Journal of Intelligent & Robotic Systems (2021) 101: 49
https://doi.org/10.1007/s10846-020-01303-z

Towards Autonomous Robotic Assembly: Using Combined Visual and Tactile Sensing for Adaptive Task Execution

Korbinian Nottensteiner[1] · **Arne Sachtler**[2] · **Alin Albu-Schäffer**[1,2]

Received: 1 July 2020 / Accepted: 21 December 2020 / Published online: 20 February 2021

Abstract

Robotic assembly tasks are typically implemented in static settings in which parts are kept at fixed locations by making use of part holders. Very few works deal with the problem of moving parts in industrial assembly applications. However, having autonomous robots that are able to execute assembly tasks in dynamic environments could lead to more flexible facilities with reduced implementation efforts for individual products. In this paper, we present a general approach towards autonomous robotic assembly that combines visual and intrinsic tactile sensing to continuously track parts within a single Bayesian framework. Based on this, it is possible to implement object-centric assembly skills that are guided by the estimated poses of the parts, including cases where occlusions block the vision system. In particular, we investigate the application of this approach for peg-in-hole assembly. A tilt-and-align strategy is implemented using a Cartesian impedance controller, and combined with an adaptive path executor. Experimental results with multiple part combinations are provided and analyzed in detail.

Keywords Autonomous assembly · Sequential Monte Carlo · Compliant manipulation · Sensor fusion · Peg-in-hole · Future manufacturing

1 Introduction

The growing individualization of products demands facilities that can manufacture small batch sizes with little effort. Autonomous robots can help increase the required flexibility. At the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR), we are developing an autonomous robotic assembly system for flexible manufacturing (see Fig. 1). It is capable of assembling unique products with parts from an aluminum profile construction set [52]. Assembly sequencing at task level is performed automatically using multiple abstraction levels [56]. Furthermore, a reliable task execution is required for similar but different product variants. For this purpose, we implemented robust and reusable robotic skills using compliant control methods of the lightweight robot technology [1]. However, high-level feedback is only incorporated in specific situations where logic decisions are required, and geometric uncertainties are only passively compensated for during execution. In order to increase the level of autonomy, we need an adaptive task execution that actively reacts to the current state of the objects in the robotic cell.

Compared to the previous version of the system with only a single robotic arm [52], we removed all part holders to increase flexibility with respect to product types. At the same time, this step introduced significant uncertainties in object poses. However, a successful execution is still possible if the initial state is well defined.[1] In our recent work on combined visual and touch-based registration [57], we show how static objects in the robotic arm workspace can be localized autonomously at high precision. This reduces the need for manual calibration efforts and poses of objects can be initially registered automatically; any remaining uncertainties can subsequently be compensated for with passive alignment and blind-search strategies. Nevertheless, our system currently fails if parts unexpectedly move during the assembly process. Furthermore, the fact that

✉ Korbinian Nottensteiner
korbinian.nottensteiner@dlr.de

[1] German Aerospace Center (DLR), Institute of Robotics and Mechatronics (RM), Münchener Str. 20, 82234 Weßling, Germany

[2] Technical University of Munich (TUM), Department of Informatics, Chair of Sensor Based Robots and Intelligent Assistance System, Boltzmannstr. 3, 85748 Garching, Germany

[1] See https://youtu.be/XQhXGJbUURE

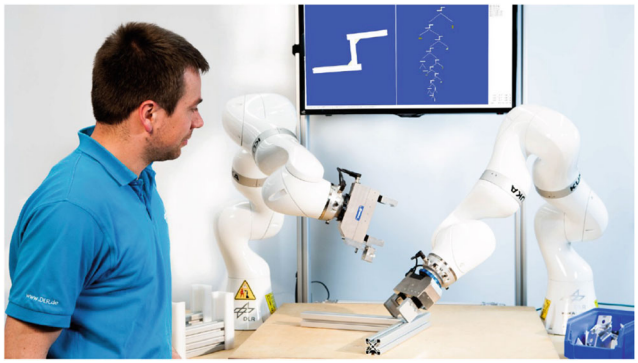


Fig. 1 Autonomous assembly of aluminum profile structures with a dual-arm robotic system without specialized holders

robots often occlude the field of view of cameras motivates us to investigate tactile sensing in the case of moving parts.

Consequently, in this work, we present how robotic skills can adapt according to the observed contact situation. In particular, we are looking into the classical peg-in-hole task in which the hole is moving with an unknown motion. Numerous approaches for peg-in-hole exist [44, 74] and Section 2 provides an overview, but only a few papers deal with moving parts. An example is provided by Jörg et al. [34], who demonstrate the insertion of a piston using visual servoing in combination with a force controller; similar solutions were also investigated for automated wheel assembly on conveyor belts, e.g., [14, 38]. Nevertheless, the existing solutions typically require a fine position estimate from the vision system and do not explicitly localize the parts with tactile measurements. In contrast, we present a general approach that combines visual and tactile sensing and continuously tracks the parts in an integrated framework. Therefore, we extend our previous works [54, 57] based on intrinsic tactile sensing with an adaptive motion generation component and combine both in an adaptive assembly skill. We provide a brief overview of the system in Section 3, and present the details of the approaches for state estimation in Section 4 and motion generation in Section 5. Experimental results are presented and discussed in Section 6.

2 Background and Related Work

In the field of assembly automation, peg-in-hole is considered an important benchmark. The main challenge is the transition of a part from free space into a highly constrained target pose. During the insertion, tight tolerances in combination with positioning errors can lead to undesired effects such as jamming [61]. It was concluded early that only compliant motions can solve this issue [29, 45]. For this purpose, passive compliant tools [21, 71] and control methods with force feedback were developed [43]. Doing this soon showed that automated insertion of parts with clearances down to the scale of microns is technically feasible [24]. Today, the challenges have shifted from solving the pure physical task to aspects that concern the reduction of implementation efforts and the increase of reusablity in the presence of large uncertainties. In the following, we provide an overview about various classes of peg-in-hole approaches and current related work in this field.

2.1 Pre-defined Strategies and Offline Planning

Nearly 50 years ago, Inoue [29] described robust procedures, called "stereotype actions," for shaft-bearing assemblies. These make use of force feedback and well-arranged shift and tilt motions to reduce uncertainty in the parts locations. Since then, further approaches using pre-defined motion strategies have been developed. Bruyninckx et al. [11] describe a search strategy with a tilted peg and a kinematic model for the alignment motion. "Blind-search" strategies follow similar ideas and were applied with multiple variations, e.g., for transmission gear assembly [50] inserting a plug for charging an electric car [33]. A systematic search to cover the uncertain region in combination with a tilt strategy is presented in [16]. Nevertheless, disadvantages to those search strategies are the time spent exploring the contacts and that the strategy must be carefully selected in advance.

Consequently, specialized offline planners were developed to automatically find an appropriate sequence of fine motions that are extremly likely to reach a goal area [20, 22, 41]. Stemmer et al. [63] describe a method that analyzes the shape of complex planar parts and automatically generates a robust alignment motion. Recently, belief space planners were applied that aim at finding optimal and robust trajectories [72]. Furthermore, online optimization techniques are developed to tune pre-defined strategies automatically and outperform humans with respect to execution times [32]. Clearly, it is of a major advantage to apply a suitable strategy to reach high performance. Limitations of the pre-defined and offline-planned strategies are that they are often only applicable in a narrow scope, require prior knowledge of the task and that online data is not always incorporated. This becomes especially important when objects are not fixed, but can move within the environment. In this work, we also apply a pre-defined tilt strategy and will show how it makes use of visual and tactile feedback to track moving parts.

2.2 Human Demonstrations and Learning

Modeled strategies are often inspired by human manipulation strategies. A shortcut to directly implement human

strategies is programming by demonstration. Hirzinger showed early on how force-torque sensors can be used to teach new tasks [27]. For specific situation, these types of methods provide quick solutions and are nowadays the default teach-in technique for so-called "cobots". Nevertheless, it is difficult to generalize over multiple tasks, and trajectories are usually not reusable. Recent works in the field of kinesthetic teaching and imitation learning try to generalize demonstrations, e.g., [19, 37, 59]. Those methods might be important in the future for acquiring robotic skills. Right now, an open question is still how the demonstrations can be generalized efficiently and wheter they are also applicable for environments with moving parts. Multiple works also aim at enabling the robots to learn appropriate skills directly based on experience without human intervention. For example, Simons et al. [60] implement a self-learning controller mapping force to corrective motions; neural networks and reinforcement learning methods were also applied for learning compliant controllers, e.g., in [5, 25]. Recently, new approaches using deep learning and unsupervised learning for solving peg-in-hole were published [30, 39, 42]. The latest advances show promising results. However, the approaches still depend heavily on the amount and quality of training data for specific use cases.

2.3 Bayesian State Estimation

The novel machine learning approaches are sometimes criticized for the limited explainability of the mapping between inputs and outputs. In contrast, approaches based on Bayesian probability theory provide interpretable models for tracking of uncertainties. Besides classical methods in this field like Kalman Filters, particle filtering methods have gained more attention in robotics since the pioneering works of Thrun et al. [69]. They have been used not only for mobile robotics, but also in the field of assembly. Nguyen et al. [51] present a framework for tracking pose uncertainties with vision and tactile data. The uncertainty information is used to adapt an elliptical spiral search pattern for peg-in-hole with static parts. Wirnshofer et al. [73] present Bayesian state estimation in multiple scenarios including peg-in-hole, but do not make use of force measurements in the probability update. Force measurements enable robots to distinguish contact states and keep a controlled contact. Meeussen et al. [47, 48] implement a particle filter for contact state detection and show how to use it for estimating geometric uncertainties and executing compliant motions. Multiple works estimate geometric uncertainties with particle filters and force measurements in peg-in-hole assembly [4, 15, 54, 65, 68], but all of them consider a fixed and rigid hole pose during the assembly. In this work, we will extend our previous works in this field [54, 57] for moving parts and suggest an adaptive motion generation procedure for the execution of assembly skills.

3 Autonomous Robotic Assembly Framework

Increasing the level of autonomy requires systems that execute goal-directed actions while considering the currently observed world state. In this section, we describe components of such an autonomous robotic assembly system, explain the concept of robotic skills, and introduce Bayesian methods used for state estimation and motion generation in the implementation of an adaptive assembly skill.

3.1 Components of the Autonomous Assembly System

The considered assembly system is composed of a task planning unit, a knowledge base, a scheduler and a collection of robotic skills (see Fig. 2). A task typically represents the specification of one one step necessary for assembly. A skill is defined here as a robotic behavior that robotic behavior that reaches desired goal states in multiple situations and under varying conditions (see Section 3.2). The deliberative task planning unit selects robotic skills, which are in principle capable of solving the tasks under the constraints that arise from the goal specification and the assumed world state. For this, we are using a sequence planner that automatically decomposes the assembly of a desired product into a sequence of tasks and selects using representations of the parts and the system on multiple abstraction levels [56]. The knowledge base provides information about properties of objects and grounds them in physical quantities as far as possible. States can be defined based on the object entities in the knowledge base. A central runtime component keeps track of the overall world state of all objects [40]. The skill executor schedules robotic skills in compliance with the present world state and orchestrates the execution at runtime.

3.2 Robotic Skills

As stated above, our assembly system makes use of the concept of robotic skills, which is known from various related works [7, 8, 52, 62, 67] with comparable definitions. In contrast to traditional implementations of robotic programs in the industry, which blindly follow pre-programmed paths and routines, robotic skills adapt to the current situation by observing the execution and changes in the state of the world. Furthermore, they are formulated object-centric to be efficiently reuseable in various situations. The

interested reader might also like to compare the robotic skills with the philosophical view on agents' abilities and is referred to [31]. As depicted in Fig. 2, we suggest that the implementation of a robotic skill for assembly might be composed of a feature detector, a state estimator, a component for motion generation and finally a robot controller.

The feature detector recognizes the presence of features of physical objects. In our case, we assume that CAD data and semantic descriptions of the geometry of the objects and their features are available through the central knowledge base. The features then provide state variables, which can be tracked by a state estimator. The estimator fuses all information about detected features and measurements in order to estimate the states relevant for skill execution, e.g., the relative pose between two parts. The motion generator is a component that generates motion commands based on the comparison of estimated and desired states of the features. In combination with the state estimator, the motion generator can realize reactive and sensor-guided motions. The robot controller abstracts the robotic hardware and provides interfaces to execute motion commands, such as motion primitives to execute impedance-controlled trajectories.

3.3 State Estimation and Motion Generation

We model the tracking of features as a recursive Bayesian estimation problem, where features are represented as states of a hidden stochastic process. The states can contain pose and shape information. We denote the state vector at time $t = t_k$ by $\boldsymbol{x}_k \in \mathbb{R}^n$ and furthermore assume that it is not directly observable. Instead, observations from dedicated feature detectors are collected in a measurement vector $\boldsymbol{y}_k \in \mathbb{R}^m$. Then, the objective is to then estimate the current state up to time t_k given all past measurements denoted by the probability density function $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$. Bayesian estimation provides recursive methods to solve this probabilistic inference task. Each cycle involves two steps: (1) predicting $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$ and (2) updating $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$, where the distribution is updated using the measurement likelihood $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ and the relation $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \propto p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$.

