Robotics and Computer Integrated Manufacturing 67 (2021) 102032

Contents lists available at ScienceDirect

Robotics and Computer Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Full length Article

Reconfigurable Grasp Planning Pipeline with Grasp Synthesis and Selection Applied to Picking Operations in Aerospace Factories[☆]

João Pedro Carvalho de Souza[*,a,b,c], Carlos M. Costa[a,b], Luís F. Rocha[b], Rafael Arrais[a,b], A. Paulo Moreira[a,b], E.J. Solteiro Pires[b,c], José Boaventura-Cunha[b,c]

[a] Faculty of Engineering of University of Porto (FEUP), Porto, Portugal
[b] Institute for Systems and Computer Engineering, Technology and Science, Porto, Portugal
[c] Engineering Department, University of Trás-os-Montes and Alto Douro, Vila Real, Portugal

ARTICLE INFO

Keywords:
Grasping Pipeline
Robot Pick and Place
Manufacturing Robot Processes

ABSTRACT

Several approaches with interesting results have been proposed over the years for robot grasp planning. However, the industry suffers from the lack of an intuitive and reliable system able to automatically estimate grasp poses while also allowing the integration of grasp information from the accumulated knowledge of the end user. In the presented paper it is proposed a non-object-agnostic grasping pipeline motivated by picking use cases from the aerospace industry. The planning system extends the functionality of the simulated annealing optimization algorithm for allowing its application within an industrial use case. Therefore, this paper addresses the first step of the design of a reconfigurable and modular grasping pipeline. The key idea is the creation of an intuitive and functional grasping framework for being used by factory floor operators according to the task demands. This software pipeline is capable of generating grasp solutions in an offline phase, and later on, in the robot operation phase, can choose the best grasp pose by taking into consideration a set of heuristics that try to achieve a successful grasp while also requiring the least effort for the robotic arm. The results are presented in a simulated and a real factory environment, relying on a mobile platform developed for intralogistic tasks. With this architecture, new state-of-art methodologies can be integrated in the future for growing the grasping pipeline and make it more robust and applicable to a wider range of use cases.

1. Introduction

Planning and performing a grasp movement is done effortlessly by humans, but for robots, this is a significant challenge. The current established industrial solutions are only capable of dealing with this problem in well-structured and controlled environments. Typically, these solutions resort to techniques that depend on the operators' expertise, which manually programs the robotic system, or are based on inflexible, application oriented, software tools (e.g., drive through, lead through and offline programming), which do not convey with modern industry paradigms that ultimately seek for new autonomous and efficient techniques to enhance the flexibility of industrial robotic systems.

For decades the study of grasp techniques in complex scenarios has been explored by the scientific community, which led to the appearance of several analytical [1–6] and data-driven [7] approaches aiming for the improvement of production lines, logistics processes, assembling operations, and bin-picking tasks. Despite these significant contributions, the complexity associated with designing a task-oriented analytical method or build a large dataset, required for the training of Machine Learning (ML) systems, limits the effective adoption of these technologies as an efficient, user-friendly, and applicable industrial solution.

In this context, this paper introduces a reconfigurable robot grasping software pipeline. It is based on a sequential architecture to autonomously compute the grasp solution for a robotic arm in an industrial application. This pipeline is built on top of Robot Operating System (ROS) and "GraspIt!" simulator [8], extending the applicability of Simulated Annealing (SA) [9] with a feasible application time.

[☆] This work is co-financed by the following agencies: Portuguese funding agency – FCT, Fundação para a Ciência e a Tecnologia – project UIDB/50014/2020; ERDF – European Regional Development Fund – through the Operational Programme for Competitiveness and Internationalisation – COMPETE 2020; Lisboa2020 under the PORTUGAL 2020 Partnership Agreement, and through the Portuguese National Innovation Agency – ANI – project PRODUTECH SIF: POCI-01-0247-FEDER-024541; European Union's Horizon 2020 – The EU Framework Programme for Research and Innovation 2014–2020 – under grant agreement no 777096.

[*] Corresponding author.
E-mail addresses: jsouza@fe.up.pt, joao.p.souza@inesctec.pt (J.P. Carvalho de Souza).

https://doi.org/10.1016/j.rcim.2020.102032
Received 9 July 2020; Accepted 9 July 2020

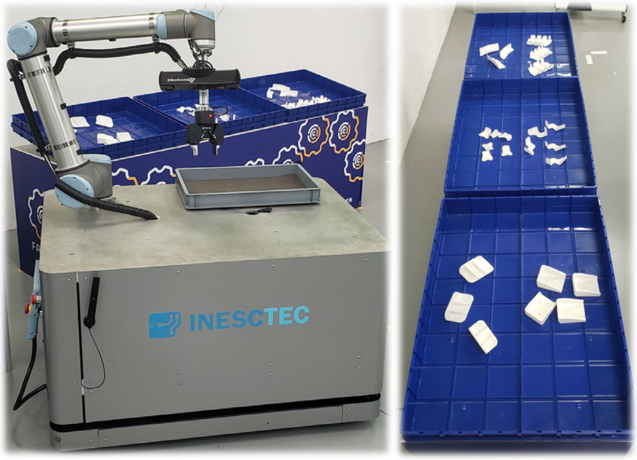

Fig. 1. Omnidirectional mobile manipulator (left) and bins with objects (right).

With this work the authors goal is to deploy to both the industrial and scientific community a software tool capable of automatically generating robot grasp poses over a set of objects. Namely, this paper presents the backbone of the proposed modular and configurable pipeline, where methodologies and tools already consolidated in the scientific community will be further integrated. Furthermore, this tool will serve as the basis for future developments on the robot grasping topic.

Practical results are presented considering a real aerospace factory use case, targeting the execution of intralogistic operations by an omnidirectional robot equipped with a robotic arm, i.e., a mobile manipulator (Fig. 1).

Bearing these ideas in mind, this paper is structured as follows: Section 2 discusses the related work. Section 3 presents relevant background on robot based grasp topic. Section 4, presents the proposed grasp planning software backbone. Finally, in Section 5 the experimental results are presented and discussed, followed by the Conclusions and Future Work (Section 6).

2. Related Work

The robotic grasp was firstly investigated by works such as [1–6]. Typically, they explore the stability of multi-fingered grasps considering closure conditions in wrench space analyses. These approaches demonstrated that the computation of valid grasping poses can be complex according to the task demands and mathematical modeling practical assumptions, e.g., the number of fingers, friction or frictionless contacts, object-agnostic or not. Their formulation, however, allowed the research community to derive some definitions used even in ML methodologies. For example, the criteria formulation to define the grasping quality by Ferrari and Canny in [6] and the force and form-closure conditions of Nguyen [1,2].

Nowadays, and driven by the appearance of new processing technologies and computer techniques, a new category of robotic grasping systems has emerged: the experience-based. In this field, several authors investigate the integration of optimization techniques into analytical analyses [9,10] and ML methods to reach grasping results that can be summarized in supervised learning (structured labeled datasets [11–23] or Learning by Demonstration (LbD) [24–27]) and Reinforcement Learning (RL) [28,29].

