



Automated Fixture Design using an Imprint-based Design Approach & Optimisation in Simulation

Lukas Christoffer Malte Wiuf Schwartz, Lars-Peter Ellekilde, and Norbert Krüger

Maersk McKinney-Møller Institute,
University of Southern Denmark, Odense, Denmark
lukaswiufs@hotmail.com {lpe,norbert}@mmmi.sdu.dk
www.sdu.dk/

Received (11/02/2018)

Revised (06/15/2019)

Accepted (07/15/2019)

Abstract. Object aligning and holding fixtures for robotic assembly tasks are important in industry in order to successfully complete an assembly. However, the designing of a fixture is usually done manually which can be a long and tedious process including many iterations, even for experienced engineers. This paper presents a method to design fixtures automatically for use in robotic assemblies and pick-and-place tasks. To achieve this, a new automated method to design the cut-out for a fixture is introduced. The method uses a parameterized version of the object's imprint to design the cut-out. The fixtures generated using this method are optimized in simulations to determine their final parameters for a specific application. Dynamic simulations are used to evaluate each iteration of the cut-out. Lastly, the method is applied to a use-case from the industry to design a fixture for use in a robotic assembly task.

Keywords: Simulation; Fixture Learning; Robotics; Optimizations; Assembly.

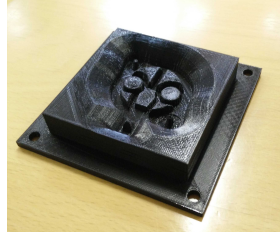
1 Introduction

Assembly processes are often concerned with picking up and fitting two or more objects together before they are securely fixed. Considering the case shown in Fig. 1: A large drive is fitted with a smaller flat object, called the topplate. The assembly is completed by pushing the topplate onto the end of the drive.

For this, a fixture is used, which is a structure used for supporting, holding, and/or aligning objects. An example of the fixture designed in this paper is seen in Fig. 1c. With the help of the fixture, the robot can place one of the objects in



(a) Disassembled set-up. (b) The assembled parts.



(c) A 3D printed fixture.

Fig. 1: Assembly of two objects and a fixture.

the fixture. Thereafter the second object is picked up by the robot and assembled with the object in the fixture. However, this requires a fixture and a set of fingers for the robot to be designed so that the assembly can be completed successfully. The gripper and fixture designs must therefore be designed such that the objects are held with sufficient force as to resist the wrenches experienced on the objects during the assembly. Furthermore, the designed fingers and fixtures need to locate the objects precisely enough such that they can be assembled.

Designing fixtures and gripping fingers is a problem that frequently takes several iterations of trial and error evaluations, even for experienced engineers using heuristics and guidelines in the process [3,16]. Here a method is introduced which can be used to design fixtures usable in assembly and pick-and-place operations in industry by means of software. The method uses simulations in the optimization process and final evaluations are made in simulations and real world experiments to verify the designed fixtures. Furthermore, the inclusion of the optimization process replaces the need for potentially large amount of intermediate prints of the fixture and automates the design process. This simplifies and accelerates the process and also allows for a faster and more thorough testing throughout the design phase. This makes the fixture design cheaper in terms of prototyping and manufacturing.

The fixtures will be designed as static objects such that they can be manufactured quickly either using 3D printing or Computer Numerical Control (CNC) machines. This is unlike common fixtures for machining tasks involving clamps, locators etc [16].

The idea of using optimizations in simulations was inspired by the approach presented in [17] for gripper design, where a parameterized finger model is op-

timized to get the best suited gripper for a specific task. Unlike in [17], the process is significantly automated by introducing an automated parametrization method. This method, while applied here for fixtures, was also generalized for gripper design [14]. The method is covered by the patent [13].

2 State of the art

Fixture design is an area of study that has been researched for many years. Traditionally a fixture consists of a plate with holes to which the locators and clamps can be mounted to [16]. The locators are static and mounted against the object fixed in the fixture system. The layout of the locator is of varying shape and size, some intended for a specific object and others more general [2].

The early work in this field has focused on the automation of the design [2] and planning process and the analysis of the fixture [10]. [18] developed a planning algorithm that is able to plan how a given modular fixture should be built, step by step. The reconfigurability was also explored as to have robots reconfiguring single fixtures for a new production [19].

Tools have been developed to optimize a fixture layout by setting up a set of scoring systems to evaluate the fixture [8]. Some of the programs explored the option of reusing previous fixture designs from a database, also known as using the Case-Based Reasoning (CBR) approach [6,1]. This process is initiated with the selection of the base template from a set of previous cases. The fixture is then post-modified for the specific object from the base case to obtain a higher performance. The starting template is found in a library by a search algorithms to automatically find the base fixture.