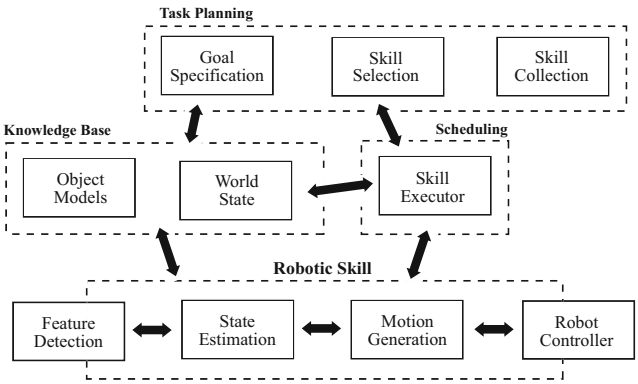


Fig. 2 Components of a robotic system for autonomous planning and adaptive execution of assembly tasks

In this work, the Bayesian state estimator is implemented in the form of a sequential Monte Carlo (SMC) algorithm [12], i.e., a particle filter. This approximates the distribution of the hidden state $\boldsymbol{x}$ using a set of weighted samples $\mathcal{X}_k = \{(W_k{}^{(\mathrm{i})}, \boldsymbol{x}_k{}^{(\mathrm{i})})\}$, where $W_k{}^{(\mathrm{i})} \in \mathbb{R}$ denotes a scalar weight and $\boldsymbol{x}_k{}^{(\mathrm{i})}$ a sample of the hidden state. The initial uncertainty at time $t = 0$ is represented by a set of N samples $\mathcal{X}_0 = \{(1/N, \boldsymbol{x}_0{}^{(1)}), \ldots, (1/N, \boldsymbol{x}_0{}^{(\mathrm{N})})\}$ drawn from the initial density $p(\boldsymbol{x}_0)$. Samples $\boldsymbol{x}_k{}^{(\mathrm{i})}$ are then repeatedly propagated with a process model $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ to get $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$, weighted by the measurement likelihood $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ and resampled according to the resulting distribution (see Fig. 3). After resampling, the weights are set to $W_k{}^{(\mathrm{i})} = 1/N$. Assuming normalized weights, statistical estimates, e.g., expected values $\hat{V}_k$ of a function $V(\boldsymbol{x}_k)$, can be approximated by the evaluation of the particle distribution [12]:

$$\hat{V}_k \approx \sum_{i=1}^{N} W_k^{(i)} V(\boldsymbol{x}_k^{(i)}). \tag{1}$$

The sample distribution represents the belief space over the feature states and can be used for motion generation. The motion generation component of the skill analyzes the distribution of samples and generates motion commands based on a policy (see Fig. 3), which can be computed in advance or online. This combination of state estimator and motion generator is comparable to a partially observable Markov decision process (POMDP) control architecture as described by Kaelbling and Lozano-Pérez [35]. In Section 4, we describe detailed models of the state estimator and in Section 5 we present how adaptive behavior can be implemented in the motion generation step.

4 State Estimation for Assembly

In this section, we provide a detailed view of the models used for the recursive Bayesian state estimation. First, the robot and uncertainty model, as well as the virtual contact model, are introduced, after which the computation of the tactile and the visual likelihood is presented. The section finishes with the update model.

4.1 Robot and Uncertainty Model

We consider manipulators with $n \geq 6$ rotational joints that are equipped with joint torque sensors. At each discrete time

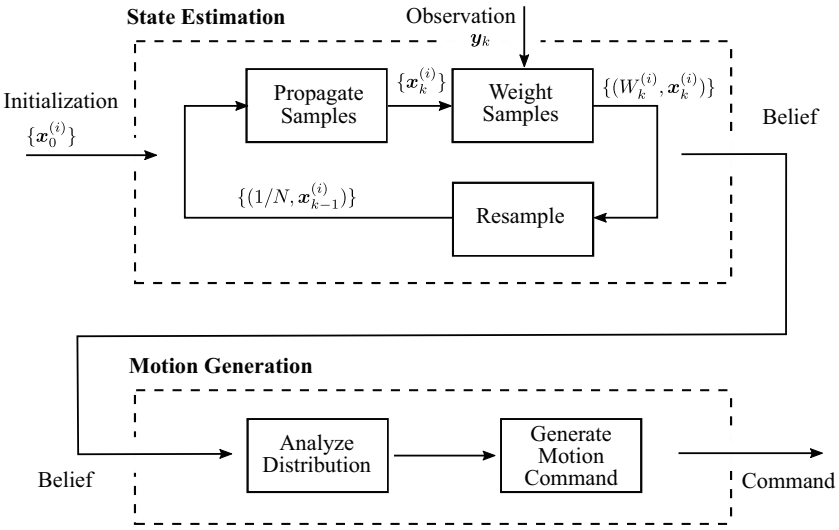


Fig. 3 State estimation and motion generation components of the assembly skill. Recursive Bayesian state estimation provides the belief state, which is then analyzed for generating motion commands

step k, the joint position $\boldsymbol{q}_k \in \mathbb{R}^n$ and the external joint torque $\boldsymbol{\tau}_k \in \mathbb{R}^n$ are measured. We assume that a peg with known geometry is grasped rigidly, i.e., does not slip inside the gripper. The grasp transformation is known and the forward kinematics can be computed from the joint position measurements. The homogeneous transformation $\boldsymbol{H}_{BD,k} = \boldsymbol{H}_{BD}(\boldsymbol{q}_k) \in SE(3)$ denotes the transformation from the robot base frame B to the reference frame D of the peg (see Fig. 4). The hole with frame C moves on an unknown path in the workspace of the manipulator. Thus, the pose of the hole is initially unknown, but is within the field of view of a vision system with frame V. In this work, we assume an eye-to-hand setting with a monocular camera at $\boldsymbol{H}_{BV} = \text{const.} \in SE(3)$. A dedicated feature detector provides measurements of the projected center points $(p_x, p_y)_k \in \mathbb{R}^2$ of the hole in the image plane.

In order to track the hole, we define the hidden state $\boldsymbol{x}_k = [x, \dot{x}, y, \dot{y}, z, \dot{z}]$, where $x, y, z \in \mathbb{R}$ are the Cartesian coordinates of the hole center with respect to a reference frame $\bar{C}$ and $\dot{x}, \dot{y}, \dot{z}$ denote the respective time derivatives. The true pose of the hole can be written as $\boldsymbol{H}_{BC}(x, y, z) = \boldsymbol{H}_{B\bar{C}}\boldsymbol{H}_{\bar{C}C}(x, y, z)$. The given task is to transfer the peg from a start frame to a desired target frame T specified with respect to the hole at a known location $\boldsymbol{H}_{CT} = \text{const.} \in SE(3)$. We define D to be located at the bottom of the peg, and T at the bottom of the hole.

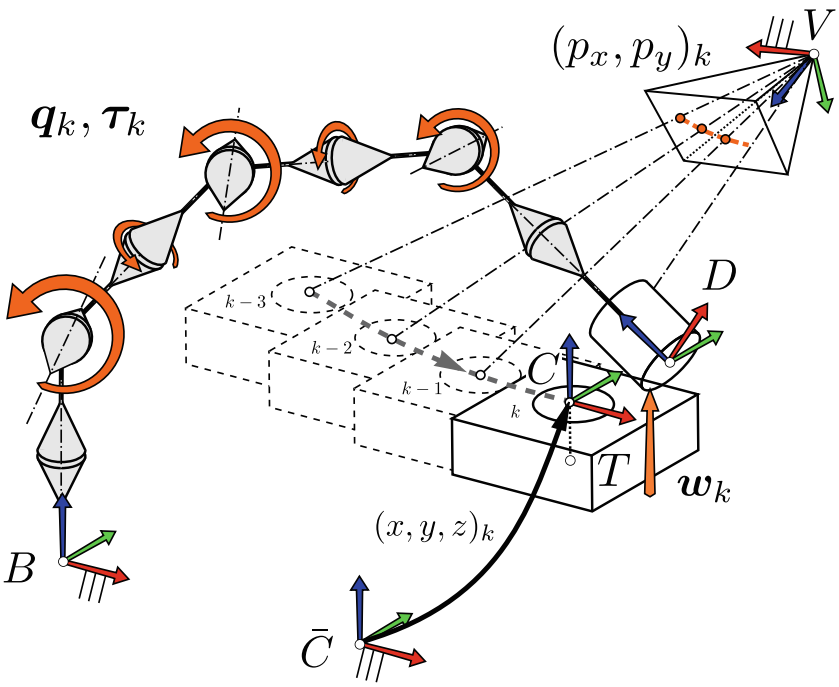


Fig. 4 Definition of frames and variables in the considered scenario. A peg with reference frame D is rigidly attached to a manipulator with base frame B. The position $(x, y, z)_k$ of a moving hole C at time $t = t_k$ is uncertain with respect to a known reference frame $\bar{C}$. The task is to transfer the peg to the target frame T. The hole is moving within the field of view of a camera with frame V. The camera provides detections of the hole center $(p_x, p_y)_k$ and the joint sensors provide joint position $\boldsymbol{q}_k$ and the external torque $\boldsymbol{\tau}_k$ induced by the contact wrench $\boldsymbol{w}_k$

4.2 Virtual Contact Model

A virtual contact model is required for the sample propagation and update in the state estimation. As in our previous works [54, 57], we use a fast and accurate penalty-based collision detection algorithm [58] for the contact force and distance computation. The implementation is based on the voxelmap-pointshell (VPS) algorithm by McNeely et al. [46]. The object geometries are efficiently represented by voxelmaps and pointshells, as depicted in Fig. 5. It can naturally handle complex and non-convex geometries, as in our work on intrinsic tactile sensing with aluminum profiles [54].

Dependent on the relative pose $\boldsymbol{H}_k = \boldsymbol{H}_{CD}(\boldsymbol{q}_k, \boldsymbol{x}_k) \in SE(3)$ of the objects, the contact model computes the virtual contact wrench $\tilde{\boldsymbol{w}}_k = (\tilde{\boldsymbol{F}}_k, \tilde{\boldsymbol{M}}_k) = \tilde{\boldsymbol{w}}(\boldsymbol{H}_k)$ with contact force $\tilde{\boldsymbol{F}}_k \in \mathbb{R}^3$ and torque $\tilde{\boldsymbol{M}}_k \in \mathbb{R}^3$. Furthermore, the contact distance $\tilde{d}_k = \tilde{d}(\boldsymbol{H}_k) \in \mathbb{R}$ is calculated, which is positive for penetrations. The contact distance defines implicitly the relative configuration space $\tilde{\mathcal{C}}$ between the virtual representations of both objects:

$$\begin{cases} no\ contact\ (\boldsymbol{H}_k \in \tilde{\mathcal{C}}): & \tilde{d}_k < 0 \\ contact\ (\boldsymbol{H}_k \in \partial\tilde{\mathcal{C}}): & 0 \leq \tilde{d}_k < d_t \\ invalid\ (\boldsymbol{H}_k \notin \tilde{\mathcal{C}}): & d_t < \tilde{d}_k, \end{cases} \tag{2}$$

where $d_t > 0$ is a threshold on the maximal feasible virtual penetration. In the contact case we allow a small intersection, which is necessary for the penalty-based algorithm. In this work, the joint torque sensors of the manipulator will be used instead of a force/torque sensor at the endeffector. Therefore, $\tilde{\boldsymbol{w}}_k$ is mapped to a virtual contact torque $\tilde{\boldsymbol{\tau}}_k$ in joint space with $\tilde{\boldsymbol{\tau}}_k = \boldsymbol{J}_k^T \tilde{\boldsymbol{w}}_k$, where $\boldsymbol{J}_k := \boldsymbol{J}_{BD}^D(\boldsymbol{q}_k) \in \mathbb{R}^{6 \times n}$ denotes the Jacobian of the robot arm with respect to D. The virtual stiffness of the contact and the threshold d_t are selected such that the real contact wrenches during the insertion are reproducible in magnitude. Furthermore, we assume a frictionless and quasi-static contact. Although the contact model simplifies the physical effects drastically, it provides adequate directional information to distinguish certain contact states and to reduce position uncertainty. Naturally, friction has a crucial effect on jamming in peg-in-hole applications, but as will be seen later, the model provides sufficient information in the considered experiments and jamming can be prevented by an appropriate motion strategy.

4.3 Propagation Model

The real motion of the hole is unknown, therefore we apply a constant velocity (CV) tracking model at first. In a second stage, we combine it with a heuristic to increase the sampling performance for the peg-in-hole use case. The first stage of the propagation is given by a general CV model [13, p. 58]:

$$\boldsymbol{x}_{I,k} = \left(\boldsymbol{I}_3 \otimes \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \right) \boldsymbol{x}_{k-1} + \boldsymbol{v}_k, \tag{3}$$

where $\boldsymbol{I}_3$ is the 3×3 identity matrix, $\otimes$ is the Kronecker product, T is the duration of the time step and $\boldsymbol{v}_k$ is Gaussian noise with covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}}$. $\boldsymbol{x}_{I,k}$ is an intermediate auxiliary state that will be passed to the second stage.

In [54], we investigated various heuristics to improve the propagation model for observing peg-in-hole tasks, which are inspired by probabilistic roadmap planning [36], namely by the Gaussian sampler of Boor et al. [9] and the bridge test by Sun et al. [64]. It was shown that especially the bridge test helped to increase the sample density within the narrow passage of the configuration space. Thus, more efficient sampling is possible with a reduced risk of sample impoverishment, which is an undesired effect of particle filtering approaches. This principle is depicted in Fig. 6 and summarized in Algorithm 1 together with the constant velocity propagation.

