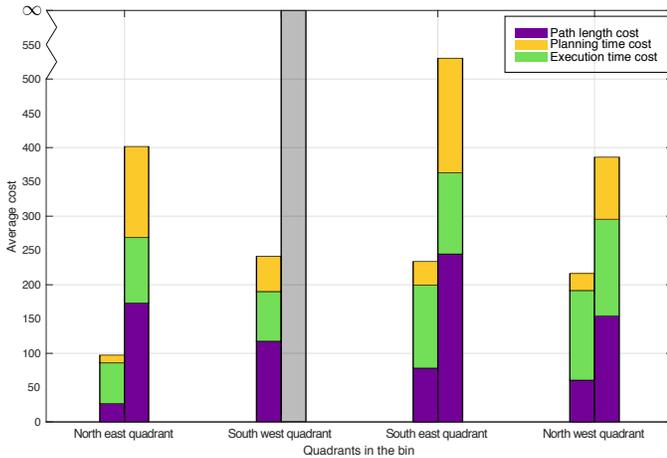


Fig. 7: Comparison of the average cost of the first 10 grasps from the neural network (column to the right) and the first 10 grasps from the re-arranged list based on path reachability (column to the left). The average cost is split into the average sub costs of path length, planning time and execution time for the grasps.

enables testing of the algorithm. To evaluate the functionality of the algorithm, all weights,  $\alpha$ ,  $\beta$  and  $\delta$  were set to 1 such that path length, planning time and execution time of the paths were considered equally important, since these three aspect influence how fast a full picking cycle is.

A comparison between the list of grasps from the neural network and the re-arranged list of grasps when considering path reachability, had to be investigated. To do so, the first 10 grasps from the network, unsorted on path reachability, and the first 10 grasps from the re-arranged list were recorded. The average cost  $J = \alpha\phi_3 + \beta\phi_4 + \delta\phi_5$  of these groups was calculated and plotted in a bar chart, seen in Fig. 7. There is one plot for each quadrant in the bin.

Viewing the figure, the columns to the right in each group represent the average cost of the first 10 grasps from the network, not considering path reachability. The columns to the left in each group represent the average cost of the first 10 grasps of the sorted list. The value of 10 was chosen to obtain a representative average when comparing the cost  $J$ . When comparing the two, it can be seen that when including path reachability, grasps with a short path length, planning time and execution time, are prioritized. Note that the y-axis goes to infinity since some of the grasps preferred by the network were unreachable by the robot in the south west quadrant of the bin, this is illustrated in Fig 7 by the grey column. When comparing the results for the south-east and south-west regions in the bin in terms of a higher cost, this is consistent with the slightly lower level of path reachability in these regions as can be seen in Fig 6b.

The time the algorithm requires to process and sort the grasps from the neural network and the map in one quadrant is under a quarter of a second, and in theory it should be able to process all four quadrants in under a second to account for path reachability. However, this must be tested on the physical set-up in real-time [21]. The results indicate that faster picking can be achieved when including path reachability as a

separate module in the bin-picking system, whilst still keeping the neural network focused on grasp planning alone.

## V. CONCLUSIONS

In this work, a mapping of the workspace of the system shown in Fig. 1 has been undertaken. Based on this mapping, the region of the workspace with the highest *path reachability* was chosen as the best location for placing the bin, when considering that a robot needs to reach the objects to be picked. Due to the constraints on the end-effector, and the constraints in the robot cell such as the pedestal and the bin, collision-free paths had to be found and considered. After placing the bin, an algorithm for predicting whether or not a path could be found to the grasp pose was implemented on the system, separate from the neural network which supplies the grasps. The algorithm evaluates a cost function including metrics such as path length, which is used to skew the grasps chosen for picking towards grasps reachable by the robot, and where a path exists. The algorithm was tested on grasps generated by the neural network in simulation and the results indicate that faster picking can be achieved when taking path reachability into consideration.

In the future, the algorithm should be tested on the physical system in depth to investigate if the picking time is influenced considerably by the extra check on the path reachability. In addition, a new robot configuration to capture depth images from should be found and considered when creating the path reachability map, if re-positioning of the bin is undertaken. Furthermore, it would be of interest to also include in the cost function, how well a grasp's likeness is to the test grasp it is coupled with in the algorithm.