In this work, the fixture design is considered from a different perspective than in the work discussed above. Instead of considering a clamp-locator fixture, a fixture is here an object that does not have any moving parts. However, it is able to hold and locate the object it is designed for. The system developed to design the fixture uses dynamic simulations to optimize the fixture design. This is different from the previous work which only uses numerical optimizations to finalize the locator and clamp positions.

The method introduced here to create fixtures was also previously applied to finger design in the Gripperz framework [17,14]. However, this paper gives a more detailed explanation of the method used to construct the imprint. Furthermore the fixture designed here is tested in real-world experiments in order to verify the simulations.

3 Methods

Designing fixtures for assembly sequences is a crucial task taking a long time if done manually. This section introduces a framework that automates this design process. Sec. 3.1 first describes the method used to design the fixture shape itself. Thereafter Sec. 3.2 describes how this method is used in a larger framework to design an optimized fixture for a specific object.

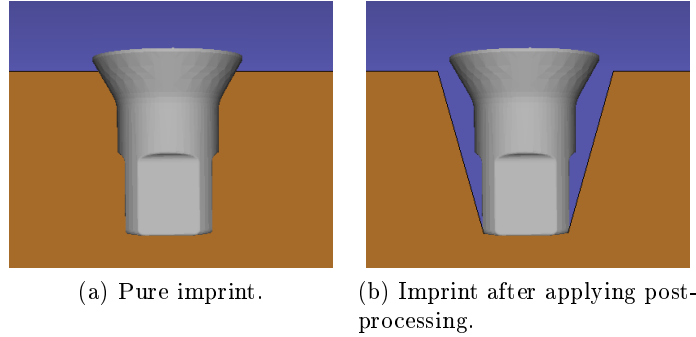


Fig. 2: Visualisation of the imprint and post-processing.

3.1 Fixture imprint parametrization

When designing fixtures for specific objects, a cut-out in a basic shape of a fixture is usually made [4]. This cut-out is what is supposed to enhance the performance of the fixture in its use since it can provide alignment capabilities, wrench resistance during assemblies, etc.

The alignment property of a fixture is a measure of the uncertainty that can be accounted for when placing the object into the fixture. High alignment therefore decreases the risk of the object not reaching the correct pose, improving the success-rate of the assembly.

Using the imprint of the object directly as the cut-out, is visualized in Fig. 2a. While this can give the fixture a high wrench resistance using form and force closure of the object, it generally results in low alignment capabilities. This is since it requires a high degree of pose certainty to place an object into its cut-out. To improve the alignment performance, the cut-out can be post-processed as seen in Fig. 2. The method used is comparable to carving out material from the clay block giving the imprint alignment capabilities while also retaining some of the objects shape as seen in Fig. 2b.

Three parameters define the result of the post processing. These are a function and two values. The effect of the parameters are illustrated in Fig. 3. The input function, further onwards called “*profile*”, is defined as the function $f : [0, 1] \rightarrow [0, 1]$. This profile describes the shape of the cut-out. The two other parameters, called *tolerance-x* (t_x) and *tolerance-y* (t_y), are from the set of $\{0, \mathbb{R}^+\}$. t_x and t_y define the width of the profile in the direction of respectively the x- and y-axis, as denoted by the subscript. How these parameters affect the cut-out is illustrated in Fig. 3 (Fig. 3 is only a 2D cross-sectional view along the x-axis and only t_x is hence shown).

The cut-out can be applied with any user specified function (*profile*) taking into account the criteria mentioned before. We decided to use the profile defined as $f^b(a) = a^b$, with $b \in \mathbb{R}^+$. The variable b is kept constant for the full generation of a cut-out. This makes it possible to optimize the shape of the profile, varying b between cut-outs during the optimization process. The profile $f^b(a) = a^b$ was chosen because it for $0 < b \leq 1$ and $a \in [0, 1]$ produce steep slopes that make

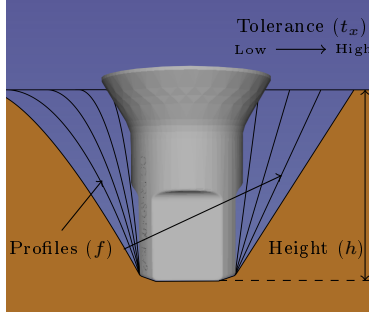


Fig. 3: Visualisation of parameters in the design of the cut-out. Profiles defining the shape and tolerance the width of the profile. Left and right two different profiles are shown.

the imprint resemble the object more closely. This makes for a larger contact surface between the fixture and object, improving the wrench the object is able to resist when placed in the fixture. Furthermore its rounded shape produces a guiding surface for the object to follow into the fixture. Thus the choice of b can be seen as selecting between optimum alignment vs holding properties.