The bridge test is an iterative policy that draws an auxiliary sample in each cycle of the loop. This auxiliary sample has a frame II in the neighborhood of the original sample frame I in order to find so-called bridge points in the configuration space, denoted with frame III. The bridge point is then located at the half distance between I and II. The function EVALCONTACT is needed to test if a sample is in the configuration space $\tilde{\mathcal{C}}$ according to Eq. 2, and the first stage propagation (3) is implemented in the function CONSTANTVELOCITY. Note that for better

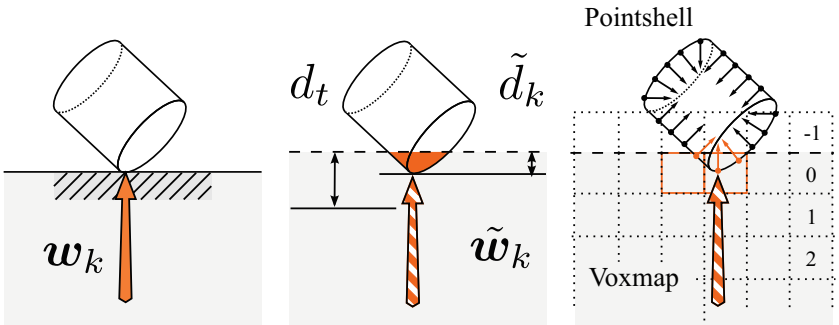


Fig. 5 Left: contact situation with contact wrench $\boldsymbol{w}_k$. Center: penalty-based contact model. Right: implementation of contact model with a voxelmap and pointshell representation of the objects. $\tilde{d}_k$ denotes the contact distance and d_t a threshold on the maximal feasible virtual penetration, $\tilde{\boldsymbol{w}}_k$ the virtual contact wrench. Compare [54]

Algorithm 1 Propagation model.

function PROPAGATESAMPLE($\boldsymbol{x}_{k-1}{}^{(i)}, \boldsymbol{q}_k$)
  $\boldsymbol{x}_I \leftarrow$ CONSTANTVELOCITY($\boldsymbol{x}_{k-1}{}^{(i)}$) ▷ (3)
  for j := 1 to L_{max} **do**
    draw $\boldsymbol{p}_{II} \sim \mathcal{N}(\boldsymbol{p}_I, \boldsymbol{\Sigma}_{\boldsymbol{p},b})$.
    $\mathcal{C}_{II} \leftarrow$ EVALCONTACT($\boldsymbol{p}_{II}, \boldsymbol{q}_k$) ▷ (2)
    if $\mathcal{C}_{II} = invalid$ **then**
      $\boldsymbol{p}_{III} \leftarrow (\boldsymbol{p}_I + \boldsymbol{p}_{II})/2$
      $\mathcal{C}_{III} \leftarrow$ EVALCONTACT($\boldsymbol{p}_{III}, \boldsymbol{q}_k$) ▷ (2)
      if $\mathcal{C}_{III} \neq invalid$ **then**
        return $\boldsymbol{p}_{III}$
  draw $\boldsymbol{p}_{IV} \sim \mathcal{N}(\boldsymbol{p}_I, \boldsymbol{\Sigma}_{\boldsymbol{p},p})$.
  return $\boldsymbol{p}_{IV}$

readability, we denote the position components of $\boldsymbol{x}$ by $\boldsymbol{p} = (x, y, z)$. Furthermore, $\mathcal{N}(\boldsymbol{p}, \boldsymbol{\Sigma})$ denotes a multivariate Gaussian distribution with mean $\boldsymbol{p}$ and covariance matrix $\boldsymbol{\Sigma}$. The operation $s \sim \mathcal{D}$ generates a sample s from a distribution $\mathcal{D}$. The covariance $\boldsymbol{\Sigma}_{\boldsymbol{p},b}$ defines the size of the neighborhood of I and can be chosen according to the gap size of the passage. The number of maximal iterations L_{max} controls the admissible effort in the search for a bridge point, and also the density in the narrow passage. If no bridge point can be found, then the sample I will be returned with small additional Gaussian noise $\boldsymbol{\Sigma}_{\boldsymbol{p},p}$ in order to avoid sample impoverishment.

4.4 Tactile Likelihood

Once a robot has grasped an object and brings it into contact with the environment, *intrinsic* tactile sensing is an important ingredient to distinguish contact states and estimate uncertainties (whereas during grasping extrinsic tactile sensing with sensors directly at the fingertips plays a major role, see [18] for a classification of robot tactile sensing approaches). In this work, the internally measured joint torques are used for intrinsic tactile sensing. The tactile likelihood in the update step of the Bayesian state estimator is computed using a comparison of the current joint position and torque measurements $\boldsymbol{y}_k^{s_t} = (\boldsymbol{q}_k, \boldsymbol{\tau}_k)$ of the robot with the virtual contact model as described in the following.

Firstly, we ensure consistency in the relative configuration space of the peg and hole feature using

$$s_d(\boldsymbol{y}_k^{s_t}|\boldsymbol{x}_k^{(i)}) = \frac{1}{\sigma_d\sqrt{2\pi}} \cdot \begin{cases} 1, & \boldsymbol{H}_k^{(i)} \in \tilde{\mathcal{C}}, \\ \exp(\frac{-(\tilde{d}_k - d_t)^2}{2\sigma_d^2}), & \boldsymbol{H}_k^{(i)} \notin \tilde{\mathcal{C}}. \end{cases} \quad (4)$$

It ensures that the virtual objects stay in the valid configuration space given by the threshold d_t on the virtual contact distance $\tilde{d}_k$ [54]. This means that the objects are not allowed to intersect. Secondly, we incorporate the force information from the contact by comparison of the measured torques $\boldsymbol{\tau}_k$ with the torques computed by the virtual model assuming normal distributed errors with covariance $\boldsymbol{\Sigma}_{\boldsymbol{\tau}}$ in the measurements [54]:

$$s_F(\boldsymbol{y}_k^{s_t}|\boldsymbol{x}_k^{(i)}) = \mathcal{N}(\boldsymbol{\tau}_k|\tilde{\boldsymbol{\tau}}_k^{(i)}, \boldsymbol{\Sigma}_{\boldsymbol{\tau}}). \quad (5)$$

Here, the magnitude and the direction of the contact forces are evaluated in joint space. Contact states can be distinguished by the directional information, which is important for the convergence of the filter in the peg-in-hole task. For instance, lateral forces acting on the peg can imply that it is already partially inserted, whereas vertical forces can mean that the upper rim of the hole is touched. The full tactile likelihood is consequently derived as the product of those two elementary likelihoods:

$$p(\boldsymbol{y}^{s_t}|\boldsymbol{x}_k^{(i)}) = s_d(\boldsymbol{y}^{s_t}|\boldsymbol{x}_k^{(i)}) \cdot s_F(\boldsymbol{y}^{s_t}|\boldsymbol{x}_k^{(i)}). \quad (6)$$

Furthermore, in the case of multiple similar parts or similar local tactile features, the concept of observable regions [66] could be introduced as suggested in our previous work on visual and touch-based sensing [57]. It states that the tactile update shall only be done for reachable samples, i.e., samples that can potentially be touched within a motion step. However, this is not necessarily required here as we are only considering a single tactile feature in the geometrical shape of the hole in its entirety.

4.5 Visual Likelihood

Generally, the proposed method is capable of handling multiple cameras with static and variable poses. However, without loss of generality, we capture images from a single monocular camera at a fixed pose $\boldsymbol{H}_{BV} = \text{const.} \in SE(3)$.

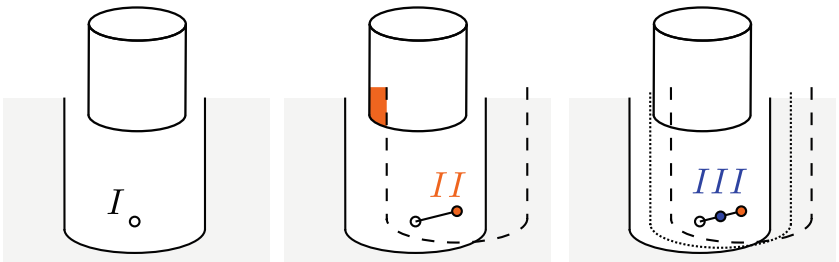


Fig. 6 The bridge test policy in three steps. An auxiliary sample with frame II is drawn in the neighborhood of the original sample frame I in order to find so-called bridge points with frame III in the configuration space, which is located at half-distance between I and II

Certainly, better visual feature detection can be achieved with multiple cameras, mobile cameras and depth image acquisition techniques. Nevertheless, we use the monocular stationary camera in order to show that the missing information can be inferred during assembly execution using tactile sensing.

We use a simple blob detection algorithm in order to extract hole features from the image. In this work, we will assume that only a single feature is present in the image, but the method is in general also applicable for multiple detections [57]. The center of the area is computed in pixel values and forms the visual measurement vector

$$\boldsymbol{y}_k^{s_v} = \begin{pmatrix} p_x & p_y \end{pmatrix}^\top , \tag{7}$$

where p_x, p_y denote the center coordinates of the detection in pixels. We assume a pin-hole camera model [26, pp. 153f] for the visual sensor model. The function project : $\mathbb{R}^6 \rightarrow \mathbb{R}^2$ implements the pin-hole model by taking the position components of the state vector and projecting them onto the image plane. Given the intrinsic parameters of the camera, this function can be straightforwardly derived.

We then use a multivariate Gaussian for the likelihood model with the mean being the projected version of the state vector

$$p(\boldsymbol{y}_k^{s_v} | \boldsymbol{x}_k^{(i)}) = \mathcal{N}(\boldsymbol{y}_k^{s_v} | \,\text{project}(\boldsymbol{x}_k^{(i)}), \boldsymbol{\Sigma}_v), \tag{8}$$

where $\boldsymbol{\Sigma}_v$ denotes the expected covariance of $\boldsymbol{y}_k^{s_v}$. We use a diagonal covariance matrix here, i.e., we assume the components of the measurement vector to be uncorrelated.

Similar to the tactile case, the concept of observable regions can be introduced for the visual domain. Visual observable regions are commonly known as *fields of view*. Detectable regions are subsets of the latter in which the features are detected with a high confidence. Occlusions, e.g., from the robot, further shrink the detectable region and we need to incorporate that particular case in our approach. Therefore, as suggested in [57], we set the likelihood $p(\boldsymbol{y}_k^{s_v} | \boldsymbol{x}_k^{(i)}) = 1$ if the robot occludes the view on a particular sample, which can be computed from the sample and the robot pose. Thus, the vision cannot decrease the likelihood of a sample in that case.

4.6 Visual Tactile Update Model

In the update step of the recursive filter, the samples are weighted using the likelihood of the measurements. In this work, the weights are computed according to the bootstrap filtering approach by Gordon et al. [23], compare [12]: $W_k^{(i)} \propto p(\boldsymbol{y}_k | \boldsymbol{x}_k^{(i)})$. We multiply the likelihoods from both tactile and visual sensors, Eqs. 6 and 8, and obtain the joint likelihood

$$p(\boldsymbol{y}_k | \boldsymbol{x}_k^{(i)}) = p(\boldsymbol{y}_k^{s_v} | \boldsymbol{x}_k^{(i)}) \cdot p(\boldsymbol{y}_k^{s_t} | \boldsymbol{x}_k^{(i)}) . \tag{9}$$

The implementation of the update model is summarized in Algorithm 2. Note that logarithmic weights are used in the implementation. Resampling is performed afterwards using systematic resampling [28].

Algorithm 2 Update model.

```
function WeightSample(y_k^{s_v}, y_k^{s_t}, x_k^{(i)})
    a ← TactileLikelihood(y_k^{s_t}, x_k^{(i)})
    b ← VisualLikelihood(y_k^{s_v}, x_k^{(i)})
    weight ← ln a + ln b        ▷ update particle weight
    return weight
```

5 Motion Generation

Assembly tasks are typically implemented in static settings where parts are kept at a constant and stable location using specialized part holders. In the previous section, we presented a general approach that combines visual and tactile sensing to continuously track the parts in dynamic environments within a single Bayesian framework. Based on this, it is now possible to implement an object-centric motion generation algorithm that is guided by the estimated poses of the parts. A tilt-and-align strategy is implemented and combined with an adaptive path executor as described in the following.

5.1 Tilt-and-Align Strategy

The investigation of peg-in-hole assembly traces back to the early history of robotics research. Inoue [29] presented strategies for loose- and close-fit cases in the example of shaft-bearing assembly. A crucial component is the tilt of the peg to increase the robustness against pose uncertainties. Multiple works use this principle in various approaches for peg-in-hole, e.g., [11, 16, 32, 63]. We will also employ a tilt-and-align strategy and follow the planning method of Stemmer et al., which was demonstrated for complex shaped planar parts [63]. The basic idea is to align the peg with the contour of the hole by pressing in the lateral direction of corner features. A pushing motion is commanded into this direction using a Cartesian impedance controller [2] in order to achieve robustness against pose uncertainties. Based on a prior analysis of the geometric shape of the contours, regions of attractions (ROA) can be identified in which the starting point of the pushing motion, i.e., the lowest point of the tilted peg, must lie in it in order to guarantee a successful and robust alignment with respect to small rotational and lateral offsets. Although the method was proven to be fast and robust against uncertainty, it did not directly incorporate the feedback of the hole pose, and

thus, is by itself insufficient for assembly with parts moving on a larger scale. However, because of its robustness, we define a nominal strategy according to [63] and will show how to combine it with an adaptive motion generation step in the next section.

5.2 Adaptive Task Execution

Following the skill-based programming approach in our system, we define an object-centric tilt-and-align strategy and use the state estimation to adapt the execution online. The object-centric formulation is suitable for many manipulation tasks and was applied in various domains, e.g., robotic assembly [70] or assistive robotics [55]. Recently, Migimatsu and Bohg [49] describe an object-centric task and motion planning approach (TAMP) and show how it can be combined with a reactive controller that allows the plans to adapt to the online measured poses of objects. However, they use visual perception only, and additional fiducial markers increase the tracking performance. In our case, we assume that the objects are only visible in the first phase and are then occluded such that tactile sensing becomes necessary.