Focused in object-agnostic grasp, Saxena et al. in [11] used visual object features in the learning task. Geometric properties of superquadrics shape objects and geometric properties associated with gripper structure were adopted by the authors of [13] and [12], respectively. These works demonstrate how complex is the feature extraction procedure during a ML algorithm supervised training, for logistic regression [11], Support Vector Machine (SVM) [13] or Neural Network (NN) [12].

In LbD [25–27] approaches, the feature modeling is not necessary. However, these techniques could limit the robot grasp efficiency since it is based on the teacher agent performance. Another problem is the high practical effort to build a learning dataset, as required in the RL methodologies [28,29].

The Deep Supervised Learning has gain attention from the researching community that, motivated by the impressive Convolutional Neural Network (CNN) results in object classification, investigate this approach into grasp policies. Works like [14–16], consider the approximation of the grasping detection as an object classification problem, achieving interesting grasp detection results. Yet, just a few of them evaluate the suggested approach in a real grasp problem, and in the instances where the authors do, a decrease in the approach performance is verified. Grasping, picking and handling objects by robots could be more complex considering cluttered environments [30], objects with curved [31] or flat [32] surfaces, flexible [33], and composite [34] materials. Mahler et al. [20–23] propose a Grasp Quality CNN (GQ-CNNs) to solve objects-agnostic grasping problem using suction and two-finger gripper. The author also study ambidextrous, and bin-picking polices. The algorithm was trained with a large dataset, called DexNet that include analytical modeling assumptions. Nevertheless, the authors encounter challenges in grasp flexible, porous objects, and with loose packing.

As shown in this section, the complexity of experience-based algorithms derives from: (i) the creation of the database, (ii) the modeling, and the interpretation of the graspable object. The database needs to be relevant to the task at hand, and the object representation must be defined as a 3D model or only with direct sensing data. Besides that, the gripper technology also needs to be evaluated. As related by Birglen et al. in [35], nowadays, several grippers structures are developed by companies led by the challenges of Industry 4.0. The possibilities for algorithms are vast, and their development must comply with the requirements of the application, the capability of generalization, fast decision, and present some tolerance to the inherent sensing errors.

All the related issues, from analytical to ML methods discussed in the previous section, difficult industries to have a suitable tool to generate grasp solutions according to application demands. Even though a completely generic solution is not achieved, the design of a reconfigurable and modular grasp pipeline, using already state-of-art methodologies could increase productivity in factory environments.

3. Background and Notation

Before presenting the proposed grasp planning pipeline, described in Section 4, this section will discuss some background and notation associated with the challenge at hand.

In this first version of the pipeline the authors assume that the object's shape is known from the beginning, and that only multi-fingered grippers are used. In this context, the following sections, 3.1 and 3.2, summarize the multi-fingered analytical formulation and the SA based grasp algorithm [9], respectively.

3.1. Multi-Fingered Grasp

A multi-fingered grasp is realized over a set of contacts between the active pairs (the workpiece and the gripper). Therefore, the determination of a suitable configuration of independent grasp points is the primary step of the fingered grasp planning.

The wrench vectors describe the forces and moments that influence a rigid body's dynamic. These vectors can be used to formulate grasp locations, and a wrench vector is presented below:

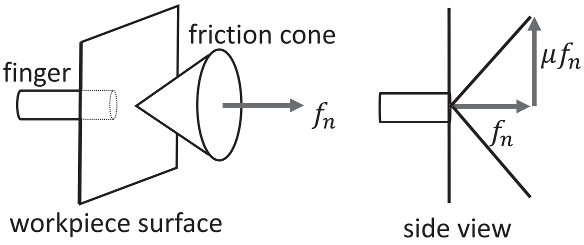

Fig. 2. Friction contact model and the geometric representation of Coulomb's law (figure based on [36]).

$$\mathbf{w_c} = \begin{bmatrix} \mathbf{F} \\ \boldsymbol{\tau} \end{bmatrix} \tag{1}$$

where **F** and $\boldsymbol{\tau}$ are the vector representations of the forces and the moments. The wrench vectors have 3 and 6 Degree of Freedoms (DOFs) in the case of IR^2 and IR^3, respectively.

The contact models can be categorized as frictionless contact, friction contact (also named hard finger contact), and soft contact [36]. The focus of this paper will be the friction contact, since this model is sufficient for the picking application addressed in this paper.

The friction contact model considers the mechanical interaction between the active pairs. Therefore, the wrench convex depends on the friction contact forces, described by Coulomb model of friction: Considering the normal force f_n, and the tangential force f_t, static friction occurs when there is no slipping between the two surfaces of contact, that is when $|f_t| \leq \mu_t f_n$ where μ is a positive value representing the static tangential coefficient of friction. Fig. 2 shows an example of hard finger contact, the geometric representation of the Coulomb's law and the friction cone convex also defined as FC_{c_i}.

A wrench representation, w.r.t the i-th contact point (c_i), is defined as follows:

$$W_{c_i} = \begin{bmatrix} 100 \\ 010 \\ 001 \\ 000 \\ 000 \\ 000 \end{bmatrix} \mathbf{f}_{c_i}, \quad \mathbf{f}_{c_i} \in FC_{c_i} \tag{2}$$

where $FC_{c_i} = \mathbf{f} \in R^3\colon \sqrt{f_x^2 + f_y^2} \leq \mu_t f_{z,} f_z \geq 0$, and μ_t is the transversal friction coefficient in c_i.

Therefore, it is possible to define the matrix that compose the wrench vector:

$$\mathbf{W}_{c_i} = \mathbf{B}_{c_i} \mathbf{f}_{c_i}, \quad \mathbf{f}_{c_i} \in FC_{c_i} \tag{3}$$

where B_{c_i} is the wrench basis matrix with dimension $p \times n$ where p is the DOFs and n the number of independent forces and moments that constitutes f_{c_i}. The contact model discussed here has as reference frame the one with the origin coincident with the contact point itself. It is more convenient to refer all contacts in a grasp model to a common frame, generally the center of mass of the work piece. Therefore the wrench transformation matrix is defined as follows:

$$^o\mathbf{T}w_{c_i} = \begin{bmatrix} ^o\mathbf{R}_{\mathrm{c_i}} & 0 \\ ^o\hat{\mathbf{t}}_{\mathrm{c_i}} {}^o\mathbf{R}_{\mathrm{c_i}} & ^o\mathbf{R}_{\mathrm{c_i}} \end{bmatrix} \in \mathrm{IR}^3 \tag{4}$$

where $^o\mathbf{R}_{c_i}$ and $^o\mathbf{t}_{c_i}$ are the rotation and translation matrix of the i-th contact point (c_i) w.r.t. object frame (o). The $\hat{\mathbf{t}}$ is the linear operator representing the cross product $^o\mathbf{t}_{c_i} \times {}^o\mathbf{R}_{c_i}$. Hence, the contact map $\mathbf{G}_i$ is defined as follows:

$$\mathbf{G}_i =^o \mathbf{Tw}'_{c_i} \mathbf{B}_{c_i} \tag{5}$$

Note that it describes the direction of each component of the i-th applied wrench and defines the constraints of the contact. The grasp map is the matrix with all contact maps that characterize the contact model (it is also named constraint matrix):

$$\mathbf{G} = \left[{}^o\mathbf{Tw}'_{c_1} \mathbf{B}_{c_1} \ \cdots \ {}^o\mathbf{Tw}'_{c_N} \mathbf{B}_{c_N} \right] \tag{6}$$

