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

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

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 parametrized 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. The 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.

Index Terms—Simulation, Fixture Learning, Robotics, Optimizations, Assembly.

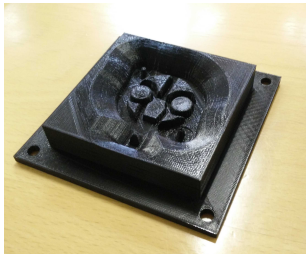
1 Introduction

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 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.

(a) Disassembled set-up.

(b) The assembled parts.



(c) A 3D printed fixture.

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

• *L.C.M.W. Schwartz, L.-P. Ellekilde and N. Krüger are with the The Maersk McKinney-Moller Institute, University of Southern Denmark, Odense, Denmark.*
E-mail: norbert@mmmi.sdu.dk

This work has received funding from the EU project ReconCell (ICT Innovation for Manufacturing SMEs (I4MS), project number 680431)

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 [1], [2]. 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 optimisation 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 [2].

The idea of using optimisations in simulations was inspired by the approach presented in [3] for gripper design, where a parametrised finger model is optimised to get the best suited gripper for a specific task. Unlike in [3], the process is significantly automated by introducing an automated parametrisation method. This method, while applied here for fixtures, was also generalised for gripper design [4]. The method is covered by the patent [5].

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 [2]. The locators are static and mounted against the object fixed in the fixture system. The layout of the locator is of varying shapes and sizes, some intended for a specific object and others more general [6].

The early work in this field has focused on the automation of the design [6] and planning process and the analysis of the fixture [7]. [8] 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 [9].

Tools have been developed to optimize a fixture layout by setting up a set of scoring systems to evaluate the fixture [10]. 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 [11], [12]. 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 [3], [4]. 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 therefore 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 bigger

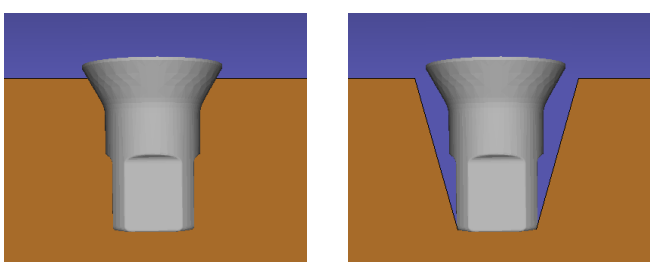
framework to design an optimised fixture for a specific object.

3.1 Fixture Imprint Parametrization

When designing fixtures for specific objects, a cut-out in a basic shape of a fixture is usually made [13]. 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 visualised in Fig. 2a. This generally results in low alignment capabilities 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.



(a) Pure imprint. (b) Imprint after applying post-processing.

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

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).

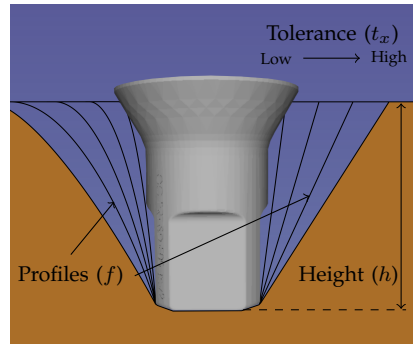


Fig. 3: Visualisation of parameters in the design of the cut-out. Profiles defining the shape and tolerance the width of the profile. Left and right two different profiles are shown.

The cut-out can be applied with any user specified function (*profile*) taking into account the criteria mentioned before. In this project, the profile is 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.

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 (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 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 $\{-n_x, ..., -1, 0, 1, ..., n_x\}$ and k is the set $\{-n_y, ..., -1, 0, 1, ..., n_y\}$. 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 $x_l \in \{1, ..., resx\}$ and $y_m \in \{1, ..., resy\}$ where $resx$ and $resy$ is the size of the discretised imprint.

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

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 is used to create the cut-out as a tri-mesh from the fixture.