First of all, we specify a nominal geometric path of the peg frame D with respect to the hole frame C according to the tilt-and-align strategy. It connects a start frame with the target frame T at the bottom of the hole and is given as a sequence $\mathcal{T} = (T_1, T_2, \ldots, T_L)$ of interpolated path frames T_l with $l = 1, ..., L$; $\boldsymbol{H}_{CT,l} = \text{const.} \in SE(3)$ denotes the homogeneous transformation from C to T_l. Note that the path frames do not need to be consistent with the real configuration space between both parts, but can include offsets to support the passive alignment of the geometries with the help of the Cartesian impedance controller. For example, we will introduce an offset for the push motion against the hole contour, and an offset in the final frame T_L to align the peg stably with the bottom of the hole, respectively. An example path is visualized as orange line in Fig. 7.

The path is then executed in a conditional loop that evaluates the distance to the next path frame as listed in Algorithm 3. The internal while-loop includes the functions for the state estimation and analyzes the sample distribution for the generation of the next peg pose. For this purpose, an estimate of the hole pose $\hat{\boldsymbol{H}}_{BC}$ is computed using (1) with $V : \boldsymbol{x} \mapsto (x, y, z)$ for the computation of the expected value. The estimated relative pose between both parts $\hat{\boldsymbol{H}}_{CD}$ can THEN be obtained by the forward kinematics. The function GETDISTANCES calculates the Euclidean distance $d_T \in \mathbb{R}$ of the position and the geodetic distance $d_R \in \mathbb{R}$ on $SO(3)$ between $\hat{\boldsymbol{H}}_{CD}$ and the current path point l with transformation $\boldsymbol{H}_{CT,l}$. The parameters $d_{T,max} \in \mathbb{R}$ and $d_{R,max} \in \mathbb{R}$ control the permissible path deviations. As long as it is not reached, a motion to T_l will be generated with the desired transformation $\boldsymbol{H}_{BD,k,d} = \hat{\boldsymbol{H}}_{BC}\boldsymbol{H}_{CT,l}$ which is send as reference to the underlying Cartesian impedance controller. We assume that the generated motions are reachable in joint space and that the robot is not in a singular configuration, which can be evaluated and guaranteed using task-specific workspace maps [6]. The underlying impedance controller ensures that the contact is stable, and passively compensates small pose errors that occur when the estimate is not yet accurate.

Algorithm 3 Adaptive motion generation.

```
function GENERATEMOTION(𝒯)
    for l := 1 to L do
        reached ← false
        while not reached do
            y_k ← GETMEASUREMENTS( )
            for all x_k^(i) ∈ 𝒳_k do
                x_k^(i) ← PROPAGATESAMPLE(x_k^(i), q_k)
                W_k^(i) ← WEIGHTSAMPLE(y_k, x_k^(i))
            𝒳_{k+1} ← RESAMPLE(𝒳_k)
            k ← k + 1
            Ĥ_BC ← ESTIMATEHOLEPOSE(𝒳_k)
            Ĥ_CD ← GETRELATIVEPOSE(Ĥ_BC, q_k)
            d_T, d_R ← GETDISTANCES(Ĥ_CD, H_CT,l)
            if d_T ≥ d_T,max ∨ d_R ≥ d_R,max then
                H_BD,k,d ← NEXT(Ĥ_BC, H_CT,l)
            else
                reached ← true
                H_BD,k,d ← NEXT(Ĥ_BC, H_CT,l+1)
            EXECUTEMOTION(H_BD,k,d)
```

6 Evaluation

We systematically evaluate the approach with a dual-arm robotic setup. In particular, the assembly skill is executed under varying conditions and with various part geometries. Furthermore, the effects of the modalities in the likelihood function are investigated.

6.1 Experimental Setup

Figure 8 shows our setup for the peg-in-hole experiments. It consists of two 7-dof KUKA LBR iiwa robots with joint torque sensors. The left robotic arm executes the assembly skill, whereas the right robotic arm simulates the unknown hole motions. The right arm is only used to measure the ground truth pose of the hole and does not share this information with the active robot executing the skill. Furthermore, a monocular camera is mounted rigidly above the table at a

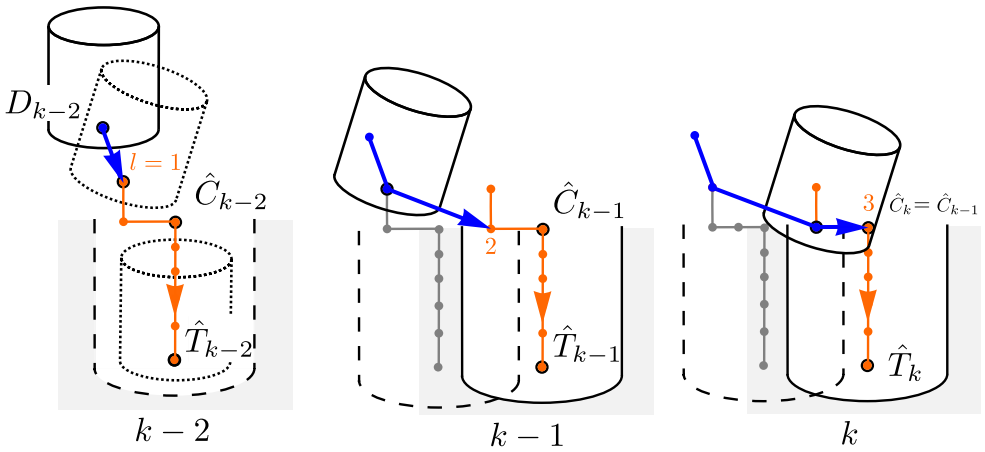


Fig. 7 Adaptive execution of an object-centric path (orange line) considering the currently estimated frame of the hole $\hat{C}_k$. The hole moves to the right between time step $k-2$ (left) and $k-1$ (center). The motion commands (blue lines) follow the estimated poses

distance of $\approx$ 1.5 m. It provides images with a resolution of 1620 $\times$ 1220 pixels. The hole feature detector provides observations at a rate of 18 Hz. In this setup, three part combinations are investigated: a configuration with square peg and hole $\mathcal{P}_\square$, one with a round peg in a square hole $\mathcal{P}_\times$ and a cylindrical peg-in-hole with round peg and round hole $\mathcal{P}_\circ$ (see Fig. 9). The parts are made of aluminum. The pegs have a chamfered edge of 2 mm, the holes are chamferless and have a depth of 60 mm; the round peg has a diameter of 78.9 mm, the round hole 79.1 mm, the side length of the square peg is 79.8 mm and of the square hole 80 mm.

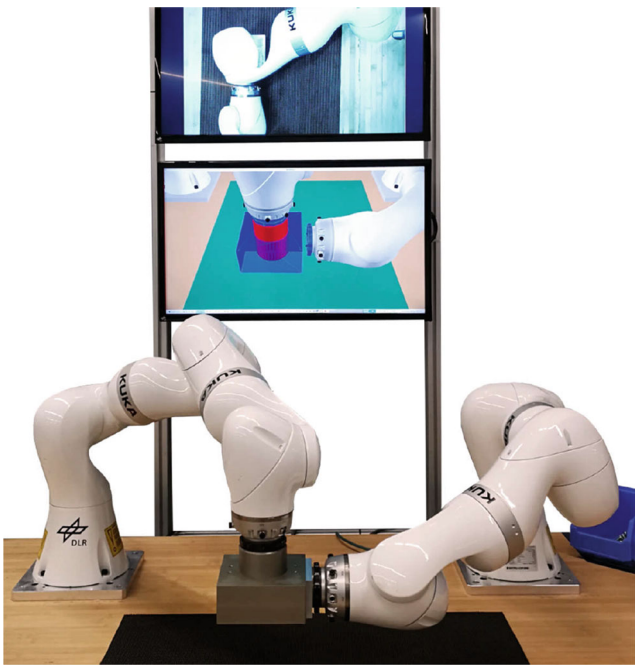


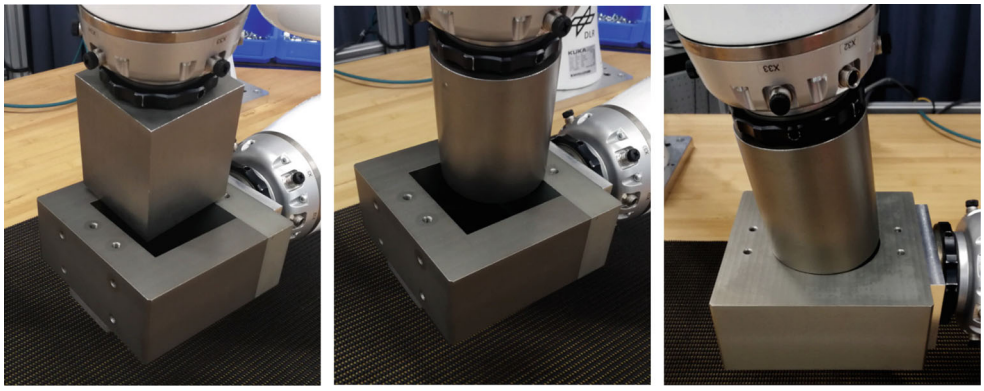
Fig. 8 Setup for the peg-in-hole experiments. The left arm executes the assembly skill. The right arm is used as a ground truth measurement device and simulates the hole motions. The camera image is visible in the upper screen, the lower screen shows the live view of the world model

The particle filter implementation features a parallel propagation and update of the samples in up to 16 threads, which is important for the collision checks in the virtual contact model, which requires $\approx$ 1 ms per call. The other functions in Algorithm 3 are executed sequentially. In the online application of the framework, we use a set of $N = 320$ samples, which is a sufficient number to provide a reliable estimate in this scenario, compare [54] for an analysis of required sample numbers. The parameters are summarized in Table 1. Given those parameters, a command rate of $\approx$ 5 Hz can be realized by the motion generator. We define a path $\bar{\mathcal{T}}$ which is applicable for all three cases; the rotational parts of the path points in Table 1 are listed with parameters α, β, γ, which are Z-Y-X Euler angles [17, p. 43]. Note that we additonally refine the path by carrying out an interpolation in the translation of 0.5 points/mm and 1 points/deg in rotation in order to obtain $\mathcal{T}$. Figure 10 visualizes the nominal peg motion (left) defined for the object-centric skill and the executed motion (right) for one of the experiments carried out.

On side of the robot, a Cartesian impedance controller is used with an additional small oscillating motion overlay for the task frame motion according to a given force amplitude and frequency. This is a common strategy for peg-in-hole tasks employed to improve robustness of the insertion against pose uncertainties. Note that the internal controller of the robot runs at a controller rate $>$ 1 kHz and generates trajectories in finer granularity and guarantees a stable execution.

6.2 Variation of the Execution Conditions

The following experimental procedure is carried out for multiple runs. First, the hole is randomly positioned in a region below the camera mounted above the table. The state estimator is then initialized with the first visual detection of the hole. Due to the projective nature of cameras, it is not possible to reconstruct a full state vector from a



(a) $\mathcal{P}_{\square}$ (b) $\mathcal{P}_{\times}$ (c) $\mathcal{P}_{\circ}$

Fig. 9 Snapshots of the assembly experiments: square peg-in-hole $\mathcal{P}_{\square}$, round peg into square hole $\mathcal{P}_{\times}$, and cylindrical peg-in-hole $\mathcal{P}_{\circ}$

single visual detection $\boldsymbol{y}^{s_v}$ without additional constraints. Therefore, we randomly sample a vertical coordinate $z_0^{(i)}$ from a uniform distribution of 10 mm width and use this value as a constraint for the reconstruction (compare [57] for a detailed algorithm) and obtain the initial set of $\mathcal{X}_0$ with the additional assumption that the feature is not moving at start time. The samples are then aligned along the ray direction of the camera for the visible hole in the image plane (Fig. 11a) and because of the constant velocity model, they start spreading in all directions of the xy-plane immediately. However, they stay in a bounded region due to the update with the visual sensor (see Fig. 11b).

At first, the hole is at a static pose and after 10 steps the hole motion is triggered. The passive robot moves the hole along a line 100 mm long with a Cartesian velocity of 2 mms^{-1}. The hole is slowly drifting away, and at this point, the motion is tracked by visual sensing only. We have designed the procedure such that the tactile sensing and robot motion start at $k = 25$. Once the robot moves the peg to the first path point relative to the estimated hole pose, it occludes the camera's field of view. By comparing the peg frame D and the current pose of a sample, the implemented algorithm recognizes if a sample is within the detectable region of the vision system or whether the robotic arm occludes it. If the distance between the projected frames of peg and sample in the image plane is below a threshold of 100 pixels, we assume that the sample is occluded. Doing this, we can ensure that features are always visible completely and no offset occurs in the estimate due to a shifted blob center of a partially occluded hole. The samples outside of the detectable region are then only updated using the tactile likelihood (compare Section 4.5). The transition from Fig. 11c to d shows how the sample distribution reshapes according to the influence of the geometry of the parts when the peg comes closer. The spread of the sample distribution is then limited by the borders of the relative configuration space between both parts.

In the following phase, the bridge test policy helps to pull samples into the narrow passage in the relative configuration space and the distribution appears funnel-shaped. During the insertion, the samples then align along the hole axis (Fig. 11f) and condense in a small region (Fig. 11h). Note that in Fig. 11g) the peg has already reached the physical bottom of the hole, but that there is still a significant spread in the z-direction. This is due to the fact that the controller has not yet generated enough force through the contact. Nevertheless, an accurate estimate of the hole pose can be obtained at the end with the help of the force feedback.