Then, including the magnitude of the forces, a workpiece wrench can be written:

$$^o\mathbf{W} = [\mathbf{G}_1, \ldots, \mathbf{G}_N][\mathbf{f}_{c1}, \ldots, \mathbf{f}_{cN}]' = \mathbf{GF} \tag{7}$$

where: $\mathbf{F} \in FC$ and $FC = FC_{c_1} \times \ldots \times FC_{C_N}$

The $^o\mathbf{W}$ also defines the GWS (grasp wrench space) of the grasp. It is obtained by means of the L_∞ or L_1 norm. The L_∞ defines the GWS ($\mathbf{W}_{L_\infty}$) considering the limitation of the maximum allowable normal contact force, while L_1 defines the GWS ($\mathbf{W}_{L_1}$) by the sum magnitude of the normal contact forces. The norms operation yields to:

$$\begin{aligned} \mathbf{W}_{L_1} &= \mathrm{ConvexHull}\left(\bigcup_{c_i}^{N} \mathbf{w}_{p1_{c_i}}, \ldots, \mathbf{w}_{pD_{c_i}} \right) \\ \mathbf{W}_{L\infty} &= \mathrm{ConvexHull}\left(\bigoplus_{ci} \{ \mathbf{w}_{p_{1ci}}, \ldots, \mathbf{w}_{pD_{c_i}} \} \right) \end{aligned} \tag{8}$$

where $\mathbf{w}_{pd_{c_i}} \in^o \mathbf{W}$ and $\bigoplus$ is the Minkowski sum. More detail about the norm operation can be verified in [6].

The concept of grasp closure evaluates the restraining of an object. A common assumption is the force-closure implies an equilibrium, but the inverse does not apply. A grasp has its convex hull defined by the wrenches that constitute the grasp configuration, i.e., the matrix $^o\mathbf{W}$. In a force-closure grasp, the convex hull includes the wrench space origin $\{O\}$, see Fig. 3. According to the definition presented in [37], if all wrenches in $^o\mathbf{W}$ positively span the entire wrench space, the grasp will be force-closure. Fig. 3 shows a grasp wrench space (GWS) and a convex hull of grasp configuration for force and non-force-closure, for a planar case with a fixed value for the moment (τ) in the z-axis. Therefore, it is considered $\mathbf{f}_{c_i} \in R^2\colon f_{c_i} = (f_x, f_y)$, and the resistance to perturbation in both force axes is evaluated.

Since several configurations can reach a force-closure grasp, quality metrics like ϵ-metric evaluate which one is best. The ϵ is a normalized value that represents the wrench vector's distance to the origin ({O}), which is the shortest, i.e., the worst wrench vector to support an external perturbation. An efficient grasp, ideally, has ϵ = 1. The left GWS of Fig. 3 elucidates this metric and, the readers are encouraged to a more detailed review of this grasp definition in [6].

3.2. Simulated Annealing Grasp

The SA algorithm [9] integrated in the "GraspIt!" simulator [8] is one of the tools that the grasping pipeline relies on. Since it was used from the perspective of an end user, a brief explanation will be done here, and any further information can be retrieved on the referenced papers.

The SA is a heuristic optimization algorithm based on the cooling of a set of atoms to a minimum state of energy, and it was first introduced

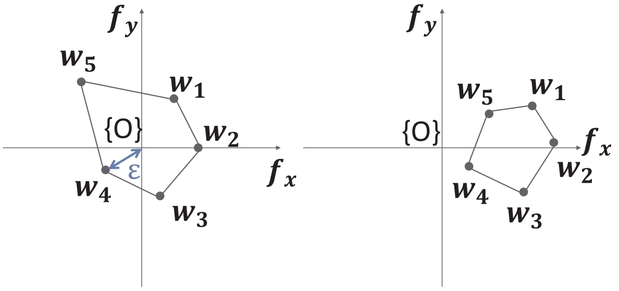

Fig. 3. Wrench convex-hull configuration. Force-closure and the ϵ-value (left). Non-force-closure (right).

by [38] in a Statistical Mechanics optimization algorithm application. The “Very Fast Simulated Re-Annealing” was an improvement made by Ingber at [39] and used here. Since it is based on temperature, Ingber proposed that its cooling process decrease as described by Eq. 9

$$T = T_0 \cdot exp(-k^{1/D}) \tag{9}$$

where D is the dimensional search space, k a SA parameter step, and T_0 is the SA the initial temperature.

Each algorithm iteration generates new state variables following a rule of neighboring. Considering current and a new variable state as $S_{current}$ and S_{new}, this rule yields Eq. 10.

$$S_{new} = S_{current} + T \cdot (-1)^{round(Rand(0,1))} \cdot \left(1 + \frac{1}{T}\right)^{Rand(-1,1)} \tag{10}$$

and the probability to change the state between the current and the new one is defined by Eq. 11 where $Q(\bullet)$ represents the objective function of the optimization problem.

$$exp\left(\frac{Q(S_{current}) - Q(S_{new})}{T}\right) > Rand(0,\, 1) \tag{11}$$

Regarding the multi-fingered grasp procedure, the objective function to be optimized by SA need to be related to the hand posture **p** and, the position and orientation of the wrist **w** as follows:

$$F_{ob} = f(\mathbf{p},\, \mathbf{w}), \quad \mathbf{p} \in \mathcal{R}^d,\, w \in \mathcal{R}^6 \tag{12}$$

where d is the number of intrinsic DOFs of the hand.

As discussed by [9], the hand posture is defined by *eigengrasps*, a subspace of movement based on how human generate hand postures. The *eigengrasps* reduces the DOFs of the hand based on how humans select appropriate grasps and hand postures. Studies show that humans simplify, unconsciously, the problem with a pattern in the movement. More information can be verified in [9,40]. The *eigengrasp* ($\mathbf{e}_i$) is defined by hand, and it is a d-dimensional direction vector that represents the motion of a group joint space that constitute it. Therefore, a posture can be defined by Eq. 13.

$$p = \mathbf{p}_m + \sum_{i=1}^{b} a_i \mathbf{e}_i \tag{13}$$

with posture origin defined by $\mathbf{p}_m$ and b the total number of *eigengrasps*. Since it is a linear combination, the parameter array $\mathbf{a} = [a_0,\, a_1,\, ...,a_b]$ will be the optimization variable in Eq. 12 together with the **w**. Therefore, the dimensional search space D has a reduced length, i.e. $D = sizeof(\mathbf{a}) + sizeof(\mathbf{w})$.