When the profile is applied to the cut-out, first the height, h , of the cut-out is found. This is used to scale the profile together with the user specified tolerances t_x and t_y , to get the profile G_p , see equation (1). The profile is then applied to the imprint.

To apply the profile to the cut-out, it is first discretised in steps. This is done by dividing the profile width t_x and t_y into a set of steps t_x^i and t_y^k with indices i and k respectively. Where $i, k \in \mathbb{Z}$ so that i is the set $\{-n_x, \dots, -1, 0, 1, \dots, n_x\}$ and k is the set $\{-n_y, \dots, -1, 0, 1, \dots, n_y\}$. n_x and n_y is the number of steps the profile is divided into along the respective axes. The values of i and k are going in the positive and negative direction because the profile is applied in both directions of the x- and y-axis.

$$G_p(x, y, i, k) = G(x, y) + h \cdot f\left(\sqrt{(t_x^i)^2 + (t_y^k)^2}\right) \quad (1)$$

Equation (2) is then applied for x_l and y_m with $x_l \in \{1, \dots, res_x\}$ and $y_m \in \{1, \dots, res_y\}$ where res_x and res_y is the size of the discretised imprint. The shape of the new imprint is stored on a second heightmap called G_{new} such that G can be used as reference during the computation. Equation (2) uses the case choosing between the entry of G_p and G_{new} to ensure that no extra material is added to the fixture. This guarantees that the object always can be placed into the fixture.

$$G_{new}(x + i, y + k) = \begin{cases} G_p(x, y, i, k), & \text{if } G_p(x, y, i, k) < G_{new}(x + i, y + k) \\ G_{new}(x + i, y + k), & \text{otherwise} \end{cases} \quad (2)$$

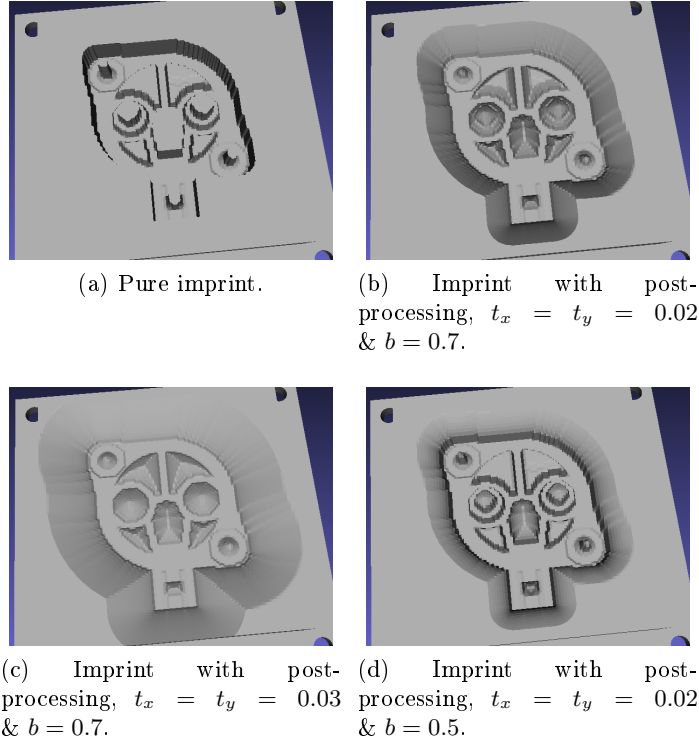


Fig. 4: Fixture layout for different parameters using the profile $f^b(a) = a^b$.

where $G_p(x, y, i, k)$ is the value of the entry $G(x + i, y + k)$ when applying the profile around $G(x, y)$. Afterwards a heightmap from the final G_{new} is used to create the cut-out as a tri-mesh from the fixture.

This results in a parameterized fixture requiring only the object and three parameters, the profile and two tolerances, to be specified. Fig. 4 illustrates how the fixture design varies for the given parameters using the profile function $f^b(a) = a^b$. First Fig. 4a shows the fixture when no post processing is done and Fig. 4b a default layout with post-processing. Finally Fig. 4c and 4d illustrate the effect of increasing the tolerance and changing the profile value respectively.

Given this parameterized fixture layout, the best suitable fixture can then be found using optimization. Sec. 3.2 therefore explains how the fixture is optimized using simulations as a tool of evaluating the fixture.

3.2 Framework for designing & optimizing fixtures

In order to design and optimize fixtures using the parametrization presented in Sec. 3.1, a framework was created featuring a set of tools. The framework utilizes the *RobWork* library [5] and its simulation package *RobWorkSim* [7] using the ODE engine [15].