This experiment was repeated 10 times for each of the three investigated cases. In all runs the peg was successfully inserted. The state evolution for one example[2] of each series is plotted together with the ground truth measurement in Fig. 12. The plot for $\mathcal{P}_{\circ}$ in particular shows a characteristic evolution of the above-described process. The distribution in the z-direction stays constant before the peg motion starts at $k = 25$, where it shrinks the first time according to the configuration space constraints. The spread in the x- and y-direction narrows at $k \approx 45$ when the parts are aligned and the insertion starts. From this point onward, the hole motion in the plane is accurately tracked. At $k \approx 73$ the hole motion stops, and soon after the peg reaches the bottom the distribution in the z-direction shrinks for the second time.

In the x- and y-direction, the final estimate is very close to the ground truth value. Yet in the z-direction, a remaining offset is observable in all three experiments. One factor for the remaining deviation to the ground truth value is the force which is still applied in the z-direction by the impedance controller due to the offset in the final path point. The virtual contact model needs a little penetration of the geometries in order to counterbalance the external force.

[2]Videos and further visualizations are provided in the supplemental material

Table 1 Parameters of experiments

Contact Model		
voxelmap resolution	1.0	mm
pointshell resolution	3.0	mm
stiffness	25000	Nm^{-1}
d_t	2	mm
Propagation Model		
# samples	320	
$\boldsymbol{\Sigma_x}$	diag [0, 0.1, 0, 0.1, 0, 0]	mm, mms^{-1}
$\boldsymbol{\Sigma_{p,b}}$	diag [6, 6, 3]	mm
$\boldsymbol{\Sigma_{p,p}}$	diag [0.1, 0.1, 0.1]	mm
L_{max}	5	
Update Model		
$\boldsymbol{\Sigma_v}$	diag [10, 10]	pixel
σ_d	0.5	mm
$\boldsymbol{\Sigma_\tau}$	diag [5, 5, 5, 5, 5, 5]	Nm
Motion Generation		
$d_{T,max}$	5	mm
$d_{R,max}$	5	deg
$\bar{\mathcal{T}} =$	$x,\ y,\ z,\ \alpha,\ \beta,\ \gamma$ $([-10,\ 0,\ 10,\ 0,\ 10,\ 0],$ $[\ 10,\ 0,\ -2,\ 0,\ 10,\ 0],$ $[\ 0,\ 0,\ -2,\ -3,\ 10,\ 0],$ $[\ 0,\ 0,\ -10,\ 3,\ 0,\ 0],$ $[\ 0,\ 0,\ -70,\ 0,\ 0,\ 0])$	mm, deg
transl. interpol.	0.5	points/mm
rot. interpol.	1.0	points/deg
command rate	5	Hz
Impedance Controller		
task frame	*D*	
transl. stiff. (x/y/z)	5000/5000/3000	Nm^{-1}
amplitude	3/3/0	N
frequency	1.5/2/-	Hz
rot. Stiff. (x/y/z)	300/300/50	Nmrad^{-1}
amplitude	0.5/0.5/0	Nm
frequency	1.5/1.4/-	Hz
cartesian velocity	20	mms^{-1}
controller rate	> 1000	Hz

Figure 13 shows the Cartesian force at frame *D*. The virtual model is capable to represent and estimate the acting external forces which is visible in the small deviation between ground truth and expected value of the force components. Between $k = 25$ and $k = 45$, the touches the upper rim of the hole; during insertion, only minimal forces act in the z-direction, and a clear step is visible at the end. Note that although friction effects are not explicitly modeled, the virtual model is able to provide sufficient directional information to support the convergence of the pose estimation, which is especially visible in the condensation of the z-position distributions between $k = 80$ and $k = 120$.

The evolution of the pose estimation error is plotted in Fig. 14 for all runs and shows the Euclidean distance between the ground truth position of the hole and the expected value computed from the samples. Due to the unobservability of the hole feature in direction of the projection line of the camera, the error stays nearly constant until $k = 25$. The robotic arm THEN occludes the field of view and the error arises because there is no feedback from the contact yet and the hole could potentially change its speed or direction. During insertion, the error gradually reduces and is in most cases at terminal time below of the initial error, see Table 2.

6.3 Comparison of Modalities

In order to compare the effects of tactile and visual modalities on the state estimation and skill execution, we carry out a series of experiments using either only the tactile likelihood (6) or only the visual likelihood (8) and compare it with the combined visual-tactile likelihood (9). All parameters are set according to Table 1. Furthermore, we assume that in all cases the visual modality is available at least at the start for a one-shot initialization of the state estimator. In all runs, the hole is positioned at the same initial pose. In particular, we evaluate two cases: at first, a baseline experiment in which the hole is kept at the inital

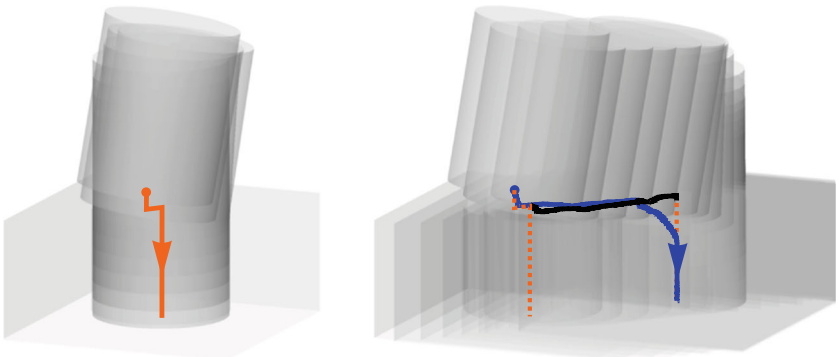


Fig. 10 Nominal object-centric peg motion following a tilt-and-align strategy (left) and finally executed peg motion (right). The nominal path is drawn in orange, the blue line represents the executed path of the peg reference frame, the black line the online estimated pose of the hole to which the motion adapts

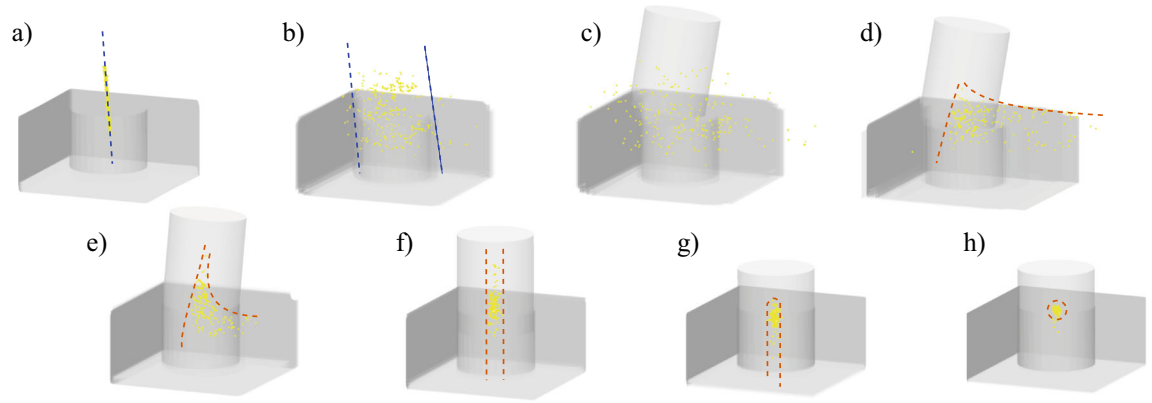
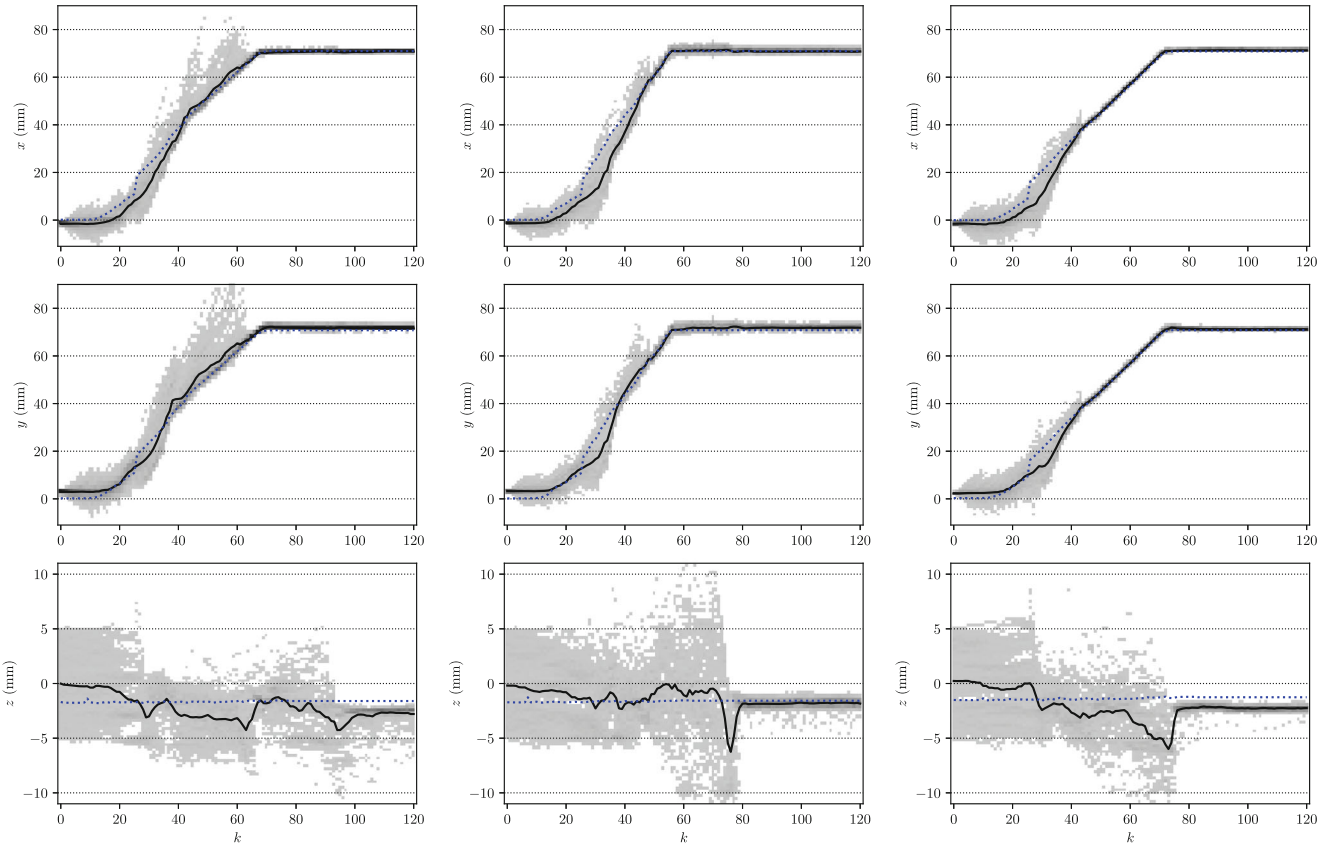


Fig. 11 Evolution of the sample distribution for the cylindrical peg-in-hole ($\mathcal{P}_\circ$). Yellow dots represent the origins of possible hole frames. For each dot the hole geometry is additionally rendered. The peg is displayed with its measured pose. **a** At first, the samples are initialized using the visual detection of the circular feature and the samples align along the projection line (blue dashed line). **b** The constant velocity model of the estimator spreads the samples in planar direction constrained by the visual likelihood. **c**–**g** The field of view is occluded by the robotic arm and the sample update can only be done with tactile measurements. Consequently, samples align according to the local configuration space between both parts (schematically drawn with dashed orange lines). **h** The samples condense at the real pose of the hole. Note that the visualization of the sample dots is scaled up in order to be better visible, whereas the offsets in the hole geometry are at actual scale



(a) $\mathcal{P}_\square$ (square peg - square hole) (b) $\mathcal{P}_\times$ (round peg - square hole) (c) $\mathcal{P}_\circ$ (round peg - round hole)

Fig. 12 Examples of the sample evolution for the x, y and z component of the state for the three investigated scenarios. The gray value indicates the sample density, the black line corresponds to the expected value and the blue dotted line represents the ground truth value from the second robot, i.e., the directly measured hole pose

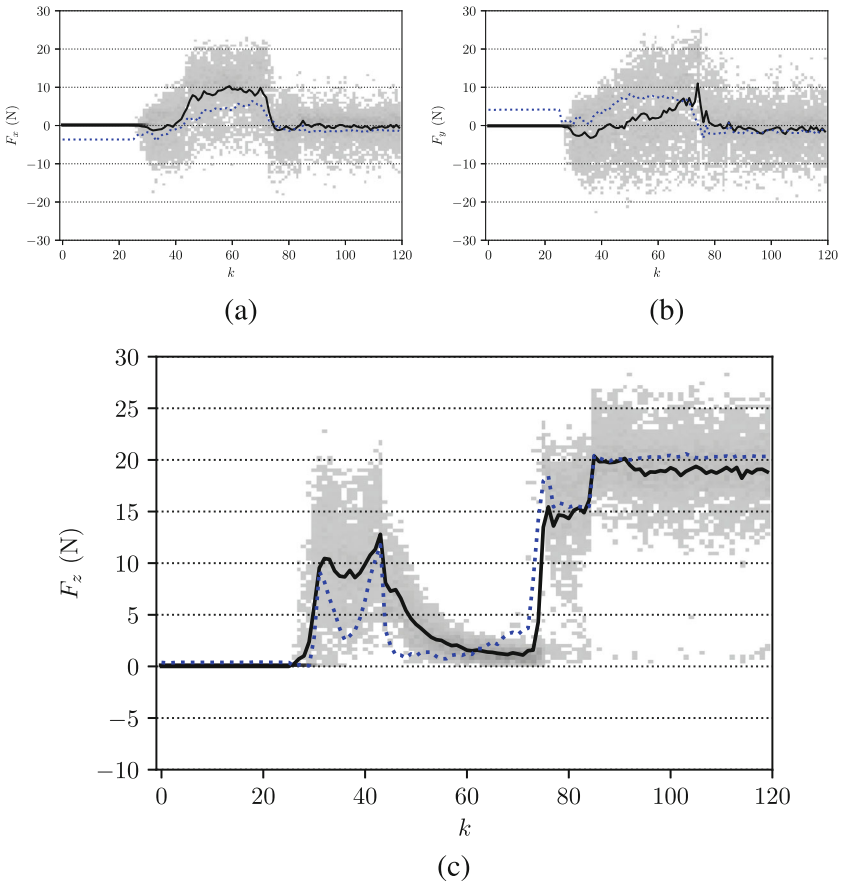


Fig. 13 Measured force (dashed blue line approximated from the joint torque measures using a pseudo inverse of the Jacobian) and force distribution represented by the samples for a run of $\mathcal{P}_\circ$. For each sample the virtual contact force is computed with respect to the peg frame D. The density is given in gray values, the black line corresponds to the expected value of the force distribution

pose, and then the case of a moving hole similar to the one in the previous section.