## REFERENCES

- [1] I. Akinola, J. Varley, B. Chen, and P. K. Allen, "Workspace aware online grasp planning," In IROS, 2018, pp. 2917-2924.
- [2] D. Kraft, L.-P. Ellekilde, and J. A. Jørgensen, "Automatic Grasp Generation and Improvement for Industrial Bin-Picking," In *Gearing up and accelerating cross-fertilization between academic and industrial robotics research in Europe*, Cham, 2014, pp. 155-176, Springer.
- [3] A. Bicchi, "On the Closure-Properties of Robotic Grasping," *Int. Journal of Robotics Research*, 14(4), Aug., pp. 319-334, 1995.
- [4] J. P. Saut and D. Sidobre, "Efficient models for grasp planning with a multi-fingered hand," *Robotics and Autonomous Systems*, 60(3), Mar., pp. 347-357, 2012.
- [5] A. T. Miller and P. K. Allen, "Grasplit! A versatile simulator for robotic grasping," *IEEE Robotics & Aut. Mag.*, 11(4), pp. 110-122, 2004.
- [6] R. Diankov and J. Kuffner, "OpenRAVE: A Planning Architecture for Autonomous Robotics," Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 152132008, vol. 79.
- [7] B. León et al., "OpenGRASP: A Toolkit for Robot Grasping Simulation," In *Simulation, Modeling, and Programming for Autonomous Robots*, 2010, pp. 109-120.
- [8] J. S. Dyrstad, M. Bakken, E. I. Grøtli, H. Schulerud, and J. R. Mathiassen, "Bin Picking of Reflective Steel Parts Using a Dual-Resolution Convolutional Neural Network Trained in a Simulated Environment," In *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2018, pp. 530-537.
- [9] I. A. Şucan and L. E. Kavraki, "Kinodynamic Motion Planning by Interior-Exterior Cell Exploration," In *Algorithmic Foundation of Robotics VIII: Selected Contributions of the Eight International Workshop on the Algorithmic Foundations of Robotics*, G. S. Chirikjian, H. Choset, M. Morales, and T. Murphey, Eds. 2010, pp. 449-464.

- [10] Ioan A. Şucan, Mark Moll, Lydia E. Kavraki, The Open Motion Planning Library, IEEE Robotics & Automation Magazine, 19(4):72–82, December 2012. <http://ompl.kavrakilab.org>
- [11] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2007, pp. 3229-3236.
- [12] F. Zacharias, C. Borst, and G. Hirzinger, "Online generation of reachable grasps for dexterous manipulation using a representation of the reachable workspace," In Int. Conf. on Adv. Robotics, 2009, pp. 1-8.
- [13] Z. Y. Ying and S. S. Iyengar, "Robot Reachability Problem - a Nonlinear Optimization Approach," *Journal of Intelligent & Robotic Systems*, vol. 12, no. 1, pp. 87-100, 1995.
- [14] D. Berenson, R. Diankov, N. Koichi, K. Satoshi, and J. Kuffner, "Grasp planning in complex scenes," In 7th IEEE-RAS International Conference on Humanoid Robots, 2007, pp. 42-48.
- [15] J. Fontanals, B. Dang-Vu, O. Porges, J. Rosell, and M. A. Roa, "Integrated grasp and motion planning using independent contact regions," In Int. Conf. on Humanoid Robots, 2014, pp. 887-893.
- [16] N. Vahrenkamp, M. Do, T. Asfour, and R. Dillmann, "Integrated Grasp and motion planning," In IEEE International Conference on Robotics and Automation, 2010, pp. 2883-2888.
- [17] <https://www.universal-robots.com/>. [Accessed: 20.05.19]
- [18] J. Meijer, Q. Lei, and M. Wisse, "Performance study of single-query motion planning for grasp execution using various manipulators," In 18th Int. Conf. on Advanced Robotics (ICAR), 2017, pp. 450-457.
- [19] L. E. Kavraki and S. M. LaValle, "Motion Planning," in Springer Handbook of Robotics, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 109-131.
- [20] D. Q. Huynh, "Metrics for 3D Rotations: Comparison and Analysis," *J. of Mathematical Imaging and Vision*, 35(2), pp. 155-164, 2009.
- [21] I. Gravdahl, "Grasp selection in bin picking tasks for robotic manipulator arm with end-effector geometric constraints", MSc thesis, Norwegian University of Science and Technology, 2019.