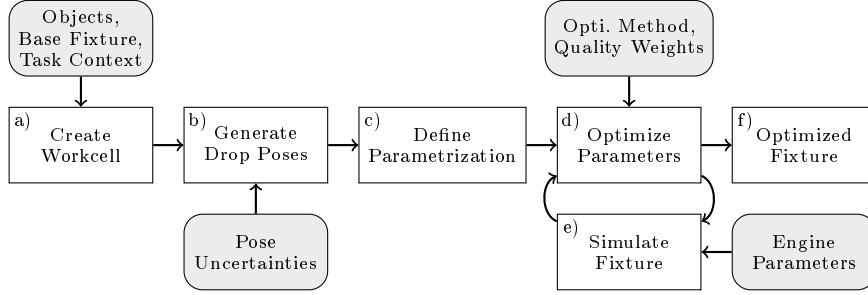


Fig. 5: The fixture design pipeline. Grey representing inputs and in white the steps the user and program goes through.

It also provides tools to evaluate the fixture alignment and wrench using simulations. The fixtures can be designed using the process pipeline seen in Fig. 5. The grey boxes illustrate the input that is supplied by the user.

The first step in the process of designing a fixture is the creation of the workcell (Fig. 5a). During runtime, the geometry of the fixture is then updated while testing and evaluating different version of the fixture.

Secondly a set of drop poses are generated (Fig. 5b), as seen in Fig. 6, from which the test object is released when put into the fixture. The drop poses can be generated using either stochastic or regular sampling. When generating the drop poses, the range in which the poses are sampled is based on requirements of how much pose uncertainty the fixture is needed to account for. The regular sampling is done in a 6D cube spanning the required volume specified by the user. The stochastic sampling is done by sampling random poses within a 3D sphere to which a random rotation is added. For this method the user is only required to supply a maximum angular and translational distance from which the poses are generated.

The last step in the setup-phase is the definition of which parameters to use and the bounds of the optimization (Fig. 5c). Currently supported are the two tolerance values, t_x and t_y , the profile parameter (in this case b in $f^b(a) = a^b$) and a height modifier of the imprint position, see Sec. 3.1. The height modifier can change the depth of the object cut-out in the base fixture.

Once this set of steps has been performed, the optimization can be started. The optimization loop (Fig. 5d and 5e) then optimizes the fixture using the simulator to evaluate each step. Upon completion of the optimization process the best performing fixture found in the process is returned to the user (Fig. 5f).

Fixture Evaluation In order to optimize the fixture, a set of quality scores are needed in order to quantify and compare the performance of the produced fixtures. The scores are normalized in a manner such that they are in the range of $[0, 1]$.

During the optimization, the scores are combined for one final objective score using the geometric mean,

$$S_{geo} = (q_a^{w_a} \cdot q_w^{w_w})^{1/(w_a + w_w)} \quad (3)$$

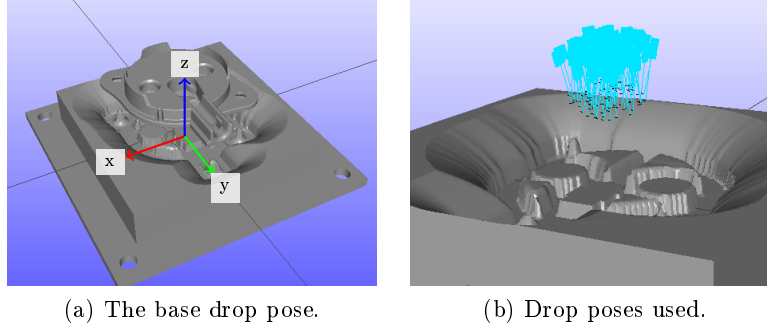


Fig. 6: Visualization of object drop poses.

where q_a and q_w are the individual quality scores calculated for alignment and wrench respectively. The values w_a and w_w are the weights associated with the given quality score and dependent on the property wished to emphasize in the optimization process.

Evaluating Fixture Alignment: Evaluations of the fixture alignment property is done by simulating the dropping of the object from above the fixture. The drops are performed using the set of specified drop locations as seen on Fig. 6b.

The alignment of the object from one specific drop pose is then evaluated simulating the dropping of the object. Based on the distance between the resting pose and expected pose of the object, it is then decided if the alignment was successful or not. The alignment score is then computed as the ratio between the number of successful and the total number of drops.

Evaluating Fixture Wrench: To evaluate the wrench space of the fixture, the Grasp Wrench Space (GWS) [11] is used. This implementation of the GWS uses an object specific torque scaling. The scale factor was set to $\lambda = 1/X$ where X is the largest norm-2 distance from the object's center of mass to its outer surface.