In all cases tested with a static hole, the insertion was successful due to the robust mating strategy, but there are differences in the state estimates. Figure 15 shows the sample evolution of the x-component of the state in the case of a cylindrical peg-in-hole.[3] Furthermore, Fig. 16a provides the error of the position estimate and the spread of the samples over time (standard deviation of the distance of a sample to the expected value of the position). For the case of tactile modality alone, we can see a growing spread of the samples, i.e., an increasing uncertainty in the estimate, as long as there is no contact between peg and hole. This is due to the modeled assumption that the hole is moving (3), and as long as there is no tactile observation available, this assumption cannot be corrected and the sample evolution is completely governed by the propagation model. Only from $k = 25$ on it can be seen that the spread shrinks due to the tactile likelihood. At the end, an accurate estimate of the hole position with only a small variance can be obtained. This is different in the case of using the visual likelihood alone. Here, the uncertainty at the start is limited, but then increases as soon as the robotic arm blocks the field of view (from $k = 25$ on). Notably, the insertion is still successful. Consequently for a static environment, visual sensing and using a robust strategy is enough for a successful insertion. But since the final phase is not observable, it is not possible to infer solely from the vision data if the peg really reached the desired pose. The visual-tactile sensing is the combination of the best of both worlds. The uncertainty is limited during nearly all all the phases of the process, and the position of the hole can be tracked during insertion.

The same comparison is carried out for the moving hole in a dynamic environment. In this case, only the visual-tactile likelihood enables a successful insertion. By using only the tactile or only the visual likelihood it, is not possible to track the part with sufficient accuracy throughout all phases. Similar to the static case, it is visible in Fig. 16b that in both cases the spread increases as soon as features are not detectable in the modality anymore. At $k \approx 30$, the spread for the tactile likelihood shrinks for a short period due to the sensed contacts. Nevertheless, too many hypotheses of potential hole poses are not longer distinguishable through the tactile feedback and the motion of the hole prevents the convergence of the estimate. In the presented approach, we have no *active* uncertainty reduction

[3] Figures showing the sample evolution for all cases and all state components are provided in the Appendix.

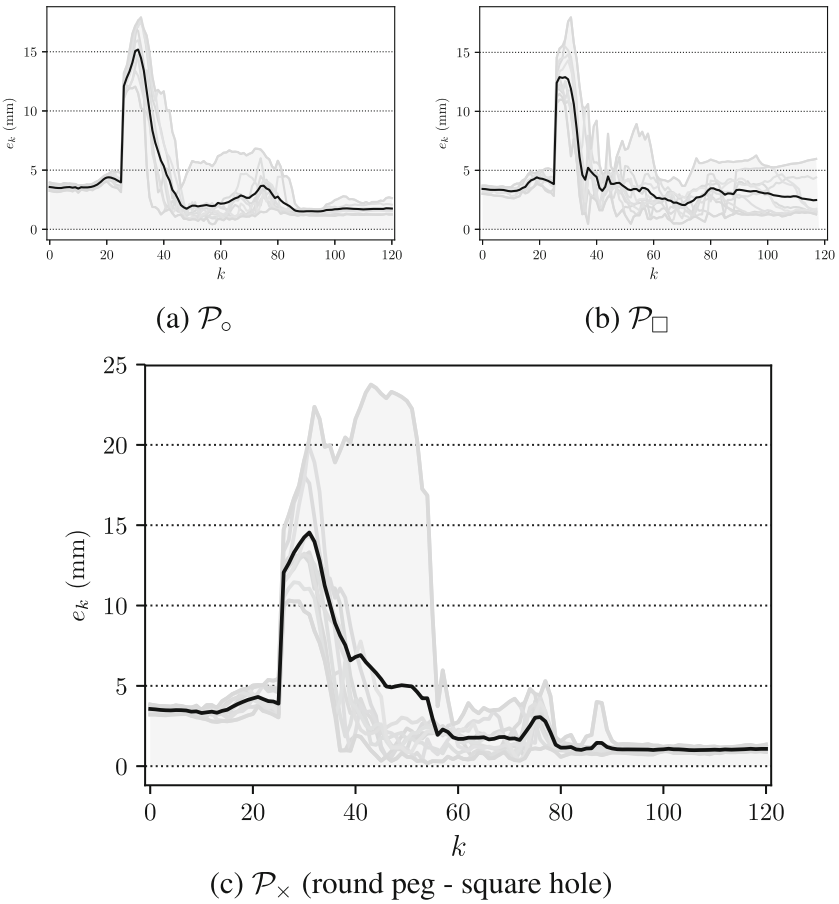


Fig. 14 Pose estimation error over time, computed from the ground truth measurement and the expected value of the sample distribution. Each light gray line represents a single run. The black line is the average of the error over all 10 runs for each case

included in the motion generation step. In future work it might be possible to overcome that issue by triggering dedicated exploration motions as soon as a certain threshold on the spread is reached.

In our experiment, we move the hole with a constant velocity. The visual tracking and identification of the velocity until $k = 25$ could theoretically be sufficient for completing the insertion task. However, offsets in the position typically occur during establishment of contacts (due to compliance, motion changes) which are not visible for the state estimator due to the occlusion. This prevents the successful insertion as the offset can no longer be corrected without feedback. In practice, this could be handled by tuning the insertion motion so that it is faster or more robust against this transition from visual feedback to *blindness*. Nevertheless, additional assumptions regarding the motion direction and speed of the hole would be potentially necessary and the implementation would loose some generality. By using a combined approach, the spread of the possible hole positions is limited through the tactile feedback once the visual features are no longer detectable. The clear advantage here is that fewer assumptions on the motion of the hole are needed and that the *reusability* of the assembly skill is therefore higher. Furthermore, the pose of the hole can accurately and explicitly be estimated during execution of the insertion process.

7 Discussion

The results clearly show that the implemented framework is able to perform peg-in-hole tasks in a dynamic environment with moving parts, but requires visual and intrinsic tactile sensing. An internal probabilistic state representation makes the robotic assembly system aware of the current situation and present uncertainties, and makes it possible to continue the execution although sensors might be occluded or might not yet provide enough information, e.g., in the absence of

Table 2 Final position estimation error

	$\mathcal{P}_\circ$	$\mathcal{P}_\square$	$\mathcal{P}_\times$	
# Runs	10	10	10	
Position error				
min.	1.245	1.261	0.887	mm
max.	2.638	5.962	1.348	mm
average	1.754	2.483	1.076	mm

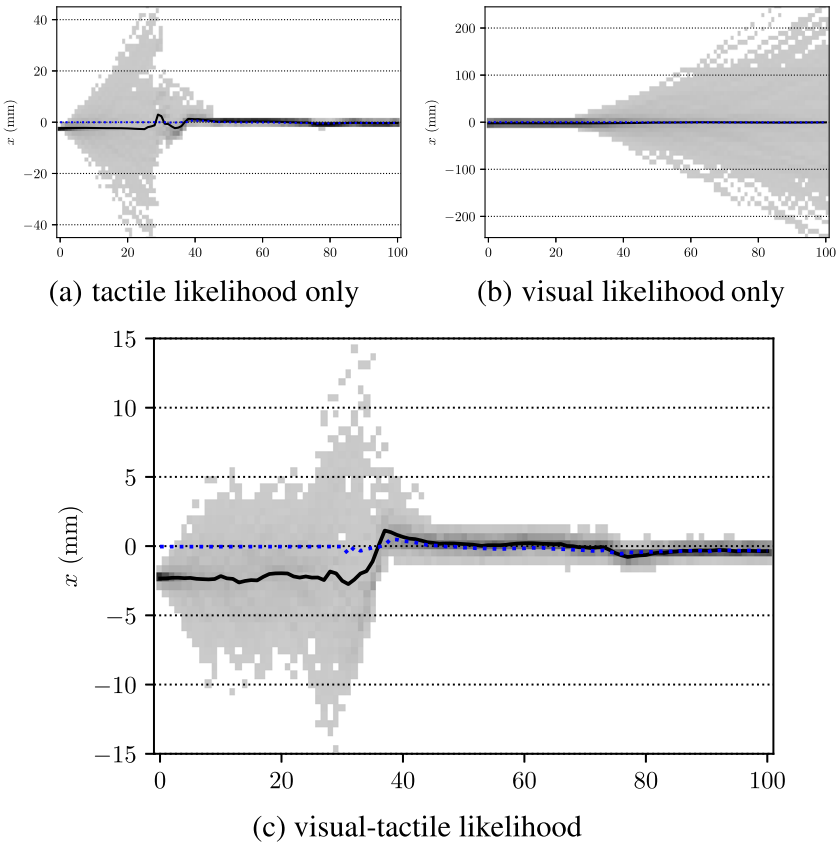


Fig. 15 Sample evolution of the x-component of the state in the case of a cylindrical peg-in-hole ($\mathcal{P}_\circ$) with static hole using three variants of likelihood functions. The gray-value indicates the sample density, the black line corresponds to the expected value and the blue dotted line represents the ground truth value

contact force. Initial uncertainties are reduced and the part position can be tracked during execution.

Theoretically, the state estimation works independently of the presence of sensor modality and the order in which modalities become available. Nevertheless, we are assuming that the vision modality is available at first so that the uncertainty can be efficiently narrowed down at the start. In general, the vision modality makes it possible to detect features globally, whereas tactile sensing typically has only a local scope (see [10] for a comparison of visual and tactile data). Therefore, it is usually better to use the vision modality at first (if available), because a wider field can be observed. The tactile data then helps to refine the estimate and determine state components which are unobservable in the other modality, e.g., a 2D coordinate in the image plane does not provide enough information to retrieve the position of a point in 3D space. This complementary advantage of both modalities were investigated in multiple works, e.g., compare the pioneering work of Allen [3].

In our particular implementation of an assembly skill, we make use of a motion strategy which requires that the lowest point of the tilted peg lie within a region of attraction of the hole (as described in Section 5.1). Accordingly for a successful execution, the uncertainty of the hole center position is not allowed to be larger than the (inner) diameter of the hole. If this is given, then the strategy can be executed successfully. The visual tracking at the beginning ensures that the uncertainty stays within these limits. If the uncertainty were larger, then a tactile exploration phase in the motion strategy would be necessary (compare the search strategies referenced in Section 2.1). Nevertheless, it is an open question as to how such an exploration phase can be implemented efficiently for moving parts in dynamic environments. Therefore, we believe that an initial phase of visual tracking is currently mandatory, and could only be omitted if there were another data source which provides sufficiently accurate position data of the moving part.

In general, the implemented peg-in-hole strategy is robust against small rotation errors up to $\pm$ 5 deg as shown experimentally by Stemmer [63]. Therefore, estimating the orientation of the parts might not always be necessary in many industrial settings. However, for an enlarged field of applications, it is possible to augment the hidden state with another part for orientation, which on the downside

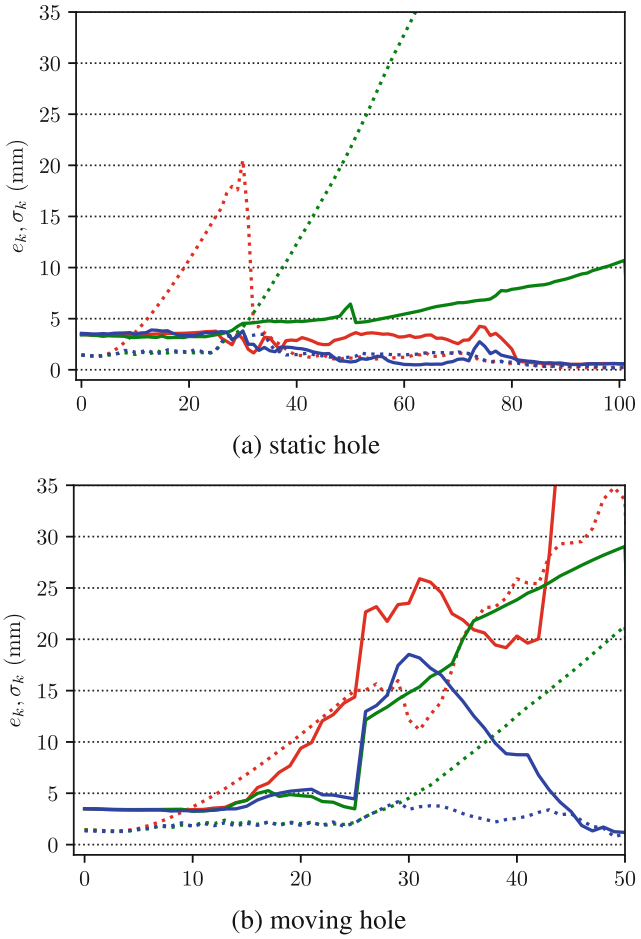


Fig. 16 Pose estimation error over time (solid) and spread of the current sample distribution (dotted line) given as standard deviation for the case of using a tactile (red), visual (green) or visual-tactile likelihood (blue). The values are plotted for $\mathcal{P}_\circ$ and for the case of a static (**a**) and moving hole (**b**)

increases the number of required samples due to the higher dimensionality of the state space. The work of Taguchi et al. [65] shows one possible solution with a Rao-Blackwellized particle filter to obtain an efficient implementation for this problem in a probing-based localization of a static part. Also in another work [53], we started to investigate constraint-based approaches in the propagation model to estimate large rotation motions, but still need to improve the implementation of the contact model to apply it in all phases of the peg-in-hole task. Nevertheless, it is clear that the suggested framework supports these future developments.