The optimization algorithm tries to minimize the linear and angular distance of the Interest Contact Point (ICP) that constitute the Interest Contact Region (ICR) (Fig. 4) adjusting the discussed optimization variables **a** and **w**. The ICR is a contact region model (a predefined group of distributed ICPs) used to calculate the interaction of the algorithm, thus it is possible to create a feasible procedure. Therefore, the objective function to be minimized is describe by Eq. 14, where N is the number of total contacts in ICR, $\hat{\mathbf{n}}_i$ is the surface normal, $\mathbf{o}_i$ the distance between the ICP and the object ($i \in N$). The scalar α is a range adjustment factor between the distance and the normalized dot product of the second sum part. It is important to note that the mapping between F_{ob} and Q is realized by simulated interaction in the “GraspIt!” [8]. A detailed description of the algorithm procedure is presented in [9].

$$Q = \sum_{i=1}^{N} (1 - \delta_i) \qquad with \qquad \delta_i = \frac{|\mathbf{o}_i|}{\alpha} + \left(1 - \frac{\hat{\mathbf{n}}_i \cdot \mathbf{o}_i}{|\mathbf{o}_i|}\right) \tag{14}$$

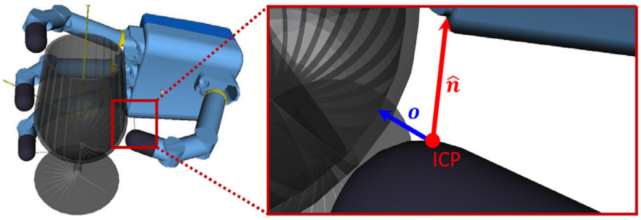

Fig. 4. Grasp optimization process elucidation.

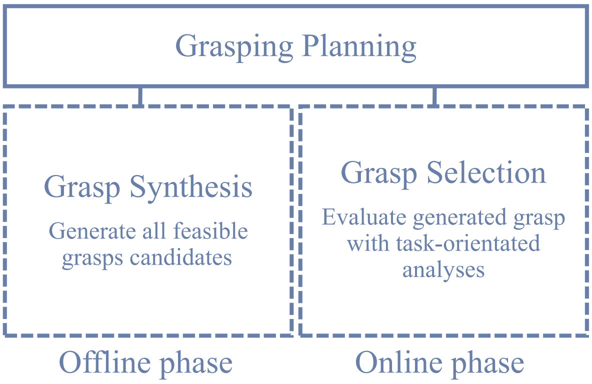
Grasping Planning
Grasp Synthesis
Generate all feasible grasps candidates
Grasp Selection
Evaluate generated grasp with task-orientated analyses
Offline phase
Online phase

Fig. 5. Grasp planning overview.

4. Proposed Grasp Planning Pipeline

The developed grasp planning pipeline is divided into two steps: grasp synthesis and grasp selection (Fig. 5). The grasp synthesis is a tool responsible for generating all the grasp poses, and it is based on the “GraspIt!” simulator. More specifically, it creates a set of hypothetical grasp candidates based on the object’s shape. It is an offline step, i.e., it runs outside the robot system in a setup phase. The generated data is then uploaded to the robot system to be used during the grasp selection step. This step is responsible for choosing the best grasp candidate following a set of heuristics and priorities. It is a task-oriented procedure that analyses the environment and the run-time constraints of the task. The following sections provide a detailed description of the procedure.

4.1. Grasp Synthesis

A summary of the grasp synthesis is presented in Fig. 6. The methodology used in this step resorts to an optimization algorithm called Very Fast Simulated Re-Annealing [41] (see Section 3.2), that was applied to the multi-fingered grasp problem (Section 3.1) proposed by the authors of [9], and currently embedded in the “GraspIt!” simulator. Its controlled stochastic structure reduces the incidence of the local minima problem, as described by [39].

This optimization algorithm is based on the feedback about the physical interaction between the gripper and the object to be grasped, i.e., in every interaction, a computer simulation is performed (see Section 3.2). Thus, the 3D models of the active fingers are needed. For the use case presented in our proposal, we relied on the RobotiQ 2F-85, which is an adaptive two-finger gripper with a maximum opening of 85 millimetres. Experimental tests with this gripper show that it can grasp the object models presented in the use case dataset. The 3D models of the gripper (adapted from [42]) are presented in Fig. 7: the left one is used to test and visualize the possible grasp solutions while the second one is effectively used in the optimization step. Both models are modeled with joints movement capabilities. However, only the second one takes into consideration the contact model, i.e., the physical model interaction associated with the visual model. Consequently, the model used in the optimization is simplified, and it reduces the computational complexity and possible collisions between the links of the gripper itself. An arbitrary number of contact points are selected for the contact model, these points constitute the ICR, and they belong to the fingertips (Fig. 8). The number of points that are considered in the ICR affects the algorithm performance: a large set of points increases the reliability of

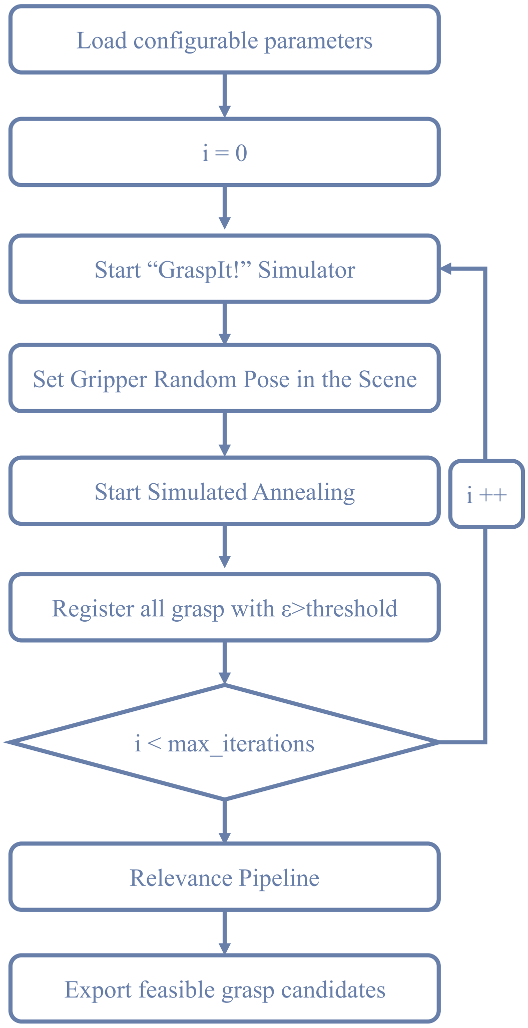
Load configurable parameters
i = 0
Start "GraspIt!" Simulator
Set Gripper Random Pose in the Scene
Start Simulated Annealing
i ++
Register all grasp with ε>threshold
i < max_iterations
Relevance Pipeline
Export feasible grasp candidates

Fig. 6. Grasp synthesis process flowchart.

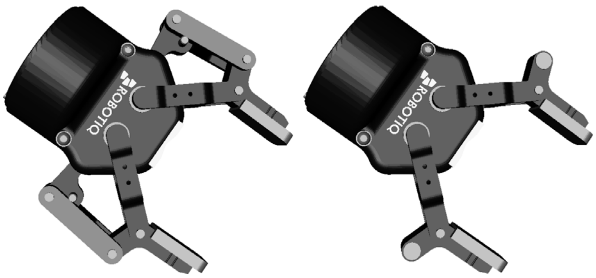

Fig. 7. 3D gripper models used (adapted from [42]). Complete model (left) and simplified model (right).