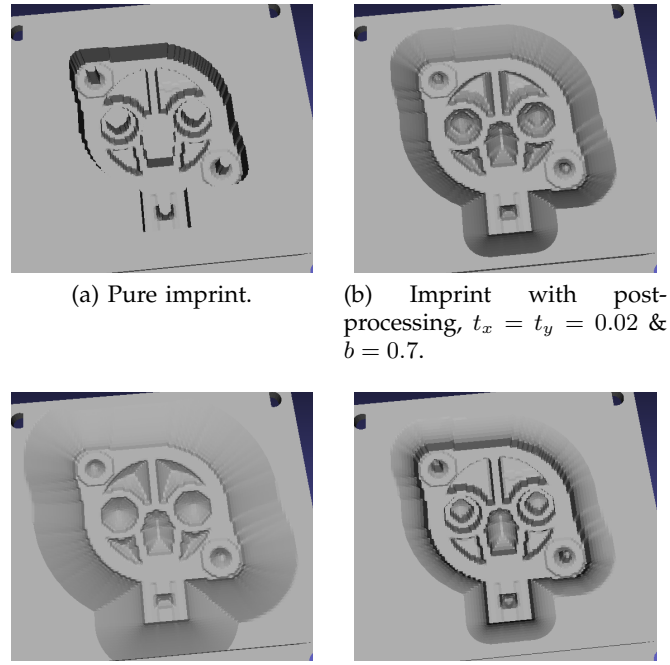
This results in a parametrized fixture requiring only the object and three parameters, the profile and two tolerances, to be specified. Fig. 4 illustrates how the fixture design varies for the given parameters using the profile function $f^b(a) = a^b$. First Fig. 4a shows the fixture when no post processing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

Given this parametrized 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 utilises the *RobWork* library [14], [15].

It also provides tools to evaluate a fixture's 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.



(a) Pure imprint.

(b) Imprint with post-processing, $t_x = t_y = 0.02$ & $b = 0.7$.

(c) Imprint with post-processing, $t_x = t_y = 0.03$ & $b = 0.7$.

(d) Imprint with post-processing, $t_x = t_y = 0.02$ & $b = 0.5$.

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

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 different version of the fixture.

Secondly a set of drop poses are generated (Fig. 5b) from which the test object is released when put into the fixture. The drop poses can be generated using either stochastic or regular sampling.

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 pro-

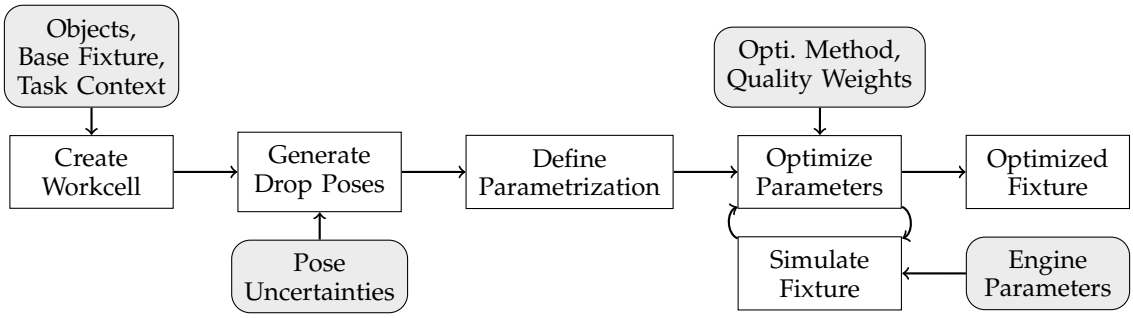


Fig. 5: The fixture design pipeline. Grey representing inputs and in white the steps the user and program goes through.

cess the best performing fixture found in the process is returned to the user (Fig. 5f).

3.2.1 Fixture Evaluation

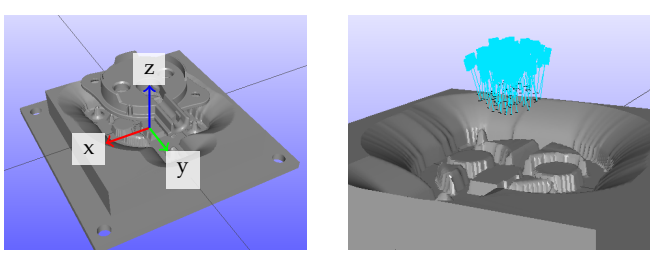
In order to optimize the fixture a set of quality scores are needed in order to quantify and compare the performance of the produced fixtures. The scores are normalized in a manner such that they are in the range of $[0, 1]$.

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

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

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 emphasised in the optimisation process.

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



(a) The base drop pose. (b) Drop poses used.

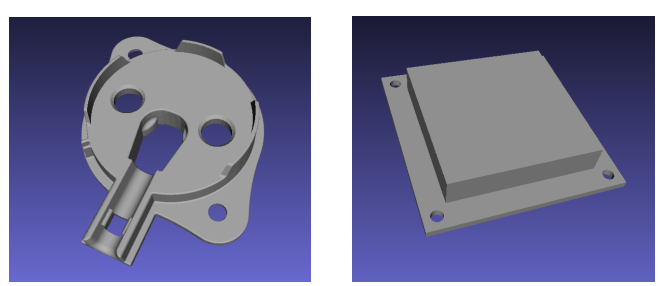
Fig. 6: Visualisation of object drop poses.