The GWS is computed from the set of contacts (C) found between the object and the fixture. All the contact forces are assumed to be unit forces at the point of contact. The set of all the wrenches computed are then encompassed in a convex hull. To get a final measure of the wrench resistance of the fixture, the volume of the convex hull of the GWS is used. The volume is scaled to a value in range $[0, 1]$ using the knowledge of the GWS being a unit 6-sphere. The wrench score (q_w) is calculated according to (4).

$$q_w = \frac{6 \cdot \text{vol}(\text{chull}(\text{wrenches}(C)))}{\pi^3} \quad (4)$$

where $\text{wrenches}()$ calculates the friction cone and associated torque, $\text{chull}()$ computes the convex hull of the 6D wrenches and $\text{vol}()$ computes the volume of the given convex hull.

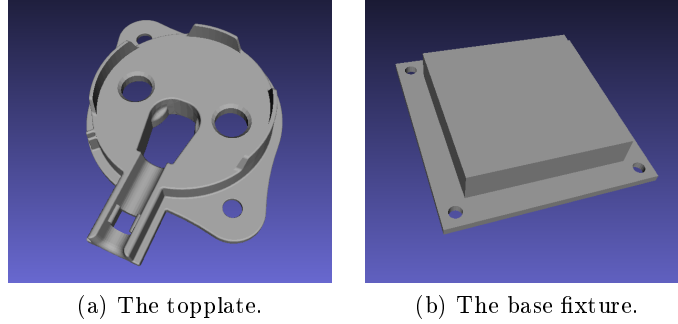


Fig. 7: Objects used for the fixture design.

4 Results

The use case object is the so-called “topplate”, see Fig. 7a, used by the company LogicData [9]. In the assembly of a drive, see Fig. 1, the topplate is fitted on top of the drive by pushing the two parts together. To do this, the topplate has to be placed in a fixture that holds the object steady while the drive is pressed on top of the topplate.

In the assembly process, it is important that the robot is able to place the topplate into the fixture. The placement operation has to be successful despite uncertainties in the location of the topplate when the robot places it into the fixture.

4.1 Fixture design & optimization

Following the fixture design pipeline shown in Fig. 5, the workcell is first created, see Fig. 8a. Then a set of drop poses were generated using the stochastic perturbation of 100 drop poses with a maximum linear and angular displacement of 10 mm and 15 degrees. The drop poses were scaled along the z-axis with 0.5 as to have a maximum linear translation of 5 mm along the z-axis. The generated drop poses are illustrated in Fig. 6b, where the black lines at the end of the blue lines represent the drop pose.

The fixture parametrization was defined using the base fixture, in which the imprint is made, as the object seen in Fig. 7b. Because of the nature of the assembly task, a slightly modified object is used for the imprint compared to the simulations. The object used is shown in Fig. 8b. Comparing the two Fig. 7a and 8b, it can be seen that the three larger holes were covered up. This was done because during the assembly of the two objects, the three holes will be occupied by the drive pushed onto the object.

The parametrization utilizes the imprint strategy optimizing the parameters seen in Tab. 1, where t_x and t_y are the tolerances of the profile along the two axes, d the depth of the object in the fixture and b the profile modifier.

Tab. 1 also shows the *Optimization* with the initial guess and optimized value found in the process using the Bound Optimization BY Quadratic Approximation (BOBYQA) [12] optimization scheme. A drop was considered *successful* if

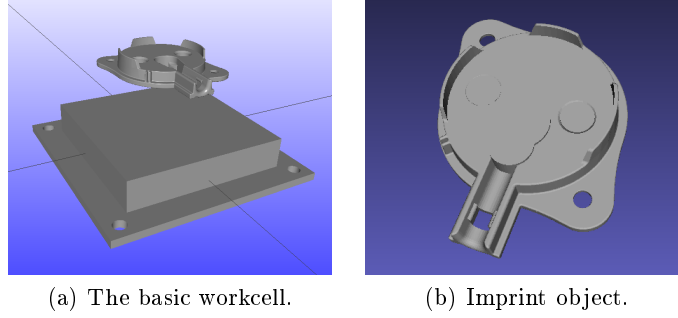


Fig. 8: Visualization of the workcell and object used for the imprint.

Table 1: Parameters, bounds and result of the optimization.