In the experiments, we tested three combinations of part shapes. Real parts in industrial use cases typically have more complex shapes. In our previous work [54], we have already demonstrated that the contact model can deal with complex and non-convex geometries in peg-in-hole, but have shown only observation results without motion generation. The implementation of the VPS algorithm is in general suitable for large scenes such as in car manufacturing [58, Sec. 5.2.3]. In future work, alternative and learned contact models could also be applied for the likelihood computation in order to support flexible materials and high friction contacts. Furthermore, for the application in an industrial setting, a speed-up of about one order of magnitude would be necessary. We are very confident that this can be reached by implementing the framework more efficiently. Furthermore, experience-based optimization of the path points and controller parameters could significantly improve execution times for repeated tasks.

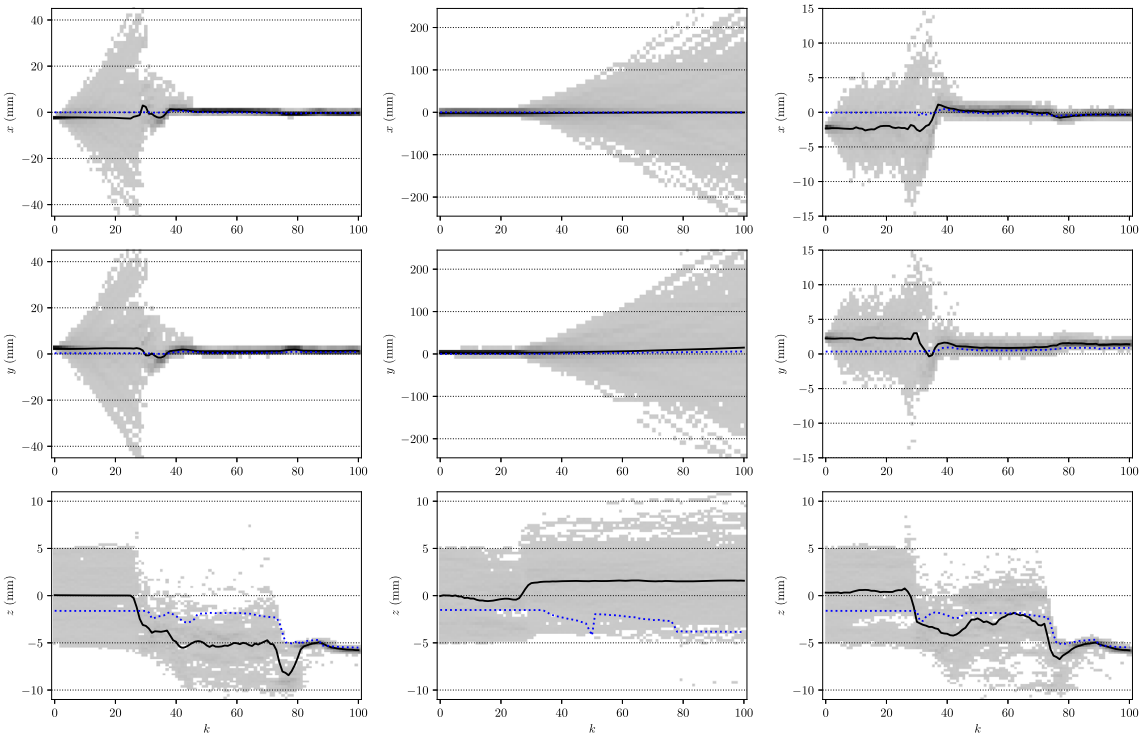
Although the filter step is computationally more expensive than in alternative approaches, an advantage is that the image of the local configuration space can be approximated by the sample distribution, and it is geometrically interpretable. A possible future extension of the presented work is to adapt the controller parameters automatically according to the current shape of the configuration space. Learning approaches could be used on top of the sample distribution to optimize the performance of the insertion strategy.

8 Conclusion

In this work, we presented an approach towards autonomous robotic assembly, which could be used in future manufacturing scenarios in order to increase the flexibility of production facilities. We showed how robotic skills can adapt to moving parts according to the currently observed contact situation by using visual and intrinsic tactile sensing. The general framework is composed of a recursive Bayesian state estimator and an adaptive robot motion generator. The state estimation makes the system aware of the present uncertainties that are affected by occlusions and unknown part motions. The motion generator provides a reactive behavior based on a probabilistic representation that selects the motion according to the currently estimated part poses. In particular, we showcase an object-centric peg-in-hole skill, which is reusable for different part combinations, different initial positions and with moving parts. This skill entails using a robust tilt-and-align assembly strategy implemented with a Cartesian impedance controller and was demonstrated successfully for three different part combinations. In future work, we plan to improve the performance of the framework with respect to execution time and orientation uncertainties. Furthermore, we want to investigate the possibility to include iterative and experience-based learning approaches to map the knowledge of the current contact configuration to controller parameters.

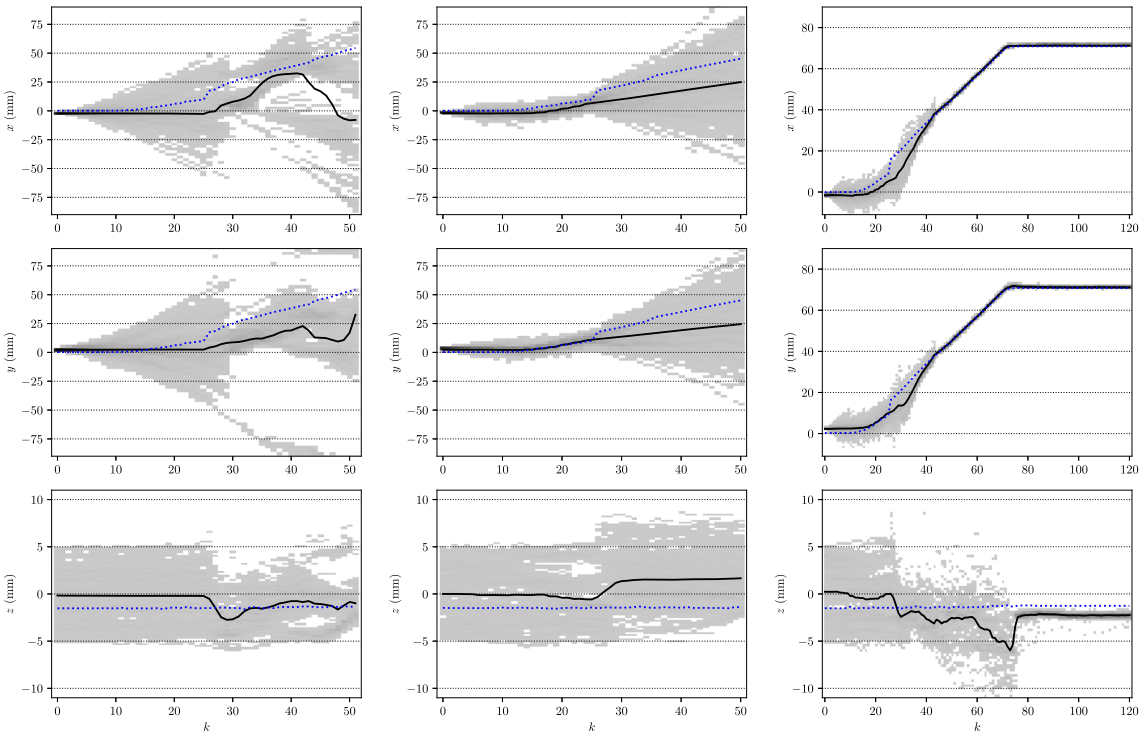
Appendix

Figures 17 and 18 show the sample evolution for the x, y and z component of the state for the experiments described in Section 6.3.



(a) $\mathcal{P}_\circ$, tactile likelihood only (b) $\mathcal{P}_\circ$, visual likelihood only (c) $\mathcal{P}_\circ$, visual-tactile likelihood

Fig. 17 Sample evolution for the x, y and z component of the state for a static cylindrical hole using three variants of likelihood functions. The gray value indicates the sample density, the black line corresponds to the expected value and the blue dotted line represents the ground truth value from the second robot, i.e. the directly measured hole pose



(a) $\mathcal{P}_\circ$, tactile likelihood only (b) $\mathcal{P}_\circ$, visual likelihood only (c) $\mathcal{P}_\circ$, visual-tactile likelihood

Fig. 18 Sample evolution for the x, y and z component of the state for a moving cylindrical hole using three variants of likelihood functions. The gray value indicates the sample density, the black line corresponds to the expected value and the blue dotted line represents the ground truth value from the second robot, i.e. the directly measured hole pose. The hole starts moving at $k = 10$, the peg is moving into contact with the hole at $k = 25$

Supplementary Information The online version contains supplementary material available at 10.1007/s10846-020-01303-z.

Acknowledgments The authors would like to thank Andreas Stemmer for the general discussions on assembly with impedance controlled robotic arms, Michael Kaßecker for the support in the implementation of the visual detector and Mikel Sagardia for providing the VPS algorithm, and Maximo A. Roa for a general revision of the paper.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Korbinian Nottensteiner and Arne Sachtler. The first draft of the manuscript was written by Korbinian Nottensteiner and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. Partial financial funding was received from the DLR-internal project "Factory of the Future".

Data Availability Videos and further visualizations of the data are provided in the supplemental material currently stored under following address.

References

1. Albu-Schäffer, A., Haddadin, S., Ott, C., Stemmer, A., Wimböck, T., Hirzinger, G.: The DLR lightweight robot: design and control concepts for robots in human environments. Industr. Robot: Int. J. **34**(5), 376–385 (2007)
2. Albu-Schäffer, A., Ott, C., Hirzinger, G.: A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. Int. J. Robot. Res. **26**(1), 23–39 (2007)
3. Allen, P.K.: Robotic Object Recognition Using Vision and Touch, vol. 34. Kluwer Academic Publishers (1987)
4. Andre, R., Jokesch, M., Thomas, U.: Reliable robot assembly using haptic rendering models in combination with particle filters. In: 2016 IEEE Int. Conf. on Automation Science and Engineering (CASE), pp. 1134–1139 (2016)
5. Asada, H.: Teaching and learning of compliance using neural nets: Representation and generation of nonlinear compliance. In: 1990 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 1237–1244. IEEE (1990)