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.

Evaluating Fixture Wrench: To evaluate the wrench space of the fixture the Grasp Wrench Space (GWS) [16] 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 centre of mass to its outer surface.

4 Results

The use case object is the so-called "topplate", see Fig. 7a, used by the company LogicData [17]. 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.



(a) The topplate. (b) The base fixture.

Fig. 7: Objects used for the fixture design.

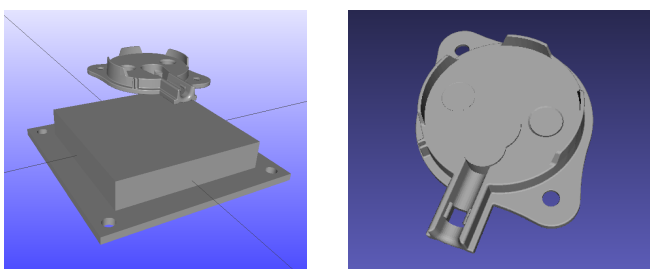
In the assembly process, it is important that the robot is able to place the topplate into

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[1].



(a) The basic workcell. (b) Imprint object.

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

The parametrization utilises 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, $height$ 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

1. 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.

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	-
$height$	-10.0	10.0	0.0 $\rightarrow$ 2.8	mm

Quadratic Approximation (BOBYQA) [18] optimization scheme. 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. 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. 9 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.

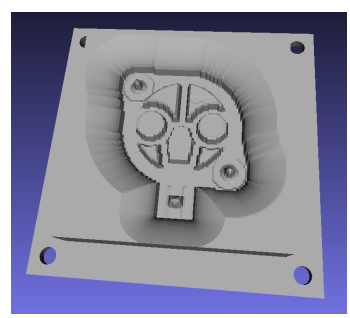


Fig. 9: Final fixture found during the optimization.

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 in Fig. 10a. 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. 10a shows *successes* with green, *misalignments* in yellow and *failures* in red.

The real-world tests were performed as in the simulations. The experiment was performed using the scene illustrated in Fig. 11 and conducted using a suction-cup to pick-up and drop the object.

The results of the real-world experiments are illustrated in Fig. 10b 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.

It can be clearly seen in Fig. 10a 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 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 was here introduced to design and optimize the cut-outs for fixtures based on imprints. The method creates a parametrized 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 fixture focuses on assembly tasks. The framework developed here is easy to use and requires little input from the user, which makes it easily usable by non-expert users.

References

[1] G. C. Causey and R. D. Quinn, "Gripper design guidelines for modular manufacturing," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 2, 5 1998, pp. 1453–1458 vol.2.

[2] H. Wang, Y. K. Rong, H. Li, and P. Shaun, "Computer aided fixture design: Recent research and trends," *Comput. Aided Des.*, vol. 42, no. 12, pp. 1085–1094, Dec. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.cad.2010.07.003

[3] A. Wolniakowski, K. Miatliuk, Z. Gosiewski, L. Bodenhagen, H. G. Petersen, L. C. M. W. Schwartz, J. A. Jørgensen, L.-P. Ellekilde, and N. Krüger, "Task and context sensitive gripper design learning using dynamic grasp simulation," *Journal of Intelligent & Robotic Systems*, pp. 1–28, 2017. [Online]. Available: http://dx.doi.org/10.1007/s10846-017-0492-y

[4] L. C. M. W. Schwartz, A. Wolniakowski, A. Werner, L.-P. Ellekilde, and N. Krüger, "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,*, INSTICC. SciTePress, 2017, pp. 304–313.

[5] L. C. M. W. Schwartz, L.-P. Ellekilde, and N. Krüger, "Design of robot grippers and fixtures for robotic manufacturing," Patent 17 196 424.0-1712.