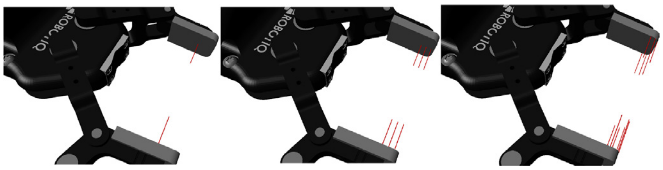

Fig. 8. Possible configuration of ICR. From left to right the number of contact points increases. The model becomes more reliable but convergence of the algorithm is more prone to issues.

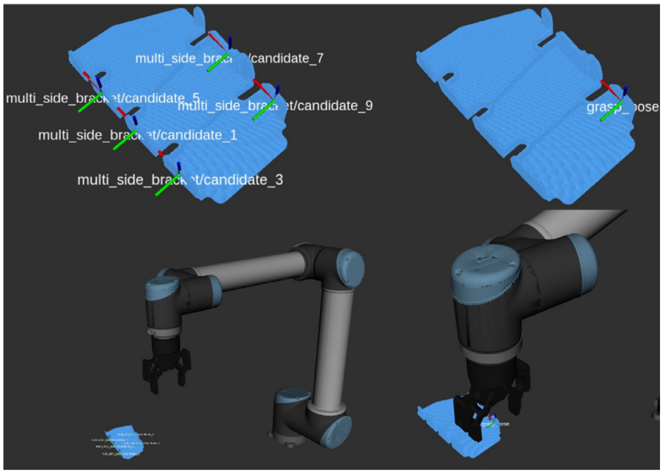
multi_side_bra /candidate_7
multi_side_bra t/candidate_5
multi_side_bra t/candidate_9
multi_side_bra t/candidate_1
multi_side_bra t/candidate_3
grasp_ ose

Fig. 9. The grasp selection process. The object with the grasp candidates (top left). The best candidate selected (top right). The picking movement (bottom).

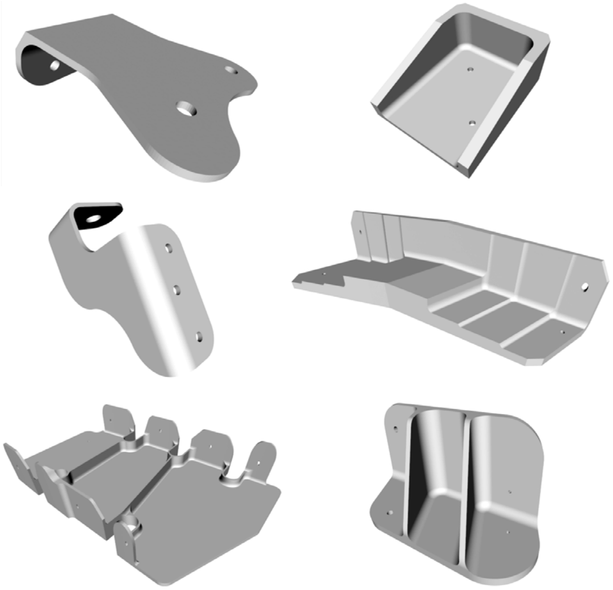

Fig. 10. CAD parts model representation of the grasp test cases, namely, bracket (top left), single-side bracket (top right), double-side bracket (middle left), support bracket (middle right), multi-side bracket (bottom left) and reinforced bracket (bottom right).

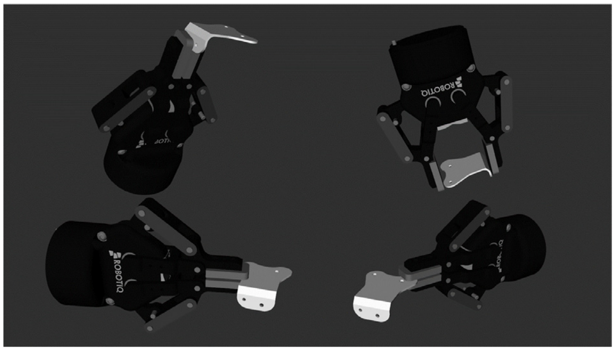

Fig. 11. Examples of generated grasps for the bracket object.

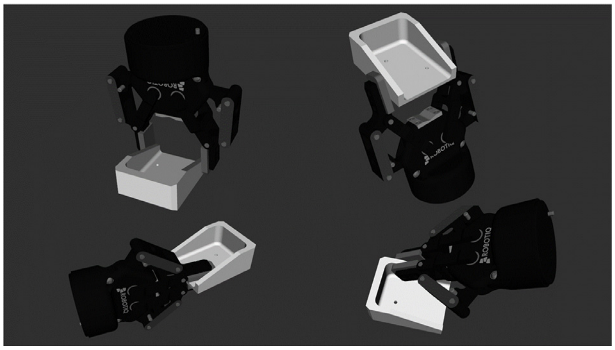

Fig. 12. Examples of generated grasps for the single-side bracket object.

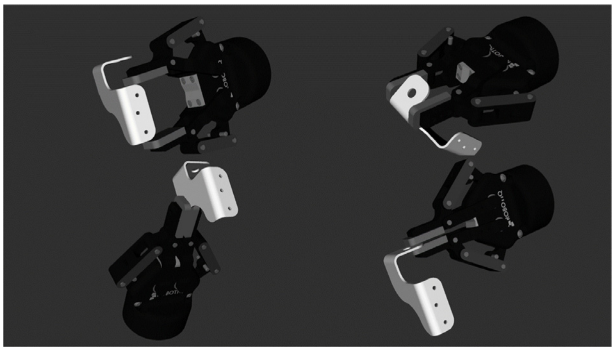

Fig. 13. Examples of generated grasps for the double-side bracket object.

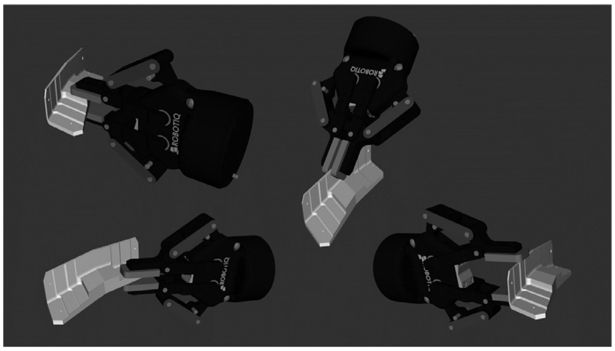

Fig. 14. Examples of generated grasps for the support bracket object.

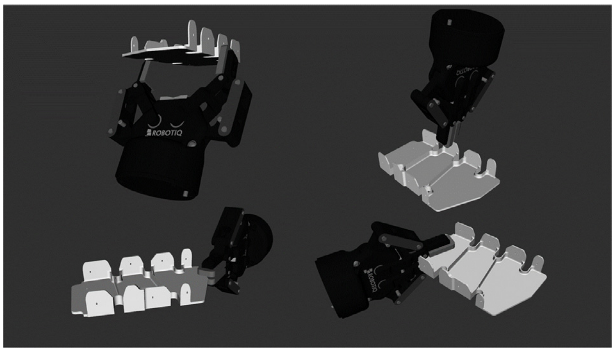

Fig. 15. Examples of generated grasps for the multi-side bracket object.

the grasp procedure, but it could compromise the convergence of the algorithm.

In the present use case, the gripper has only one *eingengrasp* (Section 3.2), simplifying the converge of the SA algorithm. This approach is adequate to model all joints movement as only one group, since the gripper does not have independent fingers and only performs the opening-closing procedure.