Param.	Min.	Max.	Optimization	Unit
t_x	0.0	100.0	20.0 \rightarrow 16.5	mm
t_x	0.0	100.0	20.0 \rightarrow 15.5	mm
b	0.2	1.0	0.50 \rightarrow 0.44	-
d	-10.0	10.0	0.0 \rightarrow 2.8	mm

the maximum linear and angular translation from its final goal is within 3.33 mm and 3.33 degrees respectively. The optimization weights were set to 1.0 for both the alignment and wrench metric. The complete setup time including deciding on reasonable bounds on the optimization took less than one hour. How the performance of the fixture changed in the course of the optimization is shown in Fig. 9. The score went from 0.20 to 0.32 over the course of the 50 steps taken by the BOBYQA optimizer. The optimization took 48.9 hours of computation time on a Intel Core i7-3610QM CPU 2.30GHz with 8GB RAM running dual threaded.

Fig. 10 illustrates the final fixture found in the process. The model of the topplate used contains close to 36,000 faces making it a time consuming simulation involving many colliding faces to compute, hence the long run time. The sudden jumps of the score in Fig. 9 are caused by the optimization algorithm at first taking a set of random sample points to explore the four dimensional space of the optimization. After the first initial steps the algorithm then explores the space around the highest scoring point in the optimization space, hence the long sequence of scores close to the optimum score. In hindsight, the optimization process should also have been stopped earlier after it peaked, since the BOBYQA algorithm had been set to a too high resolution. Setting this requirement more adequately could easily have halved the computation time.

The fixture's performance was verified in simulations by densely sampling along all six axes of the system. The base drop pose is seen in Fig. 6a and is 28 mm above the fixture. Regular sampling was used in the region of interest, predetermined in prior experiments. The results of the simulations are illustrated

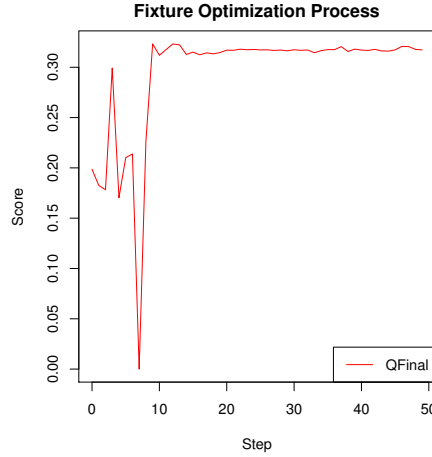


Fig. 9: Fixture score during the optimization process.

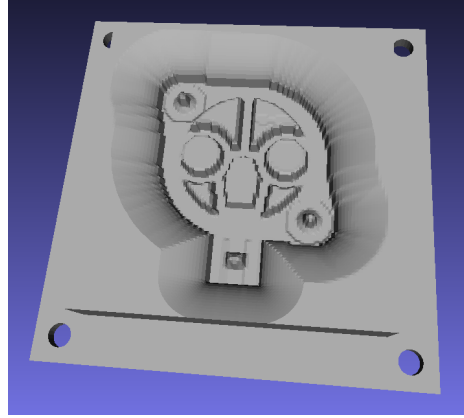
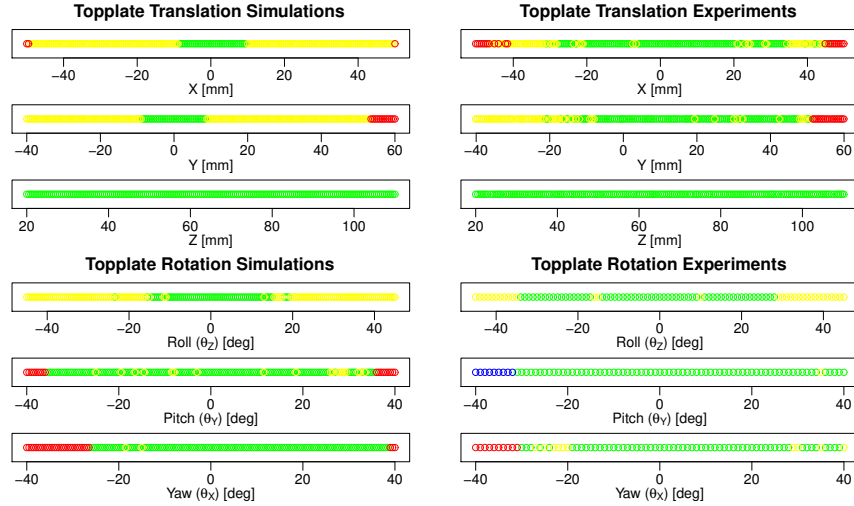


Fig. 10: Final fixture found during the optimization.

in Fig. 11a. The experiments were determined automatically in the simulation and classified as follows: *Successful* drops are when the object reaches its final pose with less than 3.33 mm of translational and 3.33 degrees of rotational offset. *Failure* is if the object's drop pose is starting in a collision, if it falls outside the workcell or the simulator fails. And *misaligned* is when the object does not make it in the criteria of successful or failure. Fig. 11a shows *successes* with green, *misalignment* in yellow and *failures* in red.