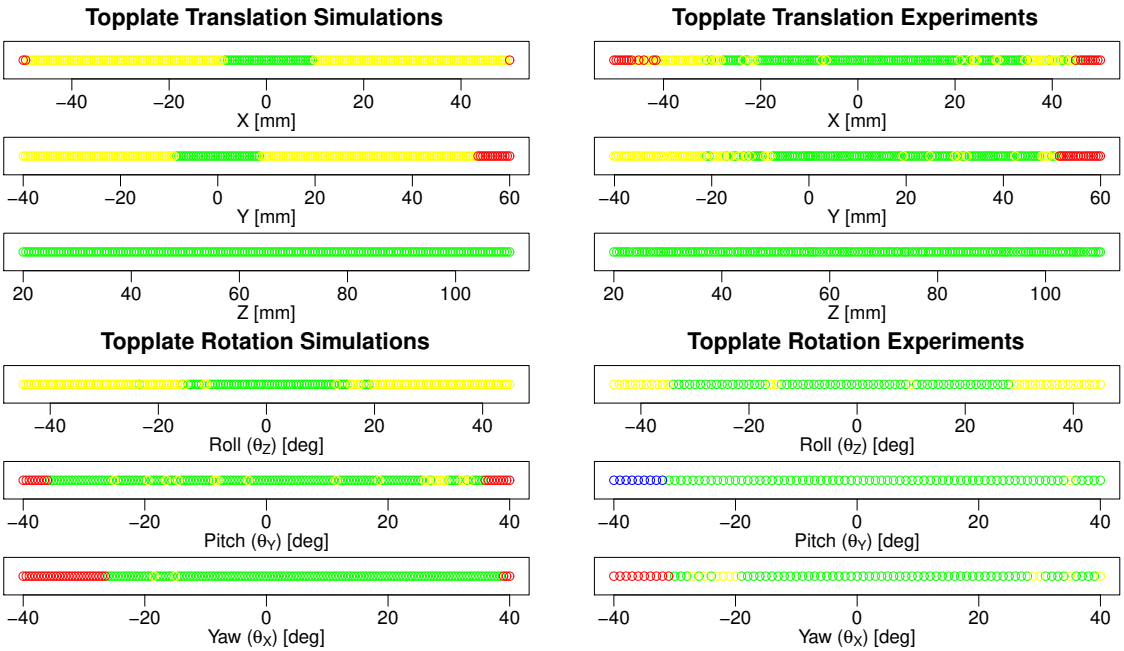
[6] R. C. Brost and R. R. Peters, "Automatic design of 3-d fixtures and assembly pallets," *The International Journal of Robotics Research*, vol. 17, no. 12, pp. 1243–1281, 1998.

[7] D. M. Pelinescu and M. Y. Wang, "Multi-objective optimal fixture layout design," *Robotics and Computer-Integrated Manufacturing*, vol. 18, no. 56, pp. 365 – 372, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736584502000273

[8] C. Yi and G. A. Bekey, "Assembly planning for modular fixtures," in *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, vol. 2, 11 1996, pp. 704–711 vol.2.

[9] C. Zieliski, W. Kasprzak, T. Kornuta, W. Szynkiewicz, P. Trojanek, M. Walcki, T. Winiarski, and T. Zieliska, "Control and programming of a multirobotbased reconfigurable fixture," *Industrial Robot: An International Journal*, vol. 40, no. 4, pp. 329–336, 2013. [Online]. Available: http://dx.doi.org/10.1108/01439911311320831

[10] Y. Kang, Y. Rong, J. Yang, and W. Ma, "Computeraided fixture design verification," *Assembly Automation*, vol. 22, no. 4, pp. 350–359, 2002. [Online]. Available: http://dx.doi.org/10.1108/01445150210446229



(a) Simulated experiments.

(b) Real-world experiments.

Fig. 10: Results of the drop experiments using the topplate. Green, yellow, red and blue indicating success, misalignment, failure and out-of-range respectively.

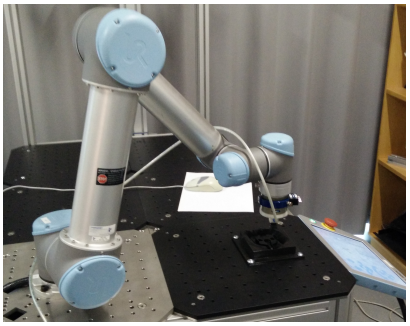


Fig. 11: Experimental setup.

[11] H. Hashemi, A. M. Shaharoun, and I. Sudin, "A case-based reasoning approach for design of machining fixture," *The International Journal of Advanced Manufacturing Technology*, vol. 74, no. 1, pp. 113–124, 2014. [Online]. Available: http://dx.doi.org/10.1007/s00170-014-5930-4

[12] I. M. Boyle, K. Rong, and D. C. Brown, "Cafixd: a case-based reasoning fixture design method. framework and indexing mechanisms," 2004.

[13] L. P. Ellekilde and H. G. Petersen, "Design and test of object aligning grippers for industrial applications," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5165–5170.

[14] L.-P. Ellekilde and J. A. Jorgensen, "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.

[15] J. A. Jørgensen, L.-P. Ellekilde, and H. G. Petersen, "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. [Online]. Available: http://www.vde-verlag.de/proceedings-en/453273198.html

[16] N. S. Pollard, "Parallel methods for synthesizing whole-hand grasps from generalized prototypes," 1994.

[17] LogicData. Logicdata. [Online]. Available: http://www.logicdata.net/

[18] M. J. D. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives," 8 2009.