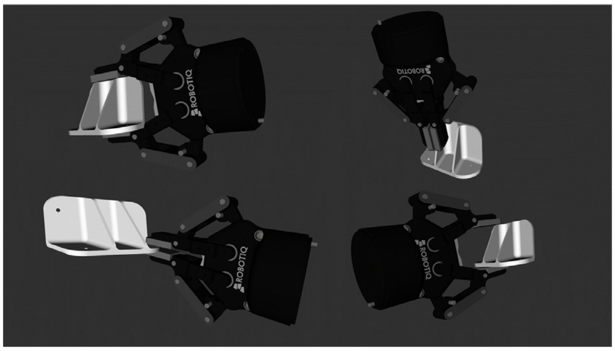

Fig. 16. Examples of generated grasps for the reinforced bracket object.

To run the grasp synthesis and initialize the process of Fig. 6, a launch file is configured with the object to be grasped, the gripper type, the ϵ-value (the quality metric used, discussed in Section 3.1) and the maximum iterations thresholds. A fine-tuning of SA parameters is also possible.

The Grasp Viewer interface (seen in Fig. 11) shows the progress of grasp finding for each iteration. At the end of each SA, a set of good grasps is stored and a new simulation is launched until the maximum iterations are reached.

After the grasp synthesis process executed all iterations of the optimization algorithm, the relevance pipeline evaluates all grasp candidates. Therefore, no redundant grasp poses are generated. This pipeline merges grasps that are close to each other by angular and linear distance based on configurable thresholds, also defined in the launch file. In the end, a configuration file (structured in a.YAML file) is exported with all the grasp candidates that will be used by the grasp selection procedure.

4.2. Grasp Selection

The grasp selection is a ROS package designed for choosing the best grasp over a set of previously taught grasp poses of an object. This step is an online procedure; i.e., it is a run-time process performed during the robot task execution. Thus, this operation needs to be fast and reliable. The grasps candidates are specified in a YAML configuration file and loaded in the ROS parameter server. These candidates are unique for each object and are generated by the grasp synthesis step. Once the candidates are created and loaded in the parameter server, the grasp selection pipeline estimates the best grasp candidate for allowing the robot to pick the object. The best grasp candidate is chosen according to a cascade of heuristics defined by the user in a YAML file for each object detected. Therefore, the object needs to be identified and localized before the grasp selection process. The heuristics cascade gives a score for each grasp candidate related to a reference frame (such as the gripper). The one with the lower cost is the eligible candidate. It is possible to define a weight for each heuristic in the pipeline based on the importance level of each method in the application. An illustration of the described procedure is presented in Fig. 9.

Below are the set of heuristics supported by the grasp selection pipeline:

- **Joint space filter:** This method is a filter that discards candidates that exceed joint thresholds using the inverse kinematics of the robotic arm. The inverse kinematics of each possible grasp is calculated using the `trac_ik` library [43]. This avoids the robot to choose a grasp candidate that requires an impossible, impracticable, or dangerous movement for picking the object.
- **Depth distance:** The score is given according to the depth distance

Table 1
Grasp planning qualitative results.

Parameters	Bracket	Single-side bracket	Double-side bracket	Support bracket	Multi-side bracket	Reinforced bracket
Iterations	30	30	30	30	80	30
Time of convergence (offline phase)	22min33s	42min16s	48min04s	32min48s	158min54s	39min24s
Number of force-closure grasps generated (offline phase)	433	278	294	385	1239	272
Number of relevant grasps (offline phase)	48	83	42	44	150	37
Number of grasps after human supervision (offline phase)	38	83	38	44	147	34
Mean decision time (online phase)	0.70s	1.77s	0.66s	0.74s	2.54s	0.57s

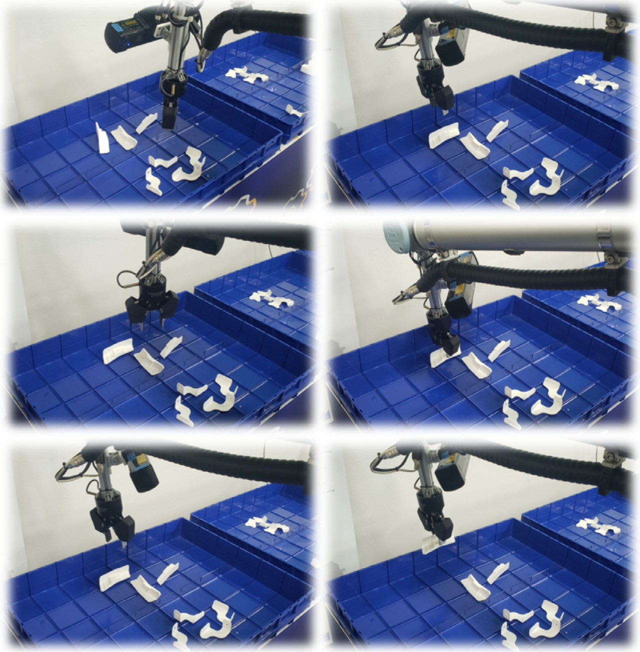

Fig. 17. Full picking process of the support bracket. Scanning position (top left). Initial movement (middle left). Approach orientation (bottom left). Approach movement (top right). Grasp action (middle right). Lifting movement (bottom right).

between the reference (gripper) and object coordinate frames.

- **Euclidean distance:** the score is given according to the Euclidean distance (3D distance) between the reference and object frames.
- **Roll distance:** The score is given according to the roll distance of the object with relation to the reference frame (cumulative Euclidean Z-Y-X order).
- **Pitch distance:** The score is given according to the pitch distance of the object with relation to the reference frame (cumulative Euclidean Z-Y-X order).
- **Yaw distance:** The score is given according to the yaw distance of the object with relation to the reference frame (cumulative Euclidean Z-Y-X order).

5. Results and Evaluation

The dataset used in the evaluation of the grasp planning pipeline is constituted by a set of objects frequently stored in aerospace automated warehouses and handled by operators. Fig. 10 presents the CAD representations of this test case which were used in the grasp synthesis pipeline with the model of the RobotiQ 2F-85 gripper (Fig. 7).

The grasp synthesis pipeline automatically generated grasp candidates for each object in an offline phase. Fig. 11 to Fig. 16 show some examples of grasp candidates rendered by the grasp viewer package. It is important to note that for all cases, 30 iterations were chosen to generate the grasps. However, for the multi-side bracket, more iterations were necessary (a total of 80, shown in Table 1). This situation happens because of the topology of the part: the multi-side bracket possesses small (the seven lateral lumps) and large (the base) graspable surfaces. The SA tends to converge to the largest region (see bottom left and bottom right grasps hypothesis of Fig. 15), i.e., to the easiest graspable part of the object shape. Therefore, a higher number of iterations may improve the detection of small graspable regions (like the top left and top right grasps hypothesis of Fig. 15) since the initial gripper's position can avoid the recurrence of the minima local problem.