As seen in Fig. 11a, the translations along the y- and x-axis are limited to a success range of roughly ± 8.5 mm. This is because the topplate after impacting on the fixture has its movements quickly damped, and does therefore not have the required momentum to slide into the fixture. The success range is therefore relative small. The failures on the y-axis around 55 to 60 mm are because the object falls out of the workcell. Considering the rotations around roll, then the same is experienced as for the x- and y-axis. The failures at the extremes of the



(a) Simulated experiments.

(b) Real-world experiments.

Fig. 11: Results of the drop experiments using the topplate. Green, yellow, red and blue indicating success, misalignment, failure and out-of-range respectively.

pitch and yaw are because the object is in collision with the fixture in its starting pose, and the simulator can hence not start because of the object penetration.

The real-world tests were performed as in the simulations. The experiment was performed using the scene illustrated in Fig. 12 and conducted using a suction-cup to pick-up and drop the object.

The results of the real-world experiments are illustrated in Fig. 11b where *success* is depicted in green, *misalignment* with yellow, a *failure* in red and *out-of-range* as blue. The classification of the samples were determined using manual inspection. The objects drops where classified with *Successful*, *Misaligned* and *Failures* as for the simulated experiment. Furthermore the *Out-of-range* was used as classification when the robot was not able to go to the drop position because of collision or joint limits.

In general the object was found to move back and forth in the fixture after impacting with the fixture. This made the object, in most of the cases, jump into the fixture. Therefore, the alignment range of the fixture in the real-world experiments was quite big with roughly ± 20 mm on the x-axis. On the y-axis -7.4 to 47.5 mm, the low range in the negative direction was because the lower part of the object, seen in Fig. 7a, often stopped the object from sliding into the fixture. The failures on the far left of the yaw axis in Fig. 11b were due to collision between the two objects when the object reached its drop pose.

The alignment range of the fixture was summarized in Tab. 2. It can be clearly seen in Fig. 11a that the success range of the fixture is larger along the three axes x, y and θ_Y in the real-world experiments than in the simulations. Therefore, the simulations given the current settings provide a slightly pessimistic result

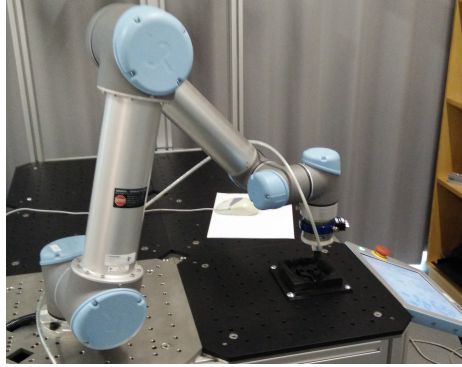


Fig. 12: Experimental setup.

Table 2: Success range of the fixture.

Axis	Simulation			Real-World		
	Min	Max	Range	Min	Max	Range
X [mm]	-8.5	9.5	18.0	-6.1	20.4	26.5
Y [mm]	-8.5	8.5	17.0	-7.4	18.9	26.3
Z [mm]	20.0	110.0	90.0	20.0	110.0	90.0
θ_Z [deg]	-10.5	12.5	23.0	-14.0	9.0	23.0
θ_Y [deg]	-2.5	11.0	13.5	-31.0	34.0	65.0
θ_X [deg]	-14.5	38.5	53.0	-19.0	28.0	47.0

of the fixtures performance compared to the real-life. This was largely found to be because the object in simulations was quickly damped when impacting the fixture, while in the real-world experiments their collisions were more elastic. The larger preservation of the kinetic energy in the real-world experiments hence made the object move around in the fixture and successfully reach the intended position.

5 Conclusion

A new method to design and optimize the cut-outs for fixtures based on imprints has been introduced. The method creates a parameterized model of a fixture that is then optimized for a given task. The framework uses dynamic simulations to quantify the fixtures cut-out's performance. Therefore, the optimizations are able to take the task context, alignments- and wrench-properties of the cut-out into account.

A fixture was designed for an industrial object involved in a assembly task. The fixture design was optimized using dynamic simulations and the resulting fixture was tested in both simulations and real-world experiments.

Compared to previous work within fixture design then this approach focuses on a design with no movable parts such as clamps and locators. The design can also easily be 3D printed allowing it to be quickly produced and taken into use.

Furthermore the framework developed is easy to use and requires little input from the user, which makes it easily usable by non-expert users.