6. Bachmann, T., Nottensteiner, K., Rodriguez, I., Stemmer, A., Roa, M.: Using task-specific workspace maps to plan and execute complex robotic tasks in a flexible multi-robot setup. In: Proc. of the 52nd Int. Symp. on Robotics (ISR) (2020)
7. Bjorkelund, A., Edstrom, L., Haage, M., Malec, J., Nilsson, K., Nugues, P., Robertz, S., Storkle, D., Blomdell, A., Johansson, R., Linderoth, M., Nilsson, A., Robertsson, A., Stolt, A., Bruyninckx, H.: On the integration of skilled robot motions for productivity in manufacturing. In: 2011 IEEE International Symposium on Assembly and Manufacturing (ISAM), pp. 1–9 (2011)
8. Bøgh, S., Nielsen, O.S., Pedersen, M.R., Krüger, V., Madsen, O.: Does your robot have skills? In: Proc. of the 43rd Int. Symp. on Robotics (ISR), vol. 6 (2012)
9. Boor, V., Overmars, M., van der Stappen, A.: The gaussian sampling strategy for probabilistic roadmap planners. In: 1999 IEEE Int. Conf. on Robotics and Automation (ICRA), vol. 2, pp. 1018–1023 (1999)
10. Boshra, M., Zhang, H.: Localizing a polyhedral object in a robot hand by integrating visual and tactile data. Pattern Recogn. **33**(3), 483–501 (2000)
11. Bruyninckx, H., Dutré, S., De Schutter, J.: Peg-on-hole: A model based solution to peg and hole alignment. In: 1995 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 1919–1924. IEEE (1995)
12. Cappé, O., Godsill, S.J., Moulines, E.: An overview of existing methods and recent advances in sequential Monte Carlo. Proc. IEEE **95**(5), 899–924 (2007)
13. Challa, S., Morelande, M.R., Mušicki, D., Evans, R.J.: Fundamentals of Object Tracking. Cambridge University Press (2011)
14. Chen, H., Eakins, W., Wang, J., Zhang, G., Fuhlbrigge, T.: Robotic wheel loading process in automotive manufacturing automation. In: 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 3814–3819. IEEE (2009)
15. Chhatpar, S., Branicky, M.: Localization for robotic assemblies using probing and particle filtering. In: 2005 IEEE/ASME Int. Conf. on Adv. Intelligent Mechatronics. Proc., pp. 1379–1384 (2005)
16. Chhatpar, S.R., Branicky, M.S.: Search strategies for peg-in-hole assemblies with position uncertainty. In: 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), vol. 3, pp. 1465–1470 (2001)
17. Craig, J.J.: Introduction to Robotics, Mechanics and Control, 3. edn. Pearson (2009)
18. Dahiya, R., Metta, G., Valle, M., Sandini, G.: Tactile sensing - from humans to humanoids. IEEE Trans. Robot. **26**(1), 1–20 (2010)
19. De Chambrier, G.P.L.: Learning search strategies from human demonstrations. Ph.D. thesis École Polytechnique Fédérale de Lausanne (2016)
20. Donald, B.: Robot motion planning with uncertainty in the geometric models of the robot and environment: A formal framework for error detection and recovery. In: 1986 IEEE Int. Conf. on Robotics and Automation (ICRA), vol. 3, pp. 1588–1593 (1986)
21. Drake, S.H.: Using compliance in lieu of sensory feedback for automatic assembly. Ph.D. thesis, Department of Mechanical Engineering Massachusetts Institute of Technology (1978)
22. Erdmann, M.: Using backprojections for fine motion planning with uncertainty. Int. J. Robot. Res. **5**(1), 19–45 (1986)
23. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-gaussian Bayesian state estimation. IEE Proc.-F Radar Signal Process. **140**(2), 107–113 (1993)
24. Goto, T., Takeyasu, K., Inoyama, T.: Control algorithm for precision insert operation robots. IEEE Trans. on Systems, Man and Cybernetics 1(10) (1980)
25. Gullapalli, V., Grupen, R.A., Barto, A.G.: Learning reactive admittance control. In: 1992 IEEE Int. Conf. on Robotics and Automation (ICRA), vol. 2, pp. 1475–1480 (1992)
26. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2 edn. Cambridge University Press (2004)
27. Hirzinger, G.: Robot-teaching via force-torque-sensors. In: Proc. of the Sixth European Meeting on Cybernetics and Systems Research, pp. 955–960 (1982)
28. Hol, J.D., Schön, T.B., Gustafsson, F.: On resampling algorithms for particle filters. In: 2006 IEEE Nonlinear Statistical Signal Processing Workshop, pp. 79–82 (2006)
29. Inoue, H.: Force feedback in precise assembly tasks. Massachusetts Institute of Technology Artificial Intelligence Labratory AI Memo 308 (1974)
30. Inoue, T., De Magistris, G., Munawar, A., Yokoya, T., Tachibana, R.: Deep reinforcement learning for high precision assembly tasks. In: 2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) (2017)
31. Jaster, R.: Agents'Abilities. De Gruyter, Berlin (2020)
32. Johannsmeier, L., Gerchow, M., Haddadin, S.: A framework for robot manipulation: Skill formalism, meta learning and adaptive control. In: 2019 Int. Conf. on Robotics and Automation (ICRA), pp. 5844–5850 (2019)
33. Jokesch, M., Suchý, J., Winkler, A., Fross, A., Thomas, U.: Generic algorithm for peg-in-hole assembly tasks for pin alignments with impedance controlled robots. In: Robot 2015: Second Iberian Robotics Conf., pp. 105–117. Springer Int. Publishing, Cham (2016)
34. Jörg, S., Langwald, J., Stelter, J., Hirzinger, G., Natale, C.: Flexible robot-assembly using a multi-sensory approach. In: 2000 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 3687–3694. IEEE (2000)
35. Kaelbling, L.P., Lozano-Pérez, T.: Integrated task and motion planning in belief space. The. Int. J. Robot. Res. **32**(9–10), 1194–1227 (2013)
36. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.: Probabilistic roadmaps for path planning in high dimensional configuration spaces. IEEE Trans. Robot. Autom. **12**(4), 566–580 (1996)
37. Kramberger, A., Gams, A., Nemec, B., Chrysostomou, D., Madsen, O., Ude, A.: Generalization of orientation trajectories and force-torque profiles for robotic assembly. Robot. Auton. Syst. **98**, 333–346 (2017)
38. Lange, F., Scharrer, J., Hirzinger, G.: Classification and prediction for accurate sensor-based assembly to moving objects. In: 2010 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 2163–2168 (2010)
39. Lee, M.A., Zhu, Y., Srinivasan, K., Shah, P., Savarese, S., Fei-Fei, L., Garg, A., Bohg, J.: Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In: 2019 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 8943–8950 (2019)
40. Leidner, D., Borst, C., Hirzinger, G.: Things are made for what they are: Solving manipulation tasks by using functional object classes. In: Humanoid Robots,12th IEEE-RAS Int. Conf. on, pp. 429–435 (2012)
41. Lozano-Pérez, T., Mason, M.T., Taylor, R.H.: Automatic synthesis of fine-motion strategies for robots. Int. J. Robot Res. **3**(1), 3–24 (1984)
42. Luo, J., Solowjow, E., Wen, C., Ojea, J.A., Agogino, A.M.: Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects. In: 2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 2062–2069 (2018)

43. Marvel, J., Falco, J.: Best practices and performance metrics using force control for robotic assembly. US Department of Commerce National Institute of Standards and Technology (2012)
44. Marvel, J.A., Bostelman, R., Falco, J.: Multi-robot assembly strategies and metrics. ACM Comput. Surv. (CSUR) **51**(1), 1–32 (2018)
45. Mason, M.T.: Compliance and force control for computer controlled manipulators. IEEE Trans. Syst. Man Cybern. **SMC-11**(6), 418–432 (1981)
46. McNeely, W.A., Puterbaugh, K.D., Troy, J.J.: Six degree-of-freedom haptic rendering using voxel sampling. In: Proc. of ACM SIGGRAPH, pp. 401–408 (1999)
47. Meeussen, W., Rutgeerts, J., Gadeyne, K., Bruyninckx, H., De Schutter, J.: Contact-state segmentation using particle filters for programming by human demonstration in compliant-motion tasks. IEEE Trans. Robot. **23**(2), 218–231 (2007)
48. Meeussen, W., Staffetti, E., Bruyninckx, H., Xiao, J., De Schutter, J.: Integration of planning and execution in force controlled compliant motion. Robot. Auton. Syst. **56**(5), 437–450 (2008)
49. Migimatsu, T., Bohg, J.: Object-centric task and motion planning in dynamic environments. IEEE Robot. Autom. Lett. **5**(2), 844–851 (2020)
50. Newman, W.S., Branicky, M.S., Podgurski, H.A., Chhatpar, S., Huang, L., Swaminathan, J., Zhang, H.: Force-responsive robotic assembly of transmission components. In: 1999 IEEE Int. Conf. on Robotics and Automation (ICRA), vol. 3 (1999)
51. Nguyen, H., Pham, Q.C.: A probabilistic framework for tracking uncertainties in robotic manipulation. arXiv:1901.00969 (2019)
52. Nottensteiner, K., Bodenmüller, T., Kaßecker, M., Roa, M.A., Stemmer, A., Stouraitis, T., Seidel, D., Thomas, U.: A complete automated chain for flexible assembly using recognition, planning and sensor-based execution. In: Proc. of ISR 2016: 47st Int. Symp. on Robotics (ISR), pp. 1–8 (2016)
53. Nottensteiner, K., Hertkorn, K.: Constraint-based sample propagation for improved state estimation in robotic assembly. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 549–556 (2017)
54. Nottensteiner, K., Sagardia, M., Stemmer, A., Borst, C.: Narrow passage sampling in the observation of robotic assembly tasks. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 130–137 (2016)
55. Quere, G., Hagengruber, A., Iskandar, M., Samuel Bustamante, D.L., Stulp, F., Vogel, J.: Shared control templates for assistive robotics. In: 2020 IEEE Int. Conf. on Robotics and Automation (ICRA) (2020)
56. Rodriguez, I., Nottensteiner, K., Leidner, D., Kaßecker, M., Stulp, F., Albu-Schäffer, A.: Iteratively refined feasibility checks in robotic assembly sequence planning. IEEE Robot. Autom. Lett., 4(2) (2019)
57. Sachtler, A., Nottensteiner, K., Kaßecker, M., Albu-Schäffer, A.: Combined visual and touch-based sensing for the autonomous registration of objects with circular features. In: 2019 19th Int. Conf. on Advanced Robotics (ICAR), pp. 426–433 (2019)
58. Sagardia Erasun, M.: Virtual manipulations with force feedback in complex interaction scenarios. Ph.D. thesis, Technische Universität München (2019)
59. Scherzinger, S., Roennau, A., Dillmann, R.: Contact skill imitation learning for robot-independent assembly programming. In: 2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 4309–4316 (2019)
60. Simons, J., Brussel, H., De Schutter, J., Verhaert, J.: A self-learning automaton with variable resolution for high precision assembly by industrial robots. IEEE Trans. Autom. Control **27**(5), 1109–1113 (1982)
61. Simunovic, S.S.N.: An information approach to parts mating. Ph.D. thesis, Department of Mechanical Engineering Massachusetts Institute of Technology (1979)
62. Steinmetz, F., Weitschat, R.: Skill parametrization approaches and skill architecture for human-robot interaction. In: 2016 IEEE International Conference on Automation Science and Engineering (CASE), pp. 280–285 (2016)
63. Stemmer, A., Albu-Schäffer, A., Hirzinger, G.: An analytical method for the planning of robust assembly tasks of complex shaped planar parts. In: 2007 IEEE Int. Conf. on Robotics and Automation (ICRA). IEEE (2007)
64. Sun, Z., Hsu, D., Jiang, T., Kurniawati, H., Reif, J.: Narrow passage sampling for probabilistic roadmap planning. IEEE Trans. Robot. **21**(6), 1105–1115 (2005)
65. Taguchi, Y., Marks, T., Okuda, H.: Rao-Blackwellized particle filtering for probing-based 6-dof localization in robotic assembly. In: 2010 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 2610–2617 (2010)
66. Takami, K., Furukawa, T., Kumon, M., Kimoto, D., Dissanayake, G.: Estimation of a nonvisible field-of-view mobile target incorporating optical and acoustic sensors. Auton. Robot. **40**(2), 343–359 (2016)
67. Thomas, U., Hirzinger, G., Rumpe, B., Schulze, C., Wortmann, A.: A new skill based robot programming language using uml/p statecharts. In: Proceedings of the 2013 IEEE International Conference on Robotics and Automation, pp. 461–466. IEEE (2013)
68. Thomas, U., Molkenstruck, S., Iser, R., Wahl, F.: Multi sensor fusion in robot assembly using particle filters. In: 2007 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 3837–3843 (2007)
69. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge (2005)
70. Wahrburg, A., Zeiss, S., Matthias, B., Peters, J., Ding, H.: Combined pose-wrench and state machine representation for modeling robotic assembly skills. In: 2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 852–857. IEEE (2015)
71. Whitney, D., Nevins, J.L.: What is the remote center compliance (RCC) and what can it do. In: Proc. of the 9th Int. Symp. on Industrial Robots (ISIR) (1979)
72. Wirnshofer, F., Schmitt, P.S., Feiten, W., Wichert, G., Burgard, W.: Robust compliant assembly via optimal belief space planning. In: 2018 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 1–5 (2018)
73. Wirnshofer, F., Schmitt, P.S., Meister, P., Wichert, G., Burgard, W.: State estimation in contact-rich manipulation. In: 2019 IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 3790–3796 (2019)
74. Xu, J., Hou, Z., Liu, Z., Qiao, H.: Compare contact model-based control and contact model-free learning: a survey of robotic peg-in-hole assembly strategies. arXiv:1904.05240 (2019)

Korbinian Nottensteiner holds a Dipl.-Ing. (Univ) degree in mechatronics and information technology from Technical University of Munich (TUM). Currently, he is doing his doctorate at TUM in the field of robotic assembly. Since 2012 he is working as a researcher at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR). He leads the team for autonomous robotic assembly systems and coordinates research and development activities in the domain of future manufacturing. A primary goal in his research is to enable autonomous robotic assembly by situation aware manipulation skills using compliant control, contact sensing and task abstractions.

Arne Sachtler is a researcher at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR) and at the informatics department of the Technical University of Munich (TUM). After receiving a bachelor's degree in computer science from DHBW Mannheim in 2017, he obtained a master's degree in Robotics, Cognition, Intelligence from TUM in 2020. His research focuses on the application of differential geometry to nonlinear dynamics and control, on fusing sensory data of multiple modalities for object pose estimation as well as on robotic assembly.

Alin Albu-Schäffer received his M.S. in electrical engineering from the Technical University of Timisoara, in 1993 and his Ph.D. in automatic control from the Technical University of Munich in 2002. Since 2012 he is the head of the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR), which he joined in 1995. Moreover, he is a professor at the Technical University of Munich, holding the Chair on "Sensor Based Robotic Systems and Intelligent Assistance Systems" at the Computer Science Department. His personal research interests include robot design, modeling and control, nonlinear control, flexible joint and variable compliance robots, impedance control, physical human-robot interaction, bio-inspired robot design and control. He received several awards, including the IEEE King-Sun Fu Best Paper Award of the Transactions on Robotics in 2012 and 2014; several ICRA and IROS Best Paper Awards as well as the DLR Science Award.