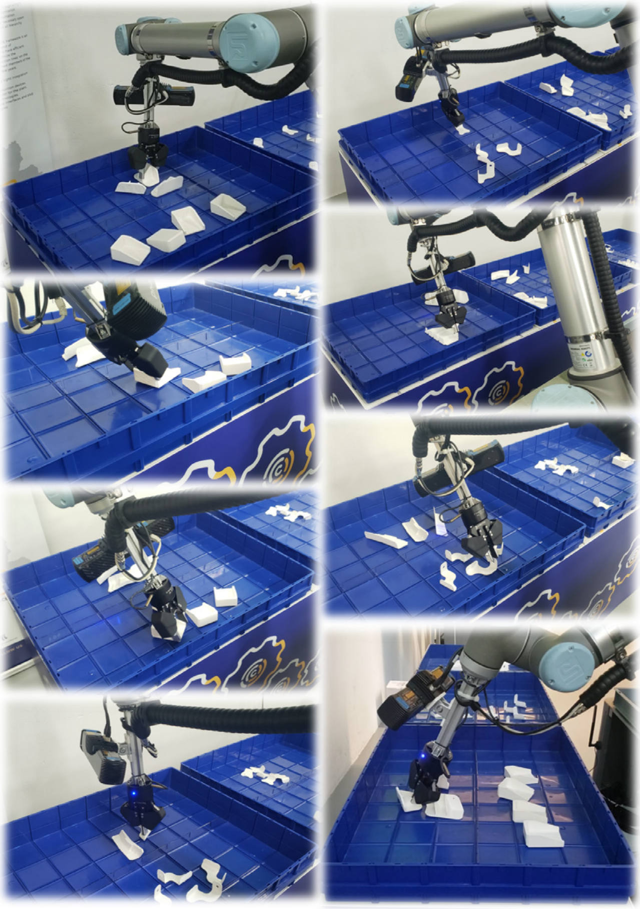

Fig. 18. Experimental sample of grasps over the objects database.

It is also important to consider the ICR model presented in Section 3.2. As already mentioned in Section 4.1, this affects the algorithm's performance. ICRs allocated in the inner part of the gripper could lead the algorithm to not converge. In that case, the fingertip's edge can collide with the object before the ICR reaches the contact, i.e., the ICR does not touch the lumps' surface in the case of the multi-side bracket. If many contact points are selected, a contact region larger than the actual grasp region could be originated, and this could lead to a situation where not every contact point belongs to the object. Small ICR or ICR located very near the fingertip edge instead can generate stable but unpractical grasps. This can be seen in the bracket (top right grasp of Fig. 11) and support bracket (bottom right grasp of Fig. 14). These cases are unpractical since any object sensing errors or bin disturbance can invalidate the grasp selection.

Therefore, it is important that a human supervisor, with the developed Grasp Viewer package, analyses and selects the grasps hypothesis to be applied in the real bin picking scenario.

The convergence time of the grasp synthesis algorithm is presented in Table 1, including the mean decision time. These performance values were generated with a medium-end computer with 12GB of RAM and 1.80GHz CPU (i7-8550U). It is important to note that for each iteration of the SA, 100000 poses were tested for the localization of contact point between the active pairs / fingers. Many poses were not force-closure candidates, i.e., there were no contacts to a minimum force-closure grasp. The relevance pipeline reduced the number of grasps that were almost the same, with a limit of 20 degrees in each angular axis and 1 centimeter of Euclidean distance between the grasp candidates. Analysing each object with the Grasp Viewer, four out of the six parts had grasp candidates removed by a human supervisor. These grasps could be unfeasible in the picking scenario use case, like the top right grasp of Fig. 11 and the bottom right grasp of Fig. 14. An important point also verified in Table 2 is the mean decision time. This parameter is the time that the grasp selection algorithm takes to decide which pose candidate will be chosen. This time needs to be small for a picking application. Therefore, with the approach of uncoupling the grasp pipeline in offline and online steps, the grasp selection becomes fast enough for a picking task.

An experimental pick and place scenario (seen in Fig. 1) was set up in the Industry and Innovation Laboratory of INESC TEC to evaluate the grasp planning pipeline. The robot used was an omnidirectional mobile manipulator equipped with the UR10 robotic arm, the RobotiQ 2F-85 gripper and the PhotoNeo PhoXi S sensor for 6 DoF object pose estimation.

The files generated by the grasp synthesis were uploaded to the robot to be used by the grasp selection pipeline in the picking task. The grasp selection cascade of heuristics used relied on the following heuristics: joint space filter, Euclidean distance, along with the roll, pitch, and yaw distances. After the object recognition and pose estimation, the grasp selection was able to choose a suitable candidate given the pose of the robot concerning the bin, the joints constraints, and the grasp list hypothesis already loaded in its memory. Fig. 17 shows a full picking procedure for the support bracket case. Fig. 18 shows different grasps for distinct parts of the data set.

6. Conclusion

This paper addressed the development of a grasp planning pipeline that is able to automatically generate grasps over a set of recognized objects and also select the best grasp with task-orientated capabilities (i.e., considering the environment and run-time constraints of the task), being endowed with methodologies and tools already consolidated in the scientific community. Tests were performed and presented considering a real intralogistic use case scenario in the aerospace industry.

Currently, the proposed grasp synthesis pipeline is built on top of the “GraspIt!” simulator and its SA optimization algorithm. Combined with the grasp selection, the grasp synthesis composes the designed grasp planning that extends the ‘GraspIt!” applicability. Therefore, the proposed grasp planning transfers the simulated methodology to the real environment with task-orientated capability. In summary, the grasp synthesis generates and stores several solutions, interactively changing the initial conditions of SA. After evaluating the stability metric, unnecessary solutions are removed, and a feasible and not redundant grasp database is created. The grasp synthesis is performed in an offline phase for acelerating the run time phase by pre-generating the grasp candidates, that later on are analysed at run time for selecting the most suitable given the task conditions and restrictions.

It should be highlighted that the proposed pipeline intends to be flexible and user-friendly being that, at this moment, it allows the user to choose the type of gripper and object to be grasped by just providing their 3D models. Furthermore, it also allows setting each heuristic and organize them in a cascade flow with different orders and importance levels.

As future work, other methods will be implemented and embed into the reconfigurable pipeline structure (e.g. centroid estimation of planar surfaces for suction grasp; the CNN for agnostic-object grasping proposed by [23]. Also, a user-friendly interface will be designed. Therefore, the user will be able to choose between different methodologies and select the best appropriate ones for its application.

Regarding the grasp selection package, new filters and heuristics will be added, such as one to score the grasp result according to the approaching and lifting vector (since some grasp points can generate collision situations between the scenario and the object).

Although motivated by intralogistic use cases in aerospace factories, the proposed reconfigurable and modular pipeline was designed to be applicable in different scenarios. Our main objective is to integrate it into our bin-picking solution that is still in development. As future work, we also intend to assess the performance of the proposed pipeline for automatic picking of product on conveyors-belts at fast consumer goods factories and for the manipulation of tools in collaborative robot applications.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] V.-D. Nguyen, Constructing force-closure grasps in 3d, Proceedings. 1987 IEEE International Conference on Robotics and Automation, 4 IEEE, 1987, pp. 240–245.

[2] V.-D. Nguyen, Constructing stable grasps, The International Journal of Robotics Research 8 (1) (1989) 26–37.

[3] B. Dizioğlu, K. Lakshiminarayana, Mechanics of form closure, Acta mechanica 52 (1-2) (1984) 107–118.