References

1. Boyle, I.M., Rong, K., Brown, D.C.: Cafixd: a case-based reasoning fixture design method. framework and indexing mechanisms (2004)
2. Brost, R.C., Peters, R.R.: Automatic design of 3-d fixtures and assembly pallets. *The International Journal of Robotics Research* **17**(12), 1243–1281 (1998)
3. Causey, G.C., Quinn, R.D.: Gripper design guidelines for modular manufacturing. In: *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*. vol. 2, pp. 1453–1458 vol.2 (5 1998). <https://doi.org/10.1109/ROBOT.1998.677309>
4. Ellekilde, L.P., Petersen, H.G.: Design and test of object aligning grippers for industrial applications. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 5165–5170 (2006). <https://doi.org/10.1109/IROS.2006.281652>
5. Ellekilde, L.P., Jorgensen, J.A.: Robwork: A flexible toolbox for robotics research and education. *Robotics (ISR)*, 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK) pp. 1 – 7 (6 2010)
6. Hashemi, H., Shaharoun, A.M., Sudin, I.: A case-based reasoning approach for design of machining fixture. *The International Journal of Advanced Manufacturing Technology* **74**(1), 113–124 (2014). <https://doi.org/10.1007/s00170-014-5930-4>, <http://dx.doi.org/10.1007/s00170-014-5930-4>
7. Jørgensen, J.A., Ellekilde, L.P., Petersen, H.G.: RobWorkSim - an Open Simulator for Sensor based Grasping. In: *ISR/ROBOTIK 2010 - ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. VDE-Verlag (Jun 2010), <http://www.vde-verlag.de/proceedings-en/453273198.html>
8. Kang, Y., Rong, Y., Yang, J., Ma, W.: Computer-aided fixture design verification. *Assembly Automation* **22**(4), 350–359 (2002). <https://doi.org/10.1108/01445150210446229>, <http://dx.doi.org/10.1108/01445150210446229>
9. LogicData: Logicdata, <http://www.logicdata.net/>
10. Pelinescu, D.M., Wang, M.Y.: Multi-objective optimal fixture layout design. *Robotics and Computer-Integrated Manufacturing* **18**(5–6), 365 – 372 (2002). [https://doi.org/http://dx.doi.org/10.1016/S0736-5845\(02\)00027-3](https://doi.org/http://dx.doi.org/10.1016/S0736-5845(02)00027-3), <http://www.sciencedirect.com/science/article/pii/S0736584502000273>
11. Pollard, N.S.: Parallel methods for synthesizing whole-hand grasps from generalized prototypes (1994)
12. Powell, M.J.D.: The BOBYQA algorithm for bound constrained optimization without derivatives (8 2009)
13. Schwartz, L.C.M.W., Ellekilde, L.P., Krüger, N.: Design of robot grippers and fixtures for robotic manufacturing
14. Schwartz, L.C.M.W., Wolniakowski, A., Werner, A., Ellekilde, L.P., Krüger, N.: Designing fingers in simulation based on imprints. In: *Proceedings of the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications - Volume 1: SIMULTECH.*, pp. 304–313. INSTICC, SciTePress (2017). <https://doi.org/10.5220/0006441003040313>
15. Smith, R.: Open dynamics engine (2008), <http://www.ode.org/>, <http://www.ode.org/>

16. Wang, H., Rong, Y.K., Li, H., Shaun, P.: Computer aided fixture design: Recent research and trends. *Comput. Aided Des.* **42**(12), 1085–1094 (Dec 2010). <https://doi.org/10.1016/j.cad.2010.07.003>, <http://dx.doi.org/10.1016/j.cad.2010.07.003>
17. Wolniakowski, A., Miatliuk, K., Gosiewski, Z., Bodenhagen, L., Petersen, H.G., Schwartz, L.C.M.W., Jørgensen, J.A., Ellekilde, L.P., Krüger, N.: Task and context sensitive gripper design learning using dynamic grasp simulation. *Journal of Intelligent & Robotic Systems* pp. 1–28 (2017). <https://doi.org/10.1007/s10846-017-0492-y>, <http://dx.doi.org/10.1007/s10846-017-0492-y>
18. Yi, C., Bekey, G.A.: Assembly planning for modular fixtures. In: *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on.* vol. 2, pp. 704–711 vol.2 (11 1996). <https://doi.org/10.1109/IROS.1996.571040>
19. Zieliński, C., Kasprzak, W., Kornuta, T., Szyrkiewicz, W., Trojanek, P., Walęcki, M., Winiarski, T., Zielińska, T.: Control and programming of a multi-robot-based reconfigurable fixture. *Industrial Robot: An International Journal* **40**(4), 329–336 (2013). <https://doi.org/10.1108/01439911311320831>, <http://dx.doi.org/10.1108/01439911311320831>