[4] J. Ponce, B. Faverjon, On computing three-finger force-closure grasps of polygonal objects, IEEE Transactions on robotics and automation 11 (6) (1995) 868–881.

[5] J.-W. Li, H. Liu, H.-G. Cai, On computing three-finger force-closure grasps of 2-d and 3-d objects, IEEE Transactions on Robotics and Automation 19 (1) (2003) 155–161.

[6] C. Ferrari, J.F. Canny, Planning optimal grasps. ICRA, 3 (1992), pp. 2290–2295.

[7] J. Bohg, A. Morales, T. Asfour, D. Kragic, Data-driven grasp synthesis a survey, IEEE Transactions on Robotics 30 (2) (2013) 289–309.

[8] A.T. Miller, P.K. Allen, Graspit! a versatile simulator for robotic grasping, IEEE Robotics & Automation Magazine 11 (4) (2004) 110–122.

[9] M.T. Ciocarlie, P.K. Allen, Hand posture subspaces for dexterous robotic grasping, The International Journal of Robotics Research 28 (7) (2009) 851–867.

[10] M.A. Roa, R. Suárez, Finding locally optimum force-closure grasps, Robotics and Computer-Integrated Manufacturing 25 (3) (2009) 536–544.

[11] A. Saxena, J. Driemeyer, A.Y. Ng, Robotic grasping of novel objects using vision, The International Journal of Robotics Research 27 (2) (2008) 157–173.

[12] G. Dini, F. Failli, Planning grasps for industrial robotized applications using neural networks, Robotics and Computer-Integrated Manufacturing 16 (6) (2000) 451–463.

[13] R. Pelossof, A. Miller, P. Allen, T. Jebara, An svm learning approach to robotic grasping, IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004, 4 IEEE, 2004, pp. 3512–3518.

[14] J. Redmon, A. Angelova, Real-time grasp detection using convolutional neural networks, 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015-June IEEE, 2015, pp. 1316–1322, https://doi.org/10.1109/ICRA.2015.7139361.

[15] U. Asif, J. Tang, S. Harrer, Ensemblenet: Improving grasp detection using an ensemble of convolutional neural networks. BMVC, (2018), p. 10.

[16] Y. Song, L. Gao, X. Li, W. Shen, A novel robotic grasp detection method based on region proposal networks, Robotics and Computer-Integrated Manufacturing 65 (2020) 101963.

[17] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, N. Xi, A hybrid deep architecture for robotic grasp detection, 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 1609–1614, https://doi.org/10.1109/ICRA.

2017.7989191.

[18] A. Gariépy, J.-C. Ruel, B. Chaib-draa, P. Giguere, Gq-stn: Optimizing one-shot grasp detection based on robustness classifier, arXiv preprint arXiv:1903.02489 (2019).

[19] G. Ghazaei, I. Laina, C. Rupprecht, F. Tombari, N. Navab, K. Nazarpour, Dealing with Ambiguity in Robotic Grasping via Multiple Predictions, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11364 LNCS (2019) 38–55, https://doi.org/10.1007/978-3-030-20870-7_3.

[20] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J.A. Ojea, K. Goldberg, Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics, arXiv preprint arXiv:1703.09312 (2017).

[21] J. Mahler, F.T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, K. Goldberg, Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards, 2016 IEEE international conference on robotics and automation (ICRA), IEEE, 2016, pp. 1957–1964.

[22] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, K. Goldberg, Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning, 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 1–8.

[23] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, K. Goldberg, Learning ambidextrous robot grasping policies, Science Robotics 4 (26) (2019) eaau4984.

[24] M.U. Suleman, M.M. Awais, Learning from demonstration in robots: Experimental comparison of neural architectures, Robotics and Computer-Integrated Manufacturing 27 (4) (2011) 794–801.

[25] O. Kroemer, R. Detry, J. Piater, J. Peters, Combining active learning and reactive control for robot grasping, Robotics and Autonomous systems 58 (9) (2010) 1105–1116.

[26] S. Ekvall, D. Kragic, Learning and evaluation of the approach vector for automatic grasp generation and planning, Proceedings 2007 IEEE International Conference on Robotics and Automation, IEEE, 2007, pp. 4715–4720.

[27] J. Romero, H. Kjellstrom, D. Kragic, Modeling and evaluation of human-to-robot mapping of grasps, 2009 International Conference on Advanced Robotics, IEEE, 2009, pp. 1–6.

[28] D.S. Wheeler, A.H. Fagg, R.A. Grupen, Learning prospective pick and place behavior, Proceedings - 2nd International Conference on Development and Learning, ICDL 2002 (2002) 197–202, https://doi.org/10.1109/DEVLRN.2002.1011865.

[29] A. Boularias, J.A. Bagnell, A. Stentz, Learning to manipulate unknown objects in clutter by reinforcement, Twenty-Ninth AAAI Conference on Artificial Intelligence, (2015).

[30] S. D'Avella, P. Tripicchio, C.A. Avizzano, A study on picking objects in cluttered environments: Exploiting depth features for a custom low-cost universal jamming gripper, Robotics and Computer-Integrated Manufacturing 63 (2020) 101888.

[31] J. Chen, S. Yang, Y. Li, Y. Huang, Z. Yin, Active curved surface deforming of flexible conformal electronics by multi-fingered actuator, Robotics and Computer-Integrated Manufacturing 64 (2020) 101942.

[32] V. Babin, D. St-Onge, C. Gosselin, Stable and repeatable grasping of flat objects on hard surfaces using passive and epicyclic mechanisms, Robotics and Computer-Integrated Manufacturing 55 (2019) 1–10.

[33] A. Delgado, C. Jara, F. Torres, In-hand recognition and manipulation of elastic objects using a servo-tactile control strategy, Robotics and Computer-Integrated Manufacturing 48 (2017) 102–112.

[34] A. Björnsson, M. Jonsson, K. Johansen, Automated material handling in composite manufacturing using pick-and-place systems–a review, Robotics and Computer-Integrated Manufacturing 51 (2018) 222–229.

[35] L. Birglen, T. Schlicht, A statistical review of industrial robotic grippers, Robotics and Computer-Integrated Manufacturing 49 (2018) 88–97.

[36] R.M. Murray, Z. Li, S.S. Sastry, S.S. Sastry, A mathematical introduction to robotic manipulation, CRC press, 1994.

[37] J.K. Salisbury, B. Roth, Kinematic and force analysis of articulated mechanical hands (1983).

[38] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by Simulated Annealing, Science 220 (4598) (1983) 671–680.

[39] L. Ingber, Very fast simulated re-annealing, Mathematical and computer modelling 12 (8) (1989) 967–973.

[40] M. Santello, M. Flanders, J.F. Soechting, Patterns of Hand Motion during Grasping and the Influence of Sensory Guidance, 22 (4) (2002) 1426–1435.

[41] G. Britain, M.P. Macmillan, N. Ingber, VERY FAST SIMULATED RE-ANNEALING, 12 (8) (1988) 967–973.

[42] StanleyInnovation, robotiq_85_gripper, 2019, (accessed 19 February 2020).

[43] tracklabs, track_ik, 2020, (accessed 19 February